Wireless Communications and Mobile Computing

Securing AI-powered Internet of Things (IoT) Ecosystems

Lead Guest Editor: Yan Huo
Guest Editors: Liran Ma and Ruinian Li

Securing AI-powered Internet of Things (IoT) Ecosystems

Wireless Communications and Mobile Computing

Securing AI-powered Internet of Things (IoT) Ecosystems

Lead Guest Editor: Yan Huo
Guest Editors: Liran Ma and Ruinian Li

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8984789, 12 pages
https://doi.org/10.1155/2022/8984789

Research Article

DIM-Based Random Number Generation Using Quantum Noise Resources

Hansaem Wi,[1] **Seyoon Lee**,[1] **and Okyeon Yi**[2]

[1]*Department of Financial Information Security at Kookmin University, Seoul 02707, Republic of Korea*
[2]*Department of Information Security Cryptology and Mathematics at Kookmin University, Seoul 02707, Republic of Korea*

Correspondence should be addressed to Okyeon Yi; oyyi@kookmin.ac.kr

Received 29 March 2022; Revised 17 August 2022; Accepted 28 September 2022; Published 9 November 2022

Academic Editor: Yan Huo

Currently, unmanned aircraft systems (UASs) or drones are in service in various industrial fields, and each UAS operator establishes and operates their own independent drone system. These individual drone systems interact only with their own components without any integrated management. As the number of UASs is increasing due to the expansion of the drone industry, standardized operation is required. Therefore, to integrate and manage existing drone systems, the Federal Aviation Administration and National Aeronautics and Space Administration devised UAS Traffic Management (UTM). The drone identity module (DIM), which is being developed as a drone identification device, securely stores the remote identification (RID) of each drone and performs a cryptographic operation to secure information between the drone and UTM infrastructure. The DIM performs cryptographic authentication protocols to achieve cryptographic identification and authentication with the UTM infrastructure, which requires random numbers. Modern cryptographic systems rely on difficult computations, and an environment capable of generating secure cryptographic random numbers must be configured to provide high computational costs to attackers. In this paper, we explain the need for random numbers in the DIM, analyze random number generators used in related drone-based studies, and analyze the characteristics of noise resource generation devices that can be used in existing drone systems. Subsequently, based on the analysis results, existing methods are used to generate random numbers in the DIM, and limitations are derived. To overcome these limitations, we propose a method of generating random numbers in the DIM using quantum noise resources. For our proposal, we conduct an analysis of the physical specifications of noise resource generation devices, DIM prototypes, and quantum noise resource generators in existing drone systems, and we present the results of NIST 800-90B entropy measurement using data collected from quantum random number generators.

1. Introduction

With the advent of 4th Industrial Revolution, data security in the IoT environment is becoming important, and various security technologies are being actively studied accordingly [1, 2]. In addition, drone systems are used in various industrial fields in interaction with IoT and are becoming one of the most important industrial fields in the 4th Industrial Revolution. It is estimated that the number of unmanned aircraft systems (UASs), i.e., drones, currently operated at low altitude for recreational and commercial purposes in the United States will increase from approximately 2 million in 2021 to approximately 3 million by 2023 [3]. To design an architecture capable of integrating and managing such an increasing number of UASs, the Federal Aviation Administration (FAA), a US airspace management agency, announced Concept of Operation v2.0 (ConOps v2.0) [3], which explains the necessity of UAS Traffic Management (UTM), details of UTM design, drone operation scenarios within UTM, and requirements for drones participating in UTM. Drones are a well-known example of UASs, and UTM includes systems for operating drones. From an information protection and cryptographic perspective, the most important of the various requirements described in Ref. [3] are identification and certification of drones performing services within UTM. The expansion of the drone

industry, such as drone taxis, drone delivery, and precision agriculture using drones, will necessarily require UTM, thereby making it an important infrastructure for the country. However, security incidents caused by security threats in critical infrastructure can cause enormous economic and human damage. Accordingly, security for data communicated between drones and UTM is essential. To form a secure channel between drones and UTM, identification and authentication are essential.

However, existing drone systems use products released by large drone manufactures, such as DJI and Parrot, or use open platforms such as Pixhawk and Raspberry Pi, and there is no standard between these heterogeneous drones. These drones cannot be included and operated in UTM because they only use functions dependent on existing platforms without requirements for separate identification and authentication functions. Therefore, for existing drone systems to be included in UTM and perform services, an authentication method that can integrate each drone system is required, and ISO 23629-8 standardizes requirements for remote identification of drones [4]. In addition, ISO/IEC 22460-2 standardizes requirements and details for the dataset, cryptographic operation function, and hardware of the drone identity module (DIM), a standardized device for identifying drones included in UTM [5]. Currently, standardization of the DIM is in the preparatory stage, and if standardization is completed and applied to drones, it will be a security-only module that provides information security for communication between drones and UTM, not just identification information storage for a drone.

ConOps v2.0 states that cryptographic authentication protocols are required for identification and authentication of drones flying under UTM [3]. In other words, the DIM installed on the drone must form a secure data communication channel based on cryptographic authentication with the UTM authentication server. In this case, the cryptographic authentication protocol requires a random number, and accordingly, the cryptographic random number generation function is essential for the authentication server of the DIM and UTM. Cryptographic random numbers are used not only as time variants in cryptographic authentication protocols but also to generate security parameters, such as cryptographic keys that determine the safety of the cryptographic system. Determining the security strength of cryptographic random numbers is the entropy of the noise resources constituting the seed for generating random numbers. Therefore, the DIM should be equipped with a device capable of generating sufficient noise resources to satisfy the security strength of the cryptographic random number along with the cryptographic random number generation function. However, there are no published data on a method of generating random numbers for the DIM. Moreover, the specific details of available noise resource generating devices and related research are insufficient. In this study, we conducted an analysis focusing on how to generate noise resources to derive the requirements necessary to propose a random number generation method for the DIM. We analyzed the limitations of noise resource generation methods using existing drone systems, based on which we propose a random number generation method for the DIM using quantum noise resources.

The contributions of this study can be summarized as follows:

(i) Analysis of differences in operating environment between existing drone systems and UTM: from the perspective of random number generation methods, we analyzed the differences between existing drone systems and the operating environment of drone systems in UTM, and we analyzed how these differences affect random number generation methods

(ii) Analysis of usage limits for random number generator in existing drones: based on the contents of related studies, we analyzed the limitations of methods used for random number generation in existing drone systems when applied to the DIM, and based on this, we derived the requirements necessary to prepare a random number generation method in the DIM

(iii) Proposal of DIM using quantum noise source: based on the analysis results of the limitations of noise resource generation methods using existing drone systems, we propose a random number generation method for the DIM using quantum noise resources

2. Background

2.1. UTM System. ConOps v2.0 shows the interaction between participants in UTM through the notional UTM architecture and describes the functions and requirements of each participant in UTM [3].

Each participant in UTM described in Figure 1 refers to a system developed and distributed by the FAA and industry as a specific function for UTM operation, and UTM is operated through organic data communication between these systems. The main participants that make up UTM include flight information management system (FIMS), UAS service supplier (USS), supplementary data service provider (SDSP), and UAS operators. FIMS is a system operated by a national airspace management agency and is UTM's best management system to record UAS operations and track data that can be used for incident management and audit in the future. The USS provides a network that allows multiple UAS operators to receive real-time flight information from FIMS and SDSP databases and provides flight management. UAS operators can receive auxiliary data, such as current flight plans and weather management required for flight through the USS. For organic data communication of these components, UTM recommends configuring secure channels based on cryptographic identification and authentication between entities [3].

2.2. Drone Identity Module. Currently, there is no standard document for reference to the operation of the DIM. However, some information can be obtained from Dr. Tak's presentation at the Unmanned System Congress held in

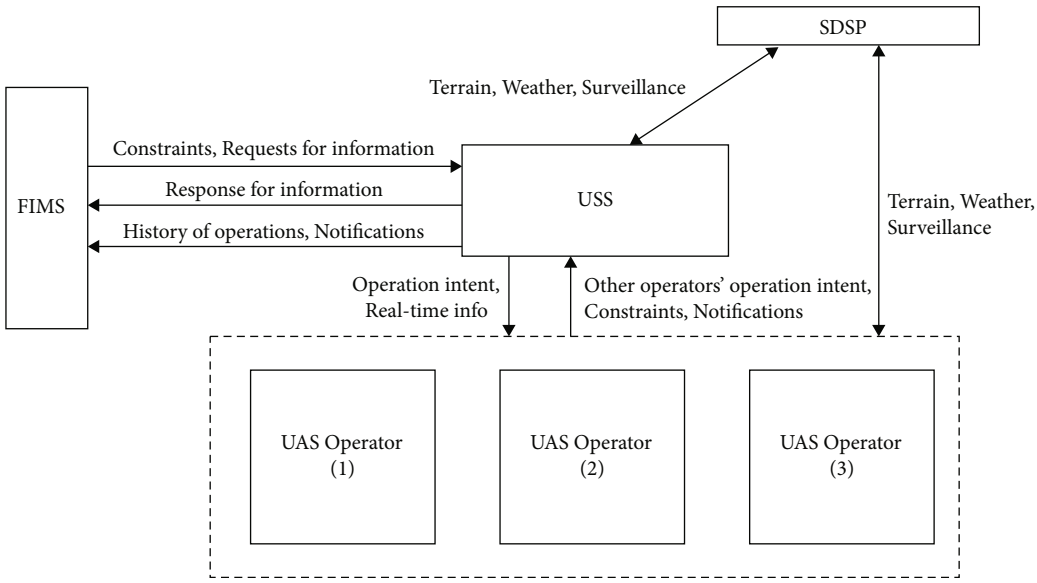


Figure 1: Notional UTM architecture [3]. This figure shows the general operation structure of UTM. FIMS requests USS to record flights performed by constrain and UAS operators at UTM, and USS, which manages UAS operators, requests such information back to UAS operators and delivers it to FIMS. FIMS stores and manages records of all flights occurring in UTM and uses these records as data for future accidents or audits. SDSP can deliver data related to terrain, weather, surveillance, etc. that can support drone flight to UAS operators through USS or directly to UAS operators. For the above interaction, drones operated by UAS operators must perform the flight permit authorization process to participate in UTM.

Korea in 2021. Dr. Tak is the convener of ISO/IEC JTC 1/SC 17/WG 12, which performs the standardization of ISO/IEC 22460. According to his presentation, drones must obtain flight permission from the UTM Flight Permit Authority Server to fly under UTM, and in the process, cryptographic mutual authentication between the DIM and UTM Flight Permit Authority Server is performed. The Flight Permit Authority Server does not exist in existing drone systems, and the process of obtaining a drone's flight permission is an additional process in UTM. Existing drone systems consisting of commercialized platforms initiate flights in a manner specified by the platforms used by each UAS rather than an integrated flight permit process, and identification and authentication processes using cryptographic methods are not necessarily required. However, Dr. Tak's presentation showed that for drones to start flying in UTM using the DIM, they must perform a Flight Permit Authority process using standardized RIDs and cryptographic authentication protocols [4, 6]. This process is to prevent unauthorized illegal drones from entering the UTM and securely manage drones that perform legitimate services, and this process of preflight licensing is the biggest difference between existing drone systems and UTM operations. Figure 2 shows the components of the DIM. The DIM uses the drone's RID to perform flight authorization, and the identification issued after registration allows UTM to manage drone behavior while the drone is flying. As described in ISO/IEC 22460-2, the DIM is a module for drones that can securely store identification information and perform cryptographic functions necessary for information security [5].

Two essential functions are required from the DIM [5]. First, a function capable of storing information necessary for drone identification and authentication is required. Here, the information required for identification and authentication includes identification information, registration information, a certificate, and an encryption key that allows the drone to obtain airspace flight permission and maintain its status during flight. The DIM should be in the form of a black box so users are not authorized to access such information. Second, a cryptographic operation function for the information security function performed by the DIM is required. A drone uses the DIM to perform identification and certification processes with UTM infrastructure before and during flight using RIDs stored in the DIM. When performing the identification and authentication process, mutual authentication is performed through a cryptographic protocol, and the DIM must be able to execute cryptographic functions, such as electronic signature, hash function, and random number generation at this time.

2.3. Entropy Sources, Random Numbers, and Quantum Random Number Generator. When building a cryptosystem, generating random numbers that can satisfy unpredictable security strengths with the computing power of an attacker is as important as accurately implementing a cryptographic algorithm. Cryptographic algorithms in modern cryptosystems, such as encryption, message authentication code generation, and public key systems, provide security against computational complexity assuming that keys generated in a cryptographically secure manner are securely stored [7, 8]. In other words, if a cryptographic random number generator is implemented where bits of security parameters are predictable enough to make an attack meaningful or biased

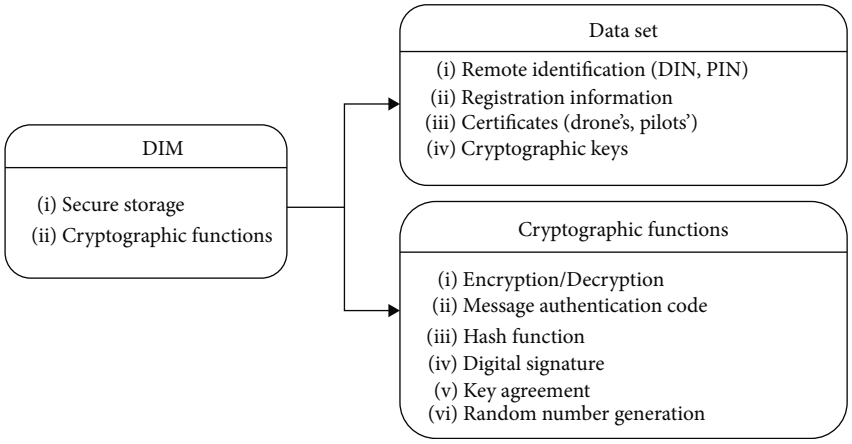


Figure 2: DIM components. It must have a black box type storage that can securely store identification information and security parameters and a cryptographic function for performing information security functions [5].

enough to not be used as security parameters, the cryptographic system may be destroyed by an attacker.

The random number generator required by ISO/IEC 19790 sets an international standard for the security of cryptographic modules in the form of software and hardware in various devices, including IT systems. ISO/IEC 19790 requires a random number generator that refers to the model presented in NIST 800-90A [9]. The model is a structure that generates a seed based on noise resources and enables repeated random number generation using a deterministic random number generator. The deterministic random number generator operates based on block cipher, Hash, and HMAC, and the security of the generated random number depends on the seed. The entropy of the seed input to the deterministic random bit generator (DRBG) must satisfy the security strength of random number to be generated. Figure 3 shows the random number generation method given by NIST 800-90A [10].

Entropy resource refers to a bit string after postprocessing, such as a process for removing bias that may occur due to noise resource characteristics. Assuming that the DRBG described in Figure 3 is implemented without errors, the entropy of noise resources with nondeterministic properties determines the security of the corresponding random number generator. Noise resources available in random number generators used in general IT systems use data indicating the state of software running on the operating system (OS) or values input through user interfaces, such as mouse and keyboard, with entropy characteristics. Noise resources obtained based on hardware include the thermal noise of a resistor and atmospheric noise of a diode. The collected noise resources are used either directly to construct the seed of the random number generator or as entropy resources after increasing the entropy ratio through a postprocessing process [11]. Random numbers generated through the process shown in Figure 3 are used to generate security parameters in the cryptosystem. A well-known example is that when executing cryptographic authentication protocols, the provider generates random numbers to defend against attacker replay and interleaving attacks and uses them as time variants to provide unity and timeliness of the protocol being executed [12].

Quantum random number generator (QRNG) refers to a device that generates unpredictable random numbers based on quantum noise. It is assumed that the usage structure of QRNG described in this paper follows the structure of the random number generator shown in [10], as described in Figure 4. The reason for this assumption is that the type of random number generator suggested in Ref. [10] follows the structure referenced in [9], which is an international standard, so the QRNG must have this structure to be applied from private facilities to important national infrastructure. Well-known QRNG methods are to use noise resources generated during beam splitter transmission of a single photon and radioisotope decay [13–15]. Quantum noise is a signal that is collected based on the phenomenon that protons, which are very small particles, inherit the uncertainty of both. Therefore, compared to noise obtained from general software and hardware, it is possible to create an entropy source with high entropy, and because quantum particles are very small, it is possible to construct an environment that can generate noise in a physically small space [13].

3. Related Work

3.1. Authentication Scenarios with DIM [16]. In [16], DIM standardization editors propose the cryptographic authentication structure of the DIM with UTM's authentication server and implemented the authentication structure using the oneM2M platform, a protocol for application data communication of IoT devices. In [16], a certificate-based authentication protocol between the DIM and authentication server is proposed, and the implementation process and experimental results are presented. It should be noted that in [16], the proposed authentication structure attempted to provide freshness for sessions that perform authentication processes using random numbers generated by the DIM and authentication server, and normal random number generation is premised.

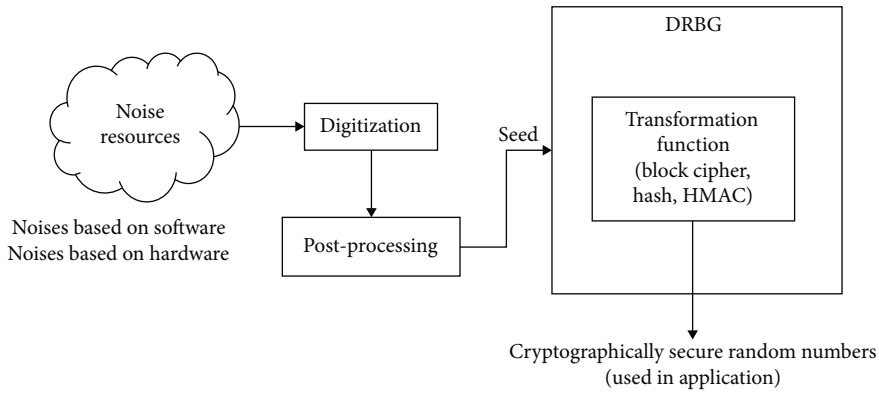


Figure 3: Random number generation process in cryptographic system using DRBG [10]. The security strength of the random number depends on the seed obtained by processing the noise resources. In general, noise resources may be obtained from an environment for operating an existing IT system or a device of an IoT system. In some cases, a dedicated device for generating noise resources is configured and used to generate random numbers.

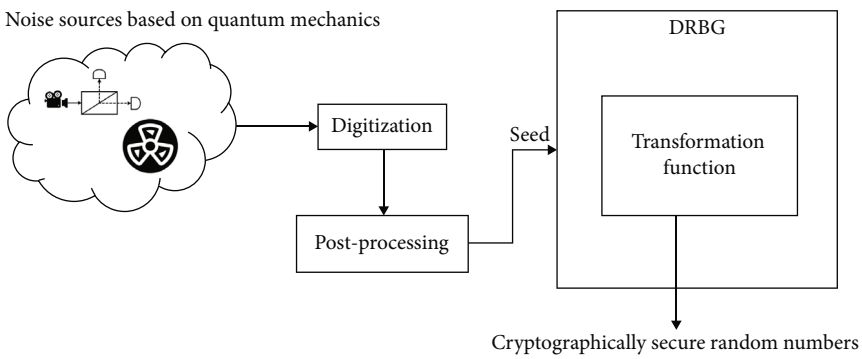


Figure 4: Random number generation process in cryptographic system using quantum mechanics. When configuring the quantum random number generator, the most important aspect is building small parts that generate quantum noise resources. This is directly related to the cost of a quantum random number generator, and accordingly, commercialized quantum random number generators are produced in a compact size compared to general hardware parts. It is well-known how to generate a random sequence using the spacing of random signals generated during the decay of radioactive isotopes or by calculating photons passing through or reflected from beam splitters.

Figure 5 shows the cryptographic mutual authentication structure proposed in [16], and it can be seen that it has an authentication structure based on a certificate. The server uses the oneM2M platform to send and receive messages, and after the authentication process is successfully completed, the shared key is calculated using the random numbers shared with each other. The authentication and key matching process using the certificate proposed in Ref. [16] is widely used in general IT systems as well as in drone systems combined with IoT environments, and the oneM2M platform is a widely used message protocol in IoT environments.

However, the authentication structure proposed in Ref. [16] is performed assuming normal random number generation in the DIM and authentication server. It can be seen that random number generation is essential in the DIM, and accordingly, it is necessary to consider how to generate random numbers considering the software and hardware characteristics constituting the DIM.

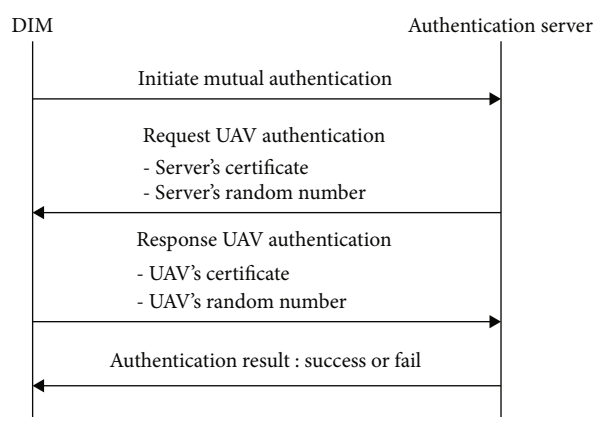


Figure 5: Mutual authentication structure proposed in [16].

3.2. Drone Random Number Generator Based on Sensor Data. In [17], a drone random number generator is proposed using data from sensors installed to control and monitor the state of the drone, which is an essential component in a general drone system, as noise resources. The motivation for Ref. [17] was Ref. [18], which studied the secure version of MAVLink, an open drone control message

protocol. In Ref. [17], the authors claim that the security of the cryptographic key must be ensured for the operation of the cryptographic system in the drone, which was conducted in [18]. For this reason, they designed a random number generator for the secure generation of security parameters, such as the cryptographic key. The drone used for the experiment in this research is equipped with Pixhawk and Raspberry Pi as flight controllers. Pixhawk and Raspberry Pi are currently widely used in DIY drones due to their convenience of development and openness of development-related materials. It is thought that the authors chose these settings to design a random number generator that can be commonly applied to general drone systems. In [17], data collected from an accelerometer, gyroscope, and barometer, which are sensors related to the attitude and speed of the drone, were used as noise sources for the random number generator.

Figure 6 sequentially shows the process of the experiment performed in [17]. The authors performed the statistical tests recommended in NIST SP 800-22 [19] to check the quality of random numbers output by the random number generator. The results of the NIST 800-22 test for the random number generator output for the drone in Ref. [17] are presented in Table 2 in [17], indicating that the random number generator outputs designed by the authors of Ref. [17] are statistically random. It is worth focusing the test results of the raw sensor data collected when the drone is stationary in the results presented in Ref. [17]. The stationary state expressed by the authors refers to a state in which the drone does not fly and is stationary on the ground with the power turned on. The authors of Ref. [17] performed the NIST 800-22 test suite on raw data collected from the accelerometer, gyroscope, and barometer, and as a result, some NIST 800-22 tests failed. DroneRNG designed dividing and shuffling functions to improve the lack of randomness in raw data. The random numbers generated through the "mix and swap" process showed success on the entire NIST 800-22 test suite. The method of Ref. [17] is meaningful in that it generates random numbers necessary for operating a cryptographic system in a drone using Pixhawk and Raspberry Pi, which are widely used in commercial drone systems.

4. Experiment and Discussion

4.1. Analysis of Noise Resources Available in Existing Drone Systems. Figure 7 shows the structure of an existing drone system operated without the introduction of the UTM concept. Existing drone systems use the Ground Control System (GCS), a software that can control commercialized drones, to configure and perform services. The types of drones used range from ready-to-fly drones manufactured and released by large manufacturers, such as DJI and Parrot, to DIY drones using open hardware platforms, such as Pixhawk and Raspberry Pi. The drone must be equipped with a flight controller, a computer that processes logic related to flight, and an optional companion computer, a computer for further user data processing. The flight controller and companion computer interact with the GCS and user application server, respectively, to control, monitor, and transmit user data. The flight controller is a device that controls the posture and speed of a drone when it flies and allows the UAS operator to monitor the drone's state based on the values of sensors installed in the drone. Because drones' flight capabilities are heavily weighted, flight controllers are generally manufactured with compact specifications, such as with Pixhawk and Naze32. The companion computer is mounted on the drone and installed to process user data, such as video and photos. The companion computer is built to have higher specifications than the flight controller to process high-definition images and operates on rich operating systems, such as Windows and Linux.

The biggest change in drone systems after the introduction of the concept of UTM is that each drone system must participate in UTM in a unified manner. Unlike existing drone systems that allowed drones to fly only with a connection between the flight controller and GCS, the UTM architecture requires DIMs that store standardized RIDs to participate in the system. As explained in Background, the DIM must obtain permission to fly through cryptographic authentication with UTM's Flight Permit Authority Server and must generate random numbers. Accordingly, we select the flight controller as a candidate random number generating device among the elements constituting existing drone systems, and we analyze the limitations of this method using the research contents of [17].

Random number generation follows two steps [10]:

Step 1: construct seeds based on collected noise resources

Step 2: DRBG operation by inputting parameters for random number generation including the seed configured in Step 1

The DRBG operating in Step 2 is a deterministic algorithm that always has the same result for the same input. For this reason, it can be implemented in software and on various types of processors. However, because collecting noise resources used in Step 1 requires resources with nondeterministic characteristics generated by software or hardware, the method of generating noise resources is implemented in various ways depending on the software, hardware types, and characteristics of the encryption system.

We have established two scenarios for collecting the required noise resources in the DIM. The first is to generate noise sources using components of existing drone systems, and the second is to generate noise resources using the DIM itself. However, in the first scenario, we excluded the noise source collection scenario using the companion computer from the analysis to assume the most common situation, as the scenario of collecting noise sources from the optional companion computer component cannot be applied to drones that operate with a flight controller alone.

The flight controller has several types of sensors built in to control the posture and speed and monitor the conditions of the drone, and additional sensors are sometimes attached to improve performance. The data output by these sensors are mixed with natural noise when drones fly to create irregular patterns and have entropy characteristics. In [17], which designed a random number generator using sensor data as a noise source, a random number generator using

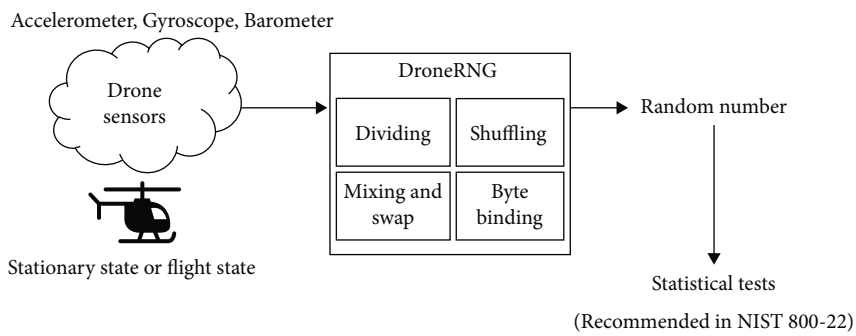


Figure 6: Random number generator and statistical test process proposed in [17].

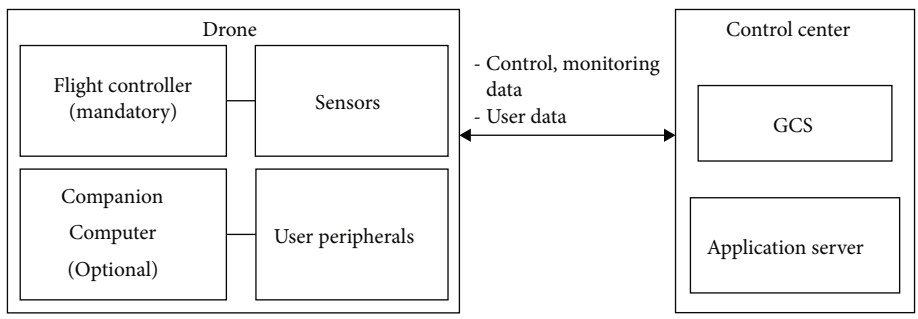


Figure 7: General drone system structure. This figure shows the general structure of existing drone systems. There are visual line of sight (VLOS) and beyond visual line of sight (BVLOS) methods for controlling drones. In the VLOS method, the pilot uses a pilot controller to control the drone within visible distance, and the BLOS method means that the drone is controlled outside the pilot's field of view. The structure of the figure represents an existing drone system using the BVLOS method. ConOps v2.0 [3] describes a UTM architecture that considers both VLOS and BLLOS drone control.

the accelerometer, barometer, and gyroscope mounted on a drone were designed using Pixhawk and Raspberry Pi as noise resources.

However, because these contents use the principle of generating noise resources using irregular pattern in air resistance, pilot manual operation, and sensor values due to obstacle avoidance, the conditioning process cannot guarantee entropy of noise resources. In addition, Ref. [17] claims that the conditioning process was performed to improve the entropy of the raw data of the sensors, and the results of the statistical random number test were presented through the results on the NIST 800-22 test suite. However, the conditioning process can only increase the entropy ratio by removing the bias of the noise resource, not the absolute entropy of the noise resource, and the NIST 800-90B test suite [11] must be performed to determine how much entropy has improved. In other words, the results presented in Ref. [17] cannot be seen as evidence for entropy improvement. The evaluation method of entropy and random number should be different, but a method for statistically evaluating random numbers was used to evaluate the collected entropy. Therefore, from a strict point of view, Ref. [17]'s experimental analysis is wrong. In order for the experimental analysis of Ref. [17] to be done properly, the authors should have measured the entropy obtained from the sensor data through statistical tests such as SP 800-90B and presented how much improvement the entropy obtained from the pure sensor data was made when the designed method was designed.

The values output by the sensors of the drone change within a small margin of error when the drone is not flying. For this reason, it is obvious that the entropy value of sensor data represents a small value compared to when flying. More entropy may be measured in places where natural noise is severe, but otherwise, it is difficult to supply the entropy required for the drone at an appropriate time.

Therefore, to generate entropy that is not affected by the state of the drone and always provides the same strength of security, a noise resource that is not affected by the state of flight is required, and the NIST 800-90B test suite must measure the level of noise entropy.

In Figure 8, the drone is in an "unlicensed" state and cannot fly under UTM. The drone will perform mutual authentication with the UTM Flight Permit Authority to acquire flight rights. In this process, the drone and Flight Permit Authority perform cryptographic protocols, and random number generation is required. Because the drone landed on the ground and received only the power needed for communication and is not in flight, the values of the sensors installed in the drone will only show small changes in values due to natural noise and will not show any significant uncertainty. Based on the above analysis results, it can be seen that noise-generating devices, such as sensors installed in the drone, depend on the flight of the drone,

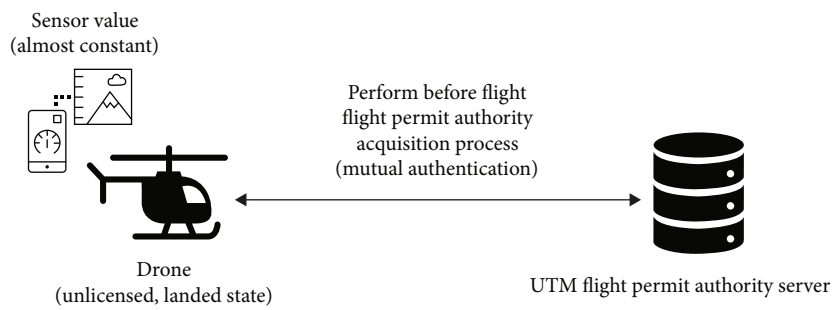


Figure 8: Status of drones and sensors when performing cryptographic authentication with UTM Flight Permit Authority Server. This figure shows the state of the sensor value installed in the drone when the drone and UTM Flight Permit Authority perform cryptographic authentication. Drones in the unlicensed state cannot fly until they obtain flight permission.

and there is a limit to cover scenarios in which the flight of the drone is not performed, such as Flight Permit Authorization. Through this, a method for collecting noise resources with the same level of entropy before and after the drone flight is performed is needed.

4.2. DIM Structure Using Quantum Noise Source. A way to overcome the limitations of the noise resource collection scenario using existing drone systems is to mount the noise resource generating device on the DIM itself, which is the second set scenario. However, to mount the noise resource generating device on the DIM, it is necessary to have a compact size and always provide the same level of entropy without being affected by the flight of the drone.

We propose DIM using QRNG as a way to satisfy these conditions. The noise resource generation device using quantum mechanics used in QRNG can be implemented at a compact size, and as long as the normal operation of the device is guaranteed, it can supply the noise resources required for the DIM independently of the drone's flight.

Table 1: Prototype DIM dimensions [20, 21]. In the case of SoC chip, it means an SoC with microcontroller mainly used in the mobile and IoT environments, and the general size is not indicated because each manufacturer has different specifications. However, compared to the flight controller and companion computer used in existing drone systems, it is manufactured with a compact size.

Hardware	General size (width × length × height; mm)
SD card	32.0 × 24.0 × 2.1
Mini SD card	21.5 × 20.0 × 1.4
Micro SD card	15.0 × 11.0 × 1.0
Full-size SIM	85.6 × 53.9 × 0.76
Mini SIM	25.0 × 15.0 × 0.76
Micro SIM	15.0 × 12.0 × 0.76
Nano SIM	12.3 × 8.8 × 0.67
SoC chip	Varies

4.2.1. Compact Size of QRNG. Availability is an essential information security function that must be satisfied along with confidentiality and integrity when composing a cryptographic system. In the case of drones, their weight and size can lead to reduced ability to fly, and smaller drones are more sensitive to this. Therefore, the weight and size of the hardware installed on the drone to configure the cryptographic system are very important. In this analysis, we compare the noise generator candidates for existing drone systems with the physical specifications of the noise generators used in QRNG to analyze whether they have sufficiently compact size to be mounted on the DIM. What should be noted in the analysis results below is the overhead of the weight and area that occurs when additionally installed in a drone to generate entropy. A mini computer such as Raspberry Pi is a device designed to run an operating system with a built-in multichip that handles multiple functions. Sensors can include chip that can digitally convert analog signals and the weight and area of the case surrounding them. Finally, the devices that generate the quantum noise source proposed in this paper are hardware composed of single chip. These devices operate through system communication with the hardware that makes up the device, and their area is also very small compared to the two aforementioned hardware.

In general, the weight of the hardware increases in proportion to the area of the hardware. The hardware that generates the chip-type quantum noise source proposed in this paper has an area of up to 5.0 × 5.0 mm, so it is much smaller than the comparison Raspberry Pi and sensor. Therefore, the weight is significantly lighter than the two hardware. For this reason, the comparison of weights is omitted and the overhead in the area is sufficient to analyze the possibility of installation in the DIM prototype.

Table 1 lists the physical specifications of SD cards and SIM cards that are being studied as DIM prototypes. Table 2 lists the devices that can collect noise resources for generating random numbers in existing drone systems and is presented to compare the difference in specifications with the DIM prototype. Although companion computer was excluded from the noise generation method using existing drone systems, the size of the Compute Module, the smallest version of Raspberry Pi, is presented in the table to show the difference between the physical specifications of the existing drone system and DIM.

Table 2: Size of Raspberry Pi Compute module and Pixhawk's sensor. The thickness of Raspberry Pi indicates only the height of the PCB excluding the height of components. In the case of sensor, it shows the specifications of the sensor used in Pixhawk, which is widely used in general drone systems. MPU 6000 is a sensor used as the main accelerometer and gyroscope of Pixhawk, and the MS5611 series is a sensor used as the main barometer of Pixhawk. The reference link to the specification for each product is as follows: Compute Module 3+: https://www.raspberrypi.com/products/compute-module-3-plus/, Compute Module 4: https://www.raspberrypi.com/products/compute-module-4/, MPU 6000: https://invensense. http://tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf, and MEAS: https://datasheetspdf.com/pdf/921406/measurement/MS5611-01BA03/1.

Manufacturer	Product	Size (width × length × height; mm)	Dependent on drone flight
RPi	Compute Module 3+	67.6 × 31.0 × 1.1	N
RPi	Compute Module 4	55.0 × 40.0 × 1.3	N
InvenSense	MPU 6000	4.0 × 4.0 × 0.9	Y
MEAS	MS5611-01BA	5.0 × 3.0 × 1.0	Y

Table 3: Size of noise-generating devices used in commercial QRNG. This table shows the sizes of noise generators used for quantum noise generation in QRNG. The environment that can generate quantum mechanical phenomena is configured in the noise generator. The noise generator manufactured by EYL uses radioactive isotope decay, and the noise generator manufactured by IDQ uses photons to generate noise. The reference link to the specification for each product is as follows: EYL: https://www.eylpartners.com/index.php/quantum-entropy-chip/ and IDQ: https://www.idquantique.com/random-number-generation/products/quantis-qrng-chip/.

Manufacturer	Product	Size (mm)	Dependent on drone flight
EYL	QEC 1.0	5.0 × 5.0	N
EYL	QEC 2.0	3.0 × 3.0	N
EYL	QEC 3.0	3.0 × 3.0	N
IDQ	IDQ6MC1	4.2 × 5.0 × 1.1	N
IDQ	IDQ20MC1	4.2 × 5.0 × 1.1	N
IDQ	IDQ250C2	2.5 × 2.5 × 0.84	N

Compared to the physical specifications of the DIM prototype presented in Table 2, the physical specifications of MPU 6000 and MS5611-01BA in Table 3 are smaller than those of the DIM prototype. However, as confirmed in the previous analysis of the noise resource collection limit in existing drone systems, the sensor data cannot provide a constant entropy independent of whether the drone is flying, so it is not suitable to be mounted as a noise-generating device in the DIM. In addition, considering only the physical specifications, the physical specifications of the Compute Module, the smallest version of Raspberry Pi, are larger than the specifications of most DIM prototypes. Therefore, this is also not suitable for mounting as a noise-generating device in the DIM.

Table 3 lists the sizes of noise generators used in commercialized QRNG. The devices presented in Table 3 have compact specifications and can be installed mostly under the standard specifications of the prototype of the DIM shown in Table 1. The reason for this is that quantum mechanical phenomena can be fabricated with very small hardware. Quantum noise-generating devices can generate a noise source that always has the same level of entropy independently of the flight of the drone if it is assumed that the hardware in which the quantum mechanical phenomenon is implemented is preserved. In addition, in the case

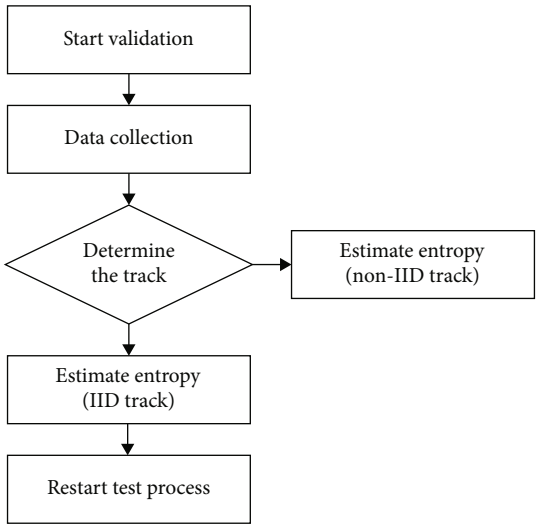


Figure 9: NIST SP 800-90B entropy estimation strategy [11]. This figure shows the entropy measurement process described in NIST SP 800-90B. The entropy evaluation process is performed in a statistical manner. If a dataset has IID characteristics, IID track is performed. Otherwise, statistical entropy measurement is performed on non-IID track. In this experiment, the restart test process was not performed.

Table 4: Specification of QRNG. This table shows the details of the QRNG used to collect the dataset for the experiment. We used QRNG based on the radioactive decay phenomenon to collect quantum noise sources, and the experiment was performed with the QRNG's microcontroller and the noise generator configured for communication.

QRNG	Description
Noise type	Radioactive decay
Sampling	60-100
Size (noise)	3.0 × 3.0
Size (QRNG)	20.0 × 20.0

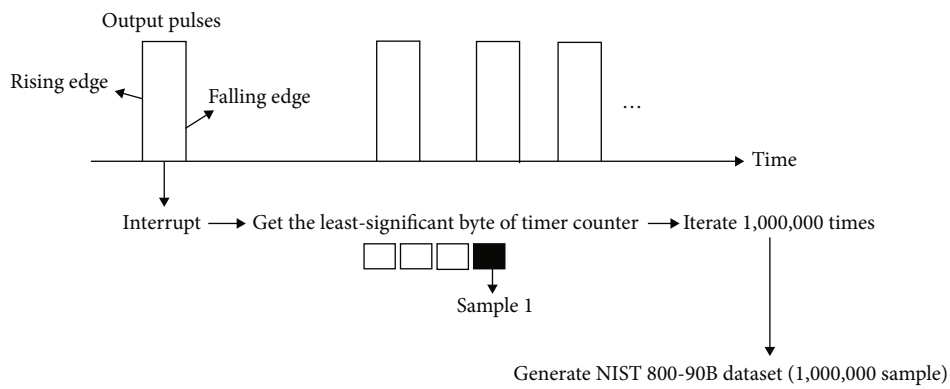


Figure 10: Process of sampling the quantum noise resource used in the experiment. The QRNG used in the experiment consists of a microcontroller in which the DRBG algorithm that generates random numbers by processing entropy is implemented with a device that generates quantum noise. We sampled the noise through communication with a microcontroller and noise generator.

of radioactive isotope decay, there is an additional advantage that the drone can perform its functions without being affected by the surrounding environment when flying because the effect of heat or pressure is insignificant [22].

4.2.2. Experiment and Analysis. To present the entropy value that can be expected when using the DIM equipped with the quantum noise source device proposed in this paper, entropy resources are collected from the device generating the quantum noise sources, the dataset is configured, and the entropy measurement test of NIST 800-90B presents the results.

The method of measuring entropy is based on several statistical tests. In other words, there are several ways to measure entropy, and the measurement result varies depending on which entropy measurement method is used. In the NIST 800-90B test performed in this study, entropy per collected single sample was measured based on Min-Entropy. This is a method of measuring the entropy of a sample from the worst distribution among the distributions obtained through statistical testing of a dataset that collects noise resource samples. Figure 9 shows the entropy evaluation process for entropy sources recommended by NIST 800-90B. Before the entropy evaluation process, permutation testing is performed on the dataset of the collected samples, and based on the results, it is determined whether the noise resources have independent and identically distributed (IID) characteristics. In this result, bit strings with IID characteristics are evaluated for entropy by a statistical method in the IID track, and for non-IID tracks, entropy is evaluated by a statistical method in the non-IID track. In this study, we used a C++-based tool distributed by NIST through GitHub for entropy evaluation of QRNG resources. Using this tool, it is possible to measure the entropy that can be obtained per sample of the noise resource collected from the QRNG constituting the collected dataset [18]. This can then be used as a basis for quantifying the amount of noise resources to sample to satisfy the required security strength when constructing a cryptographic system using QRNG. Table 4 lists the specifications of the QRNG and entropy collection device used for this experiment, and Figure 10 shows the process of collecting samples to perform the tests recommended by NIST 800-90B.

The QRNG used in the experiment consists of a chip that generates a noise resource and a microcontroller that receives the generated noise resource and executes DRBG, and it is configured to process the signal when noise occurs in the microcontroller. The type of noise resource used in QRNG utilizes the decay of radioactive isotopes, and 60–100 samples can be collected per second using the noise. We used the microcontroller's timer counter to collect samples of the noise source needed for the experiment. When the main function is executed, it is programmed to collect the lower 8 bits of the timer counter when the pulse generated by the noise generator is detected. The timer counter consists of an integer of 4 bytes and repeats the initialization process when the maximum value is reached. We judged that entropy characteristics can be expected in the lower 8 bits of the timer counter according to the generation of quantum noise. To collect 1,000,000 samples, which is the amount of data required by NIST SP 800-90B, the above process was iterated 1,000,000 times, and the configured dataset was input in binary format to the entropy measuring tool distributed by NIST.

The minimum entropy measured in the non-IID track was measured to be approximately 6.64 bits per sample of 8-bit size. This means that entropy can be obtained with an efficiency of approximately 83% per sample, and approximately 20 samplings are required to generate a security parameter with 128-bit security strength. The experimental results failed in the permutation test to determine whether it was IID, and it was determined that the dependency occurred between samples due to the repetition of the lower 8 bits of the timer counter, the sample responsible for this phenomenon.

Figure 11 shows the operation structure of the DIM using quantum noise resources. We previously demonstrated that they could be mounted on the DIM through the analysis of the physical specifications of quantum noise generators, and the results of the NIST SP 800-90B experiment indicate how much sampling is required when used in the DIM. As shown in Figure 11, the random number

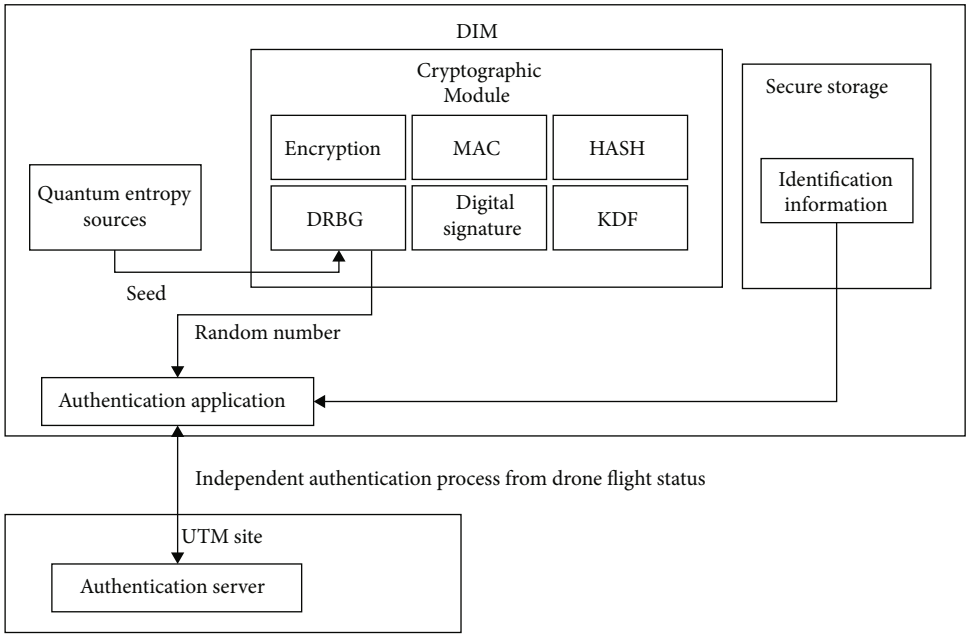


Figure 11: Operational structure of the DIM equipped with QRNG device. This figure shows the structure of a DIM equipped with a quantum noise resource generating device for constructing a cryptographic system. The DIM operating with quantum noise-generating devices can configure a seed to operate DRBG using a quantum noise resource and can perform the necessary cryptographic authentication in UTM using the random number output by DRBG. The important point here is that if the DIM structure is designed in this manner, independent cryptographic operation that is not affected by the flight status of the drone is possible.

generation method of the DIM using quantum noise resources can always satisfy the level of cryptographic random number security required by UTM. This advantage allows the Flight Permit Authority Server, which is required before flying a drone, to perform cryptographic operations independently of the drone's state in all scenarios of DIM in UTM, including cryptographic authentication, thereby securely performing the information security functions required by UTM.

5. Conclusion

In this study, we analyzed the noise resource generation method using the existing drone system and derived the limitations of the method based on the results. In addition, a method for generating random numbers in the DIM using quantum noise resources, which is a method to solve this limitation, was presented.

In the NIST SP 800-90B test performed in this study, entropy measurement through the IID track could not be performed due to permutation failure. In the entropy measurement process using a statistical method, the entropy measurement result may be different from the intended result depending on the noise resource postprocessing method. The microcontroller timer counter used in the entropy measurement experiment in this study has a size of 4 bytes, and the lower 8 bits are extracted and used to collect 8-bit samples. 8-bit has a short cycle period, which may have created a dependency between samples. In future work, we believe that entropy measurement in the IID track should also be performed by improving the postprocessing method.

UTM will become a major national infrastructure, and accordingly, information security devices to be used in UTM must perform data security using a cryptographic module whose safety has been proven through the Cryptographic Module Validation Process (CMVP). Therefore, the DIM is an information security device used in UTM and should be developed as a cryptographic module that can safely store identification information and perform independent cryptographic operation.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (Ministry of National Defense) (2022-0-00701, Development of Security Technology for Interworking between M-BcN and 5G Commercial Network).

References

[1] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communication*, vol. 38, no. 5, 2020.

[2] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *39th IEEE International Conference on Distributed Computing Systems*, Dallas, Texas, USA, 2019.

[3] Federal Aviation Administration, *Concept of Operations V2.0 – Unmanned Aircraft System (UAS) Traffic Management (UTM)*, 2020.

[4] UAS, *Traffic Management (UTM) – Part 8: Remote Identification*, ISO/CD 23629-8, Internation Organization for Standardization, Switzerland, 2022.

[5] ISO, *License and Drone Identity Module for Drone (Ultralight Vehicle or Unmanned aircraft system) – Part 2: Drone identity module (DIM)*, ISO/IEC AWI 22460-2, Internation Organization for Standardization, Switzerland, 2019.

[6] International Organization for Standardization, *Categorization and Classification of Civil Unmanned Aircraft Systems*, ISO 21895, Internation Organization for Standardization, Switzerland, 2020.

[7] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, USA, 5th edition, 2001.

[8] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, CRC Press, USA, 2nd edition, 2014.

[9] International Organization for Standardization, *Information Technology - Security Techniques–Security Requirements for Cryptographic Modules*, ISO/IEC 19790, Internation Organization for Standardization, Switzerland, 2012.

[10] National Institute of Standards and Technology, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, NIST SP800-90A Revision 1, National Institute of Standards and Technology, USA, 2015.

[11] National Institute of Standards and Technology, *Recommendation for the Entropy Sources Used for Random Bit Generation*, NIST SP800-90B, National Institute of Standards and Technology, USA, 2018.

[12] B. Schoenmakers, *Lecture Notes Cryptographic Protocols*, Department of Mathematics and Computer Science, Technical University of Eindhoven, Netherland, 2022.

[13] J. Park, S. Cho, T. Lim, and M. Tehranipoor, "QEC: a quantum entropy chip and its applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 6, pp. 1471–1484, 2020.

[14] T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, and A. Zeilinger, "A fast and compact quantum random number generator," *Review of Scientific Instruments.*, vol. 71, no. 4, pp. 1675–1680, 2000.

[15] M. Rohe, *RANDy - A True-Random Generator Based on Radioactive Decay*, 2003.

[16] K. Kim and Y. Kang, "Implentation of UAS identification and authentication on one M2M IoT platform," in *2019 International Conference on Information and Communication Technology Convergenece*, Jeju-si, Jesu-do, South Korea, 2019.

[17] S.-M. Cho, E. Hong, and S.-H. Seo, "Random number generator using sensors for drone," *IEEE Access*, vol. 8, pp. 30343–30354, 2020.

[18] A. Allouch, O. Cheikhrouhou, A. Koubaa, M. Khalgui, and T. Abbes, "MAVSec: securing the MAVLink protocol for Ardupilot/PX4 unmanned aeiral systems," http://arxiv.org/abs/1905.00265.

[19] International Organization for Standardization, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, NIST SP800-22, National Institute of Standards and Technology, USA, 2010.

[20] International Organization for Standardization, *IIdentification Cards-Physical Characteristics*, ISO/IEC 7810, Internation Organization for Standardization, Switzerland, 2019.

[21] European Telecommunications Standards Institute, *Smart Cards; UICC-Terminal interface; Physical and logical characteristics*, ETSI TS 102 221, European Telecommunications Standards Institute, France, 2013.

[22] E. D. Flakenberg, "Radioactive decay caused by neutrinos?," *Apeiron*, vol. 8, no. 2, 2001.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3678493, 10 pages
https://doi.org/10.1155/2022/3678493

Research Article

A Novel Intrusion Detection Method Based on Supplement Gate Recurrent Unit for IoT

Zi-yi Liu,[1,2] **Chang-song Yang**,[1,2] **Jun Xiao**,[1,2] **Bo-wen Song**,[1,2] **and Ke-xing Shi**[3]

[1]*Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China*
[2]*Guangxi Cooperative Innovation Centre of Cloud Computing and Big Data, Guilin University of Electronic Technology, Guilin 541004, China*
[3]*School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China*

Correspondence should be addressed to Chang-song Yang; csyang@guet.edu.cn

Received 7 April 2022; Revised 14 July 2022; Accepted 2 August 2022; Published 22 August 2022

Academic Editor: Ruinian Li

With the rapid development of information technology, the internet of things (IoT) technology has been integrated into most people's daily life and work. However, the IoT must confront many new security challenges. Specifically, the increase in the variety of IoT-connected devices has diversified the network. Meanwhile, the high data rates and spectral efficiency offered by 5G cellular networks facilitates the increasing capacity of IoT network traffic. Therefore, network traffic data are characterized by an expanded large scale, wide diversity, and high dimensions, which greatly affects the functionality and efficiency of intrusion detection methods. Although the existing neural network-based intrusion detection methods partially resolve the above problems, they need to execute a lot of nonlinear transformations when learning and characterizing data, resulting in a large loss of feature information. To address this problem, in this paper, we first design a new neural network model based on the gate recurrent unit (GRU), namely, the supplement gate recurrent unit (SGRU). Compared with a traditional GRU, through loss compensation, a SGRU can reduce the loss of feature information caused by nonlinear transformations when learning and characterizing network traffic data. Then, we adopt the SGRU to propose a novel intrusion detection method to monitor the security of the network. Finally, we developed the corresponding prototype system and verified its performance. The experimental results demonstrate that our proposed intrusion detection method is more accurate than previous intrusion detection methods.

1. Introduction

With the rapid development of information technology, especially the development of the internet of things (IoT) [1, 2] technology, IoT has gradually integrated with people's daily lives, causing significant changes to the traditional ordinary network environment. Different from the traditional ordinary network environment, the IoT is based on a wireless network, seamlessly connecting various objects into the network. This technology helps connect the network world with the physical world. However, the existing wireless network infrastructure is relatively simple, so no fixed self-protection mechanism exists. This leads to weak security of the IoT devices, which makes IoT security protection particularly important.

To protect network security, many network security protection methods are available, such as firewalls [3, 4], vulnerability scanning [5, 6], data encryption [7], and user authentication [8]. Although these methods can achieve security protection in traditional network environments, they are not perfectly suited for IoT network environments. The main reason is that in the IoT network environment, attackers may actively launch various types of attacks through system vulnerabilities, thus entering the computer system and stealing private information. In an effort to solve the aforementioned problems, intrusion detection

has gradually attracted extensive attention both in academia and industry [9, 10].

Intrusion detection is a network security software mechanism that can be used for monitoring of network traffic and provides alerts to network administrators when network traffic data are abnormal. Different from traditional network security protection methods with only passive defenses, intrusion detection is a proactive security protection technology. By deploying intrusion detection, network administrators can control the security threats faced by the IoT network systems in real time. Therefore, intrusion detection is one of the most important parts of network security protection.

Generally, the existing intrusion detection methods can be summarized as statistical-analysis-based intrusion detection methods, time-series-based intrusion detection methods, and machine-learning-based intrusion detection methods. However, in the IoT network environment, network traffic data usually present large-scale and high-dimensional characteristics. Additionally, network traffic data are often time-sequential, which brings unique challenges to existing intrusion detection methods. Specifically, on the one hand, because of high-dimensional and large-scale network traffic data, the existing statistical-based intrusion detection methods need to execute a large amount of calculations, resulting in low time-series-based intrusion detection method detection efficiency. On the other hand, the time-series-based intrusion detection methods only use time as an analysis factor and fail to consider other relevant factors. However, the network traffic data are characterized by randomness and diversity. Hence, the time-series-based intrusion detection methods cannot achieve accurate intrusion detection. Although traditional machine-learning-based intrusion detection methods can improve the efficiency of time-series-based intrusion detection method detection, they also suffer from low accuracy because of the randomness and diversity of the network traffic data.

Deep neural network (DNN) can learn the characteristics of complex and high-dimensional data effectively, which offers a new approach for the implementation of intrusion detection. As far as we know, the existing intrusion detection methods are mainly based on backpropagation neural networks (BPNN). However, they cannot work well for high-dimensional time series. Meanwhile, the existing recurrent neural network (RNN), such as the gated recurrent unit (GRU), often cause a loss of feature information due to the occurrence of many nonlinear transformations when learning and characterizing network traffic data.

1.1. Contributions. In this paper, we examine a practical and challenging problem, finding ways to increase the accuracy of intrusion detection in IoT network environments. Specifically, we design a new deep neural network model, namely, supplement gate recurrent unit (SGRU). Then, we apply the SGRU to design a novel intrusion detection method that can achieve efficient and accurate intrusion detection. Therefore, the three main contributions of this paper are summarized as follows:

(i) Most of the existing neural networks do not work well on high-dimensional time-sequential. Meanwhile, they need to perform a large number of nonlinear transformations, which leads to the problem of feature loss in characterization learning. To address the above problem, based on GRU, we design a new neural network model, namely, SGRU. Compared with the traditional GRU, SGRU can not only learns and characterizes data through the data's time-sequential, but also alleviates the loss of feature information caused by the nonlinear transformations

(ii) We design a SGRU-based intrusion detection method for the IoT network environment. Specifically, we utilize the SGRU to learn the characterization of network traffic data and give a theoretical basis. Our proposed SGRU-based intrusion detection method judges whether the network is in a secure state by analyzing the characteristics of network traffic data. Hence, the network administrators can accurately learn the security threats faced by information and networks systems, enabling them to take effective safeguard in time

(iii) We analyze the time complexity of our proposed SGRU-based intrusion detection method and three other different intrusion detection methods. It can be seen from the comparative analysis results that the time complexity of our proposed SGRU-based intrusion detection method is the same as that of the three other intrusion detection method. Moreover, a prototype system is developed, and a performance evaluation is provided. Compared with the existing intrusion detection methods, it can be seen that the intrusion detection method based on SGRU has a better effect

1.2. Related Work. Because of its prime performance, intrusion detection has been extensively studied both in industry and academia. Generally, the existing intrusion detection methods can be summarized as statistics-analysis-based intrusion detection methods [11], time-series-based intrusion detection methods [12, 13], and machine-learning-based intrusion detection methods [14, 15].

1.2.1. Statistics-Analysis-Based Intrusion Detection Methods. Gu et al. [16] developed a network traffic anomaly detection system that compared current baseline distributions with the entropy value of network traffic utilizations, which could effectively detect network anomalies, such as port scans and different types of synchronous attacks. Mazel et al. [17] introduced a method that combined interclass and subspace clustering result associations to achieve unsupervised network anomaly detections. Song and Liu [18] presented a dynamic k-nearest-neighbor (KNN) distance anomaly detection method based on cumulative storms. Compared with other methods, their method was more effective in anomaly detection. Mohammadi et al. [19] utilized a filter and wrapper to design a method for intrusion detection

using feature selection and clustering algorithm, which improved the intrusion detection performance. Zhou et al. [20] designed a new intrusion detection method, which is implemented by ensemble learning and feature selection techniques. In their method, a heuristic algorithm was used for dimensionality reduction. In addition, they utilized the voting technique and probability distribution of the base learner to identify attacks. Moustafa et al. [21] presented an integrated intrusion detection method to mitigate malicious events, which generated new statistical flow features from the protocol based on the analysis of the latent properties in the network. Unfortunately, the above methods require a lot of mathematical calculations. Therefore, the statistics-analysis-based intrusion detection methods are not efficient and accurate in the face of large-scale, multifeatured network traffic data.

1.2.2. Time-Series-Based Intrusion Detection Methods. Han and Zhang [22] used weighted self-similar parameters for detection in order to achieve network activity anomaly detection. Ye et al. [23] designed an anomaly detection method that was immune to nonstationary time series, which could achieve better evaluation performances by using the Hurst parameter estimation algorithm and the fractional Fourier transform (FRFT) algorithm. Yu et al. [24] improved the anomaly detection method using the autoregressive integrated moving average (ARIMA) model, which was improved for the imbalance and nonstationary characteristics unique to wireless sensor networks (WSNs). Pérez et al. [25] presented a new intrusion detection method. By combining time series analysis and multiplexed networks, their method could calculate the probability of an IP address being an attacker at a specific time. Abaeian et al. [26] designed a time series based intrusion detection method, which utilized the generalized autoregressive moving average (GARMA) method to study time series properties. To effectively reduce the false-positive rate, Bozdal et al. [27] proposed a wavelet-based method, which could localize the behavioral changes in the controller area network (CAN) traffic by analyzing the transmission patterns of a CAN network. However, the time-series-based intrusion detection method only uses time as an analysis factor, resulting in a low accuracy in intrusion detection.

1.2.3. Machine-Learning-Based Intrusion Detection Methods. Gu and Lu [28] presented a naive Bayesian (NB) feature embedding and a support vector machine– (SVM–) based intrusion detection method. Iwendi et al. [29] used the correlation-based feature selection approach to extract data features and then analyzed the dimensionality-reduced data through an integrated classifier, thereby constructing an intrusion detection system. Mittal et al. [30] used the low energy adaptive clustering hierarchy protocol for Levenberg-Marquardt neural networks (LEACH-LMNN) to analyze the network lifetime and the use of the gating mechanisms in wireless sensor networks. Through comparative experiments, it can be seen that this method has improved the detection accuracy. Xiao et al. [31] proposed a convolutional neural network– (CNN–) based intrusion detection method. They first used different dimensionality reduction methods to remove the redundant features of the network traffic data. Then, they utilized CNN to extract features from the data. Devan and Khare [32] designed an intrusion detection method that used XGBoost technology for feature selection and then utilized DNN to classify the network intrusions. Muhammad et al. [33] presented an intrusion detection method based on stacked autoencoders (SAE), which improved the classification accuracy. Imrana et al. [34] proposed an intrusion detection method based on bidirectional long-term and short-term memory (BiLSTM). Although the above methods improved the accuracy of the intrusion detections, they did not consider the loss of the feature information caused by the nonlinear changes in the neural network.

1.3. Organization. We introduce the work in the following Sections of this paper as follows. In Sections 2 and 3, we introduce the structure of GRU and SGRU, respectively. Then, we introduce the implementation of SGRU-based intrusion detection method in Section 4. Subsequently, we present a computational complexity comparison in Section 5. Next, we develop a prototype implementation of our proposed method and conduct comparative experiments in Section 6. Finally, we provide a brief and prospects for future work in Section 7.

2. Gate Recurrent Unit

In 2014, to address the ineffective transfer of long-term memory information and the gradient disappearance in backpropagation, Cho et al. designed a new recurrent neural network, namely, recurrent unit (GRU) [35]. Specifically, a GRU has two gate structure units, the reset gate R_t and update gate Z_t, as shown in Figure 1. The R_t gate is used to control the flow of the hidden state information from the previous moment in the current candidate set to the current moment set of the candidate hidden states. The Z_t gate is used to control how much unrelated content of the current candidate state needs to be forgotten at the previous moment and to determine how much of the current candidate set hidden state is retained.

As shown in Figure 1, in this paper, we use R_t to denote the reset gate and Z_t to denote the update gate. Then, the learning model of GRU can be described as follows.

First, in a GRU, the reset gate and update gate are determined by past information h_{t-1} and current information x_t. Then, the formulas are as follows:

$$R_t = \sigma(W_R \bullet [h_{t-1}, x_t]), \tag{1}$$

$$Z_t = \sigma(W_Z \bullet [h_{t-1}, x_t]). \tag{2}$$

Second, the candidate set of a GRU is controlled by the reset gate, and the formula can be expressed as follows:

$$\widetilde{h_t} = \tanh\,(W_{\tilde{h}} \bullet [R_t \times h_{t-1}, x_t]). \tag{3}$$

Third, in the update memory phase, a GRU updates h_t through the following formula:

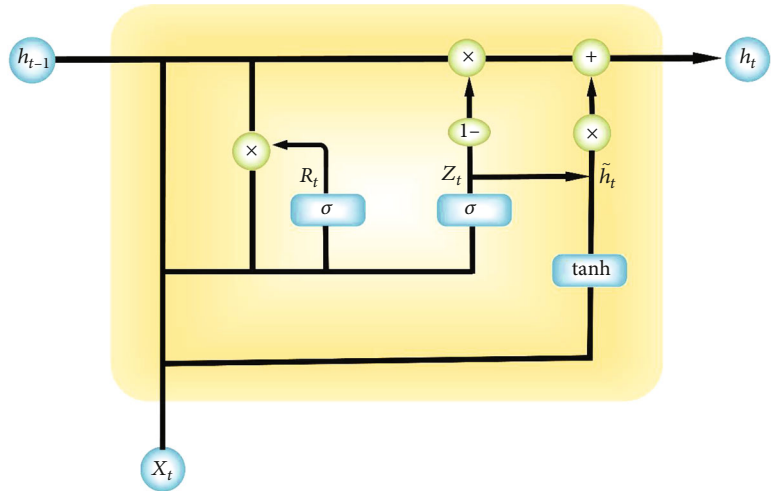


Figure 1: Structure of the GRU model.

$$h_t = (1 - Z_t) \times h_{t-1} + Z_t \times \widetilde{h_t}. \tag{4}$$

Finally, the output of the forward propagation is y_t, which can be computed through the following formula:

$$y_t = \text{softmax}(W_o \bullet h_t). \tag{5}$$

3. Supplement Gate Recurrent Unit

Generally, a GRU is capable of learning and characterization based on the temporal nature of the network traffic data. However, a GRU contains a large number of nonlinear transformations, which might lead to feature information loss in learning and characterizing the network traffic data. Therefore, we design a new neural network model based on GRU, namely, supplement gate recurrent unit (SGRU), as shown in Figure 2. Unlike the original GRU, the SGRU uses the loss compensation principle to alleviate the loss of the feature information when the SGRU executes nonlinear transformations. Therefore, the SGRU has more advantages in learning and characterizing of the IoT network traffic data.

As shown in Figure 2, the SGRU model consists of two GRUs, the OGRU and DGRU. In the SGRU, the OGRU is used for learning and characterizing the input data. Then, the DGRU is used to decode and restore the feature data after learning and characterizing by the OGRU. Without loss of generality, we use x_{input} to represent the input data, x_t to represent the characterizing data learned by the OGRU, x_{out} to represent the data restored by the DGRU, lc to represent the loss data, and lc_{out} to represent the loss compensation data. The specific implementation process of the SGRU is as follows.

First, we use the OGRU to perform the learning and characterization of the input data to obtain x_t. The specific formula is as follows:

$$x_t = OGRU\left(x_{input}\right). \tag{6}$$

Second, we use the DGRU to restore x_t to obtain x_{out}. Then, x_{input} minus x_{out} to obtain the loss data lc. The specific formulas are as follows:

$$\begin{aligned} x_{\text{out}} &= DGRU(x_t), \\ lc &= x_{input} - x_{\text{out}}. \end{aligned} \tag{7}$$

Finally, the loss data lc are subject to learning and characterization through the OGRU again to obtain the loss compensation data lc_{out}. Then, lc_{out} is added to x_t so that the loss of the feature information in x_t is supplemented, and the intrusion detection accuracy can be improved. The formulas are as follows:

$$\begin{aligned} lc_{\text{out}} &= OGRU(lc), \\ x_t &= x_t + lc_{\text{out.}} \end{aligned} \tag{8}$$

In addition, we also use the pseudocode in Algorithm 1 to describe the specific internal implementation process of the above SGRU.

In Algorithm 1, we first input the preprocessed x_{input} to the OGRU for learning and characterizing the x_t. Then, we input x_t into the DGRU for resotring to obtain x_{out}. Subsequently, we subtract the resotredx_{out} from the original input x_{input} and input the resulting loss data lc into the OGRU for learning and characterization to obtain lc_{out}. Finally, we supplement lc_{out} to x_t to obtain the output data x_t after the supplementary learning.

4. Our Proposed Method

In this section, we establish the system model of our proposed SGRU-based intrusion detection method and introduce the method.

4.1. System Model. The design of an accurate intrusion detection method is important for IoT network environments. In particular, with the widespread popularity of cloud

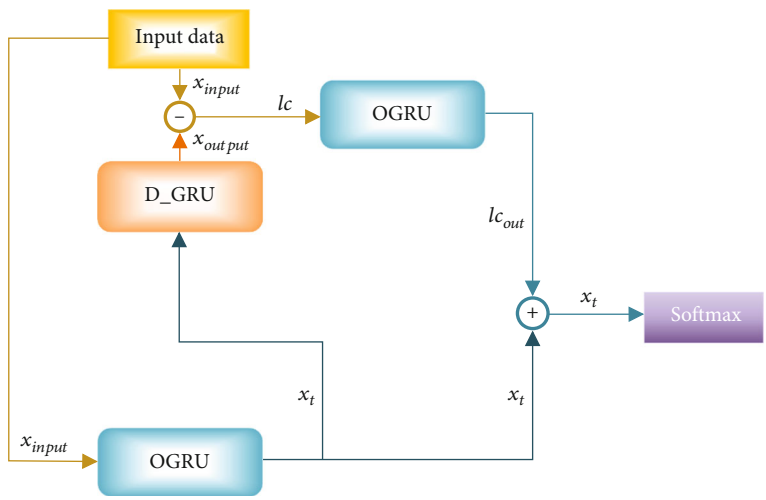


Figure 2: Structure of SGRU.

Input: x_{input}.
Output: x_t.
1 Preprocessing and other operations on x_{input} data.
2 $x_t \leftarrow OGRU(x_{input})$
5 $x_{\text{out}} \leftarrow DGRU(x_t)$
6 $lc \leftarrow (x_{input} - x_{\text{out}})$
7 $lc_{\text{out}} \leftarrow OGRU(lc)$
8 **return** $x_t \leftarrow (x_t + lc_{\text{out}})$

Algorithm 1: SGRU method implementation.

computing [36, 37], IoT [38, 39], and wireless networks [40–42], network traffic data are instilled with the characteristics of diversity, series timing, randomness, and high dimensionality, which creates many new problems for the existing intrusion detection methods. Specifically, it directly affects the accuracy and universality of the intrusion detection methods. Moreover, there are a large number of nonlinear transformations in neural networks. Hence, when the network traffic data are learned and characterized, many features are lost. To solve the above problems, we propose a SGRU-based intrusion detection method, whose system model is shown in Figure 3.

In our proposed intrusion detection method, we first design a new recurrent neural network, namely, SGRU. Then, we adopt SGRU to build a new intrusion detection method. Compared with other DNNs, a SGRU not only learns and characterizes network traffic data through time-sequential but also uses the loss compensation mechanism to reduce the feature loss caused by a large number of nonlinear transformations. As a result, the performance of our proposed SGRU-based intrusion detection method is more attractive.

4.2. SGRU-Based Intrusion Detection Method. We use Algorithm 2 to introduce our proposed SGRU-based intrusion detection method in detail.

In Algorithm 2, we utilize $X - train$ as the network data training set, $Y - test$ as the network data test set, R_{label} as the real attack label, n as the training epoch, S as the intrusion detection value, and R to represent the comparison result.

In the above Algorithm 2, we first input $X - train$ into the SGRU model for the training of the SGRU model. Then, the trained SGRU model is obtained through n epoch of training. Subsequently, $Y - test$ is input into the SGRU to obtain the corresponding detection result S. Finally, we compare S with the true label R_{label} and get the comparison result R.

5. Computational Complexity Analysis

We compare the time complexity of our proposed SGRU-based intrusion detection method with GRU, BiLSTM, and SAE-BPNN-based intrusion detection methods in this section and show them using Table 1.

For simplicity, we use m as the input dimension and n as the dimension of the hidden layer. To facilitate the calculation of the time complexity of SGRU, we first calculate the time complexity of GRU. From Formulas (1)-(4) presented in Section 2, we find that for the GRU, the total operations time overhead is $T(3 \times n \times m + 6 \times n^2 + 4 \times n)$, so the time complexity of GRU can be described as $O(\ n^2)$. Subsequently, for the SGRU, the total operations time overhead is $T(3 \times (3 \times n \times m + 6 \times n^2 + 4 \times n) + 2 \times n + 2 \times m)$, so the time complexity of SGRU is the same as that of GRU, which is $O(n^2)$. For LSTM, the total operations time overhead is $T(4 \times n \times m + 7 \times n^2 + 4 \times n,)$ and the time complexity can be expressed as $O(n^2)$. Meanwhile, the total operations time overhead of BiLSTM is $T(2 \times (4 \times n \times m + 7 \times n^2 + 4 \times n))$, and the time complexity can be expressed as $O(n^2)$. Moreover, since both SAE and BPNN are two-layer fully connected layer structures in our reproduction experiments, the total operations time overhead is $T(3 \times n \times m + 3 \times n^2)$, and the time complexity is $O(n^2)$.

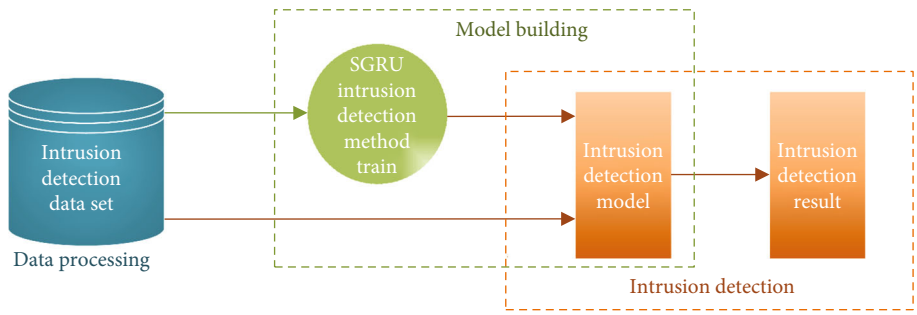


Figure 3: System model.

Input: $X - train$, $Y - test$, R_{label}, n-number of epoch.
Output: intrusion detection results R.
Initialize the network dataset.
for $i = 0$ **to** n **do**
$SGRU \leftarrow SGRU_i(X - train)$
end for
$S \leftarrow SGRU(Y - test)$
return $R \leftarrow Compare(S, R_{label})$

Algorithm 2: SGRU-based intrusion detection method.

6. Experimental Settings

In this section, we first describe the environment, datasets, and experimental standards required for the comparative experiments. Then, we give the results and analysis of the comparative experiments.

6.1. Experimental Environment and Dataset. The desktop hardware devices we used in this experiment mainly include AMD Ryzen 5 3500X CPU, 16G of main memory, and NVIDIA RTX 2060S graphics card. At the same time, this desktop also has Windows10 system, cuDNN7.4.2, and CUDA10.0 driver.

In the experiments of this paper, we use the public dataset UNSW-15NB as the experimental data [43, 44]. The UNSW-15NB dataset includes 2,540,044 pieces of data and 9 different anomaly types. The anomaly types are specifically described as follows:

(1) Analysis: web pages are hacked using tools such as network ports and scripts

(2) Backdoors: a method of attacking through holes in computer reservations or defenses

(3) DoS: using a large-scale traffic attack on the attacker will exhaust the computing power of the computer and make various computer services unusable

(4) Exploits: a means of attacking through vulnerabilities in the attacker's computer system

(5) Fuzzers: an attack method that paralyzes the victim's system by sending a large number of random numbers

(6) Generic: a method suitable for attacking block ciphers

(7) Reconnaissance: an attack that uses probing to gather information about an attack target

(8) Shellcode: an attack method that uses Shell commands to control the victim's host

(9) Worms: the self-replication method increases the computing overhead of the victim's computer, resulting in low computer efficiency and inability to work properly

For simplicity, we randomly intercept 550,000 pieces of data as experimental data, of which 50,000 are used as the test set and 500,000 are used as the training set.

6.2. Experimental Criteria. In the simulation experiments, *Accuracy*, *Precision*, *Recall*, *F1_score*, and *FRR* are used to verify the effectiveness of our proposed SGRU-based intrusion detection method.

True positive (TP): The actual is positive, and the detection result is also positive

False positive (FP): The actual is negative, while the detection result is positive

True negative (TN): The actual is negative, and the detection result is also negative

False negative (FN): The actual is positive, while the detection result is negative

Accuracy. The proportion of samples that are correctly detected in the total sample and the calculation formula is as follows:

$$Accuracy = \frac{TN + TP}{TP + TN + FP + FN}. \tag{9}$$

Precision. The proportion of samples that are detected as positive and turn out to be positive and the calculation formula is as follows:

$$Precision = \frac{TP}{FP + TP}. \tag{10}$$

Recall. The proportion of samples that were detected as positive among the samples were actually positive, and the calculation formula is as follows:

Table 1: Time complexity.

	Total operations time overhead	Time complexity
SGRU	$T(9 \times n \times m + 18 \times n^2 + 14 \times n + 2 \times m)$	$O(n^2)$
GRU [35]	$T(3 \times n \times m + 6 \times n^2 + 4 \times n)$	$O(n^2)$
BiLSTM [34]	$T(8 \times n \times m + 14 \times n^2 + 8 \times n)$	$O(n^2)$
SAE-BPNN [33]	$T(3 \times n \times m + 3 \times n^2)$	$O(n^2)$

$$Recall = \frac{TP}{FN + TP}. \tag{11}$$

F1_score. After a comprehensive evaluation of the intrusion detection results using the *Recall* and *Precision* indicators, the calculation formula is as follows:

$$F1_{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{12}$$

FRR. The proportion of samples that detected negative results among the samples were actually positive, and the calculation formula is as follows:

$$FRR = \frac{FN}{FN + TP}. \tag{13}$$

6.3. Experimental Results and Analysis. We implement our proposed intrusion detection method in this section and compare the method with the experimental results of intrusion detection methods based on GRU [35], BiLSTM [34], and SAE-BPNN [33] analyzed.

6.3.1. Effectiveness Evaluation. We use *Accuracy*, *Precision*, *Recall*, $F1_{score}$, and *FRR* metrics to compare the effectiveness of our proposed method with GRU-based intrusion detection methods, BiLSTM-based intrusion detection methods, and SAE-BPNN-based intrusion detection methods. Figure 4 shows the comparison of the four metrics, *Accuracy*, *Precision*, *Recall*, and *F1_score*.

Figure 4 demonstrates the performance results of four different intrusion detection methods from different perspectives using four different metrics. It is not difficult to see that our proposed SGRU-based intrusion detection method performs better in all aspects compared to the other three methods. Thus, network administrators can more accurately grasp the current network security status, thus greatly enhancing the effectiveness of network security protection.

It can been seen from Figure 5 that our proposed SGRU-based intrusion detection method has lower *FRR* than the other three methods. That is, our proposed method can provide network administrators with less *FRR*. Therefore, our proposed method can reduce the waste of network resources caused by *FRR* while improving the protection performance of the intrusion detection. The reason is that our proposed method considers the loss of the feature information caused by a large number of nonlinear transformations and alleviates this problem by means of a loss compensation.

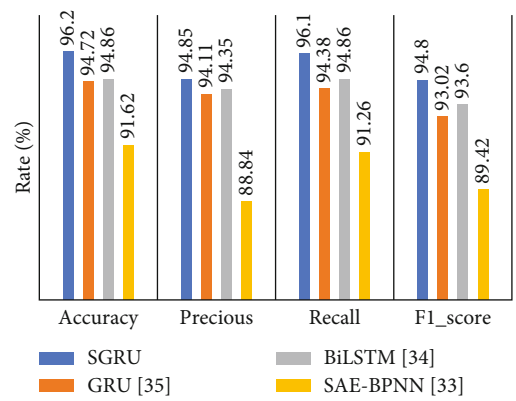


Figure 4: Comparison of effectiveness.

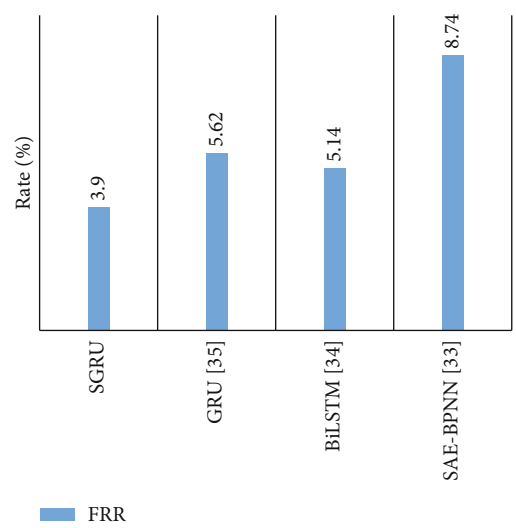


Figure 5: Comparison of *FRR*.

6.3.2. Efficiency Analysis. In this part of the section, we provide an efficiency comparison, as presented in Figures 6 and 7. Figures 6 and 7 show the total time overhead and the time overhead of the test, respectively. We can see that although our proposed method has a higher overhead in time compared to the other three intrusion detection methods, but compared with the improvement in accuracy of our proposed intrusion detection method, this part of the time overhead is acceptable. The experimental results are consistent with the time complexity analysis in Section 5.

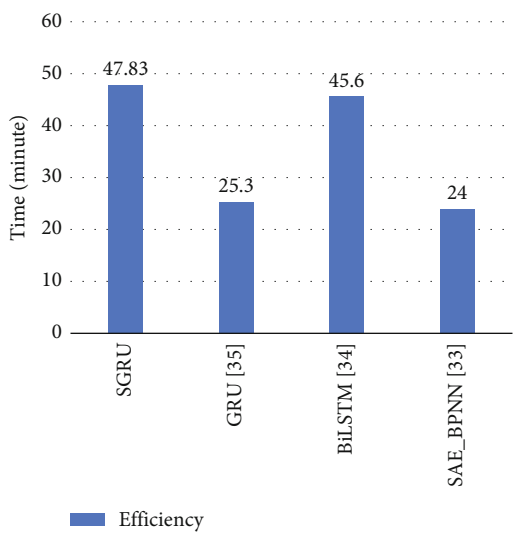


Figure 6: Total time overhead comparison.

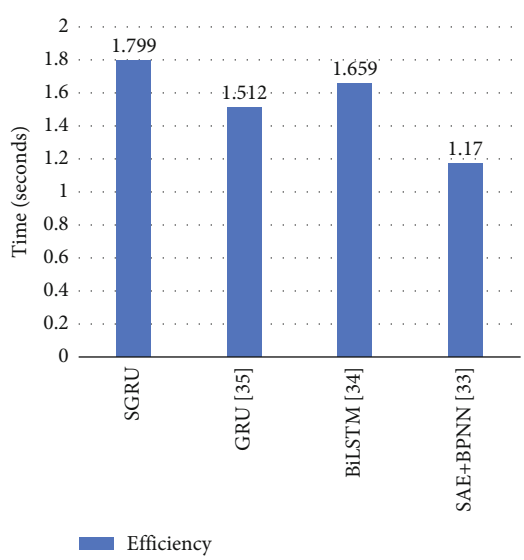


Figure 7: Efficiency comparison of the test.

7. Conclusions and Future Work

In this paper, we studied the design of an accurate intrusion detection method for the IoT network environment. First, we proposed a neural network model named SGRU by improving GRU. Then, we utilized the SGRU to propose a novel intrusion detection method. This method could greatly improve the effectiveness of intrusion detection. Finally, we used simulation experiments to implement our proposed SGRU-based intrusion detection method and evaluated the detection performance. The experimental results showed that compared with some existing intrusion detection methods, our proposed SGRU-based intrusion detection method could achieve a substantial improvement in effectiveness and accuracy.

In the future, we plan to conduct further research on intrusion detection. For example, we will explore the possibility of improving the efficiency of intrusion detection by proposing a simpler neural network structure.

Data Availability

The dataset UNSW-NB15 can be downloaded from https://research.unsw.edu.au/projects/unsw-nb15-dataset.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Science and Technology Program of Guangxi grant (AD20297028), the Guangxi Key Laboratory of Cryptography and Information Security grant (GCIS202128), and the Natural Science Foundation of Guangxi grant (2020GXNSFBA297132).

References

[1] J. Azar, A. Makhoul, R. Couturier, and J. Demerjian, "Robust IoT time series classification with data compression and deep learning," *Neurocomputing*, vol. 398, pp. 222–234, 2020.

[2] K. N. Qureshi, O. Kaiwartya, G. Jeon, and F. Piccialli, "Neurocomputing for internet of things: object recognition and detection strategy," *Neurocomputing*, vol. 485, pp. 263–273, 2022.

[3] S. Jingyao, S. Chandel, Y. Yunnan, Z. Jingji, and Z. Zhipeng, "Securing a network: how effective using firewalls and VPNs are?," in *Advances in Information and Communication. FICC 2019*, K. Arai and R. Bhatia, Eds., vol. 70 of Lecture Notes in Networks and Systems, Springer, Cham, 2019.

[4] V. H. Dixit, S. Kyung, Z. Zhao, A. Doupé, Y. Shoshitaishvili, and G.-J. Ahn, "Challenges and preparedness of SDN-based firewalls," in *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, pp. 33–38, Tempe, Arizona, USA, 2018.

[5] D. Esposito, M. Rennhard, L. Ruf, and A. Wagner, "Exploiting the potential of web application vulnerability scanning," in *ICIMP 2018 the Thirteenth International Conference on Internet Monitoring and Protection*, pp. 22–29, Barcelona, Spain, 2018.

[6] N. Schagen, K. Koning, H. Bos, and C. Giuffrida, "Towards Automated Vulnerability Scanning of Network Servers," in *Proceedings of the 11th European Workshop on Systems Security*, pp. 1–6, Porto, Portugal, 2018.

[7] M. H. Saracevic, S. Z. Adamovic, V. A. Miskovic et al., "Data encryption for internet of things applications based on catalan objects and two combinatorial structures," *IEEE Transactions on Reliability*, vol. 70, no. 2, pp. 819–830, 2021.

[8] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, and K. Salah, "A user authentication scheme of IoT devices using blockchain-enabled fog nodes," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–8, Aqaba, Jordan, 2018.

[9] C. Iwendi, J. H. Anajemba, C. Biamba, and D. Ngabo, "Security of things intrusion detection system for smart healthcare," *Electronics*, vol. 10, no. 12, article 1375, 2021.

[10] M. T. Nguyen and K. Kim, "Genetic convolutional neural network for intrusion detection systems," *Future Generation Computer Systems*, vol. 113, pp. 418–427, 2020.

[11] J. Aitchison, "The statistical analysis of compositional data," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 44, no. 2, pp. 139–160, 1982.

[12] C. W. J. Granger and M. J. Morris, "Time series modelling and interpretation," *Journal of the Royal Statistical Society. Series A (General)*, vol. 139, no. 2, pp. 246–257, 1976.

[13] D. S. Broomhead and R. Jones, "Time-series analysis," *Proceedings of the Royal Society of London A Mathematical and Physical Sciences*, vol. 423, no. 1864, pp. 103–121, 1989.

[14] M. I. Jordan and T. M. Mitchell, "Machine learning: trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[15] T. G. Dietterich, "Machine-learning research," *AI Magazine*, vol. 18, no. 4, pp. 97–97, 1997.

[16] Y. Gu, A. McCallum, and D. Towsley, "Detecting anomalies in network traffic using maximum entropy estimation," in *Proceedings of the 5th ACM SIGCOMM conference on Internet measurement - IMC '05*, p. 32, California, USA, 2005.

[17] J. Mazel, P. Casas, Y. Labit, and P. Owezarski, "Sub-space clustering, inter-clustering results association & anomaly correlation for unsupervised network anomaly detection," in *2011 7th International Conference on Network and Service Management*, pp. 1–8, Paris, France, October 2011.

[18] R. Song and F. Liu, "Real-time anomaly traffic monitoring based on dynamic k-NN cumulative-distance abnormal detection algorithm," in *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems*, pp. 187–192, Shenzhen, China, November 2014.

[19] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *Journal of Information Security and Applications*, vol. 44, pp. 80–88, 2019.

[20] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 174, article 107247, 2020.

[21] N. Moustafa, B. Turnbull, and K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2018.

[22] J. Han and J. Z. Zhang, "Network traffic anomaly detection using weighted self-similarity based on EMD," in *2013 Proceedings of IEEE Southeastcon*, pp. 1–5, Jacksonville, USA, 2013.

[23] X. Ye, J. Lan, and W. Huang, "Network traffic anomaly detection based on self-similarity using FRFT," in *2013 IEEE 4th International Conference on Software Engineering and Service Science*, pp. 837–840, Beijing, China, 2013.

[24] Q. Yu, L. Jibin, and L. Jiang, "An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2016, Article ID 9653230, 2016.

[25] S. I. Pérez, S. Moral-Rubio, and R. Criado, "A new approach to combine multiplex networks and time series attributes: building intrusion detection systems (IDS) in cybersecurity," *Chaos, Solitons & Fractals*, vol. 150, article 111143, 2021.

[26] V. Abaeian, A. Abdullah, T. Pillai, and L. Cai, "Intrusion detection forecasting using time series for improving cyber defence," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 3, no. 1, pp. 28–33, 2015.

[27] M. Bozdal, M. Samie, and I. K. Jennions, "WINDS: a wavelet-based intrusion detection system for controller area network (CAN)," *IEEE Access*, vol. 9, pp. 58621–58633, 2021.

[28] J. Gu and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," *Computers & Security*, vol. 103, article 102158, 2021.

[29] C. Iwendi, S. Khan, J. H. Anajemba, M. Mittal, M. Alenezi, and M. Alazab, "The use of ensemble models for multiple class and binary class classification for improving intrusion detection systems," *Sensors*, vol. 20, no. 9, p. 2559, 2020.

[30] M. Mittal, C. Iwendi, S. Khan, and A. R. Javed, "Analysis of security and energy efficiency for shortest route discovery in low-energy adaptive clustering hierarchy protocol using Levenberg-Marquardt neural network and gated recurrent unit for intrusion detection system," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 6, article 3997, 2021.

[31] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42210–42219, 2019.

[32] P. Devan and N. Khare, "An efficient XGBoost–DNN-based classification model for network intrusion detection system," *Neural Computing and Applications*, vol. 32, no. 16, pp. 12499–12514, 2020.

[33] G. Muhammad, M. S. Hossain, and S. Garg, "Stacked autoencoder-based intrusion detection system to combat financial fraudulent," *IEEE Internet of Things Journal*, 2020.

[34] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Systems with Applications*, vol. 185, article 115524, 2021.

[35] K. Cho, M. B. Van, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[36] C. Yang, F. Zhao, X. Tao, and Y. Wang, "Publicly verifiable outsourced data migration scheme supporting efficient integrity checking," *Journal of Network and Computer Applications*, vol. 192, article 103184, 2021.

[37] C. Yang, X. Tao, F. Zhao, and Y. Wang, "Secure data transfer and deletion from counting bloom filter in cloud computing," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 273–280, 2020.

[38] E. Tanghatari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Distributing DNN training over IoT edge devices based on transfer learning," *Neurocomputing*, vol. 467, pp. 56–65, 2022.

[39] Y. Lu, S. Wu, Z. Fang, N. Xiong, S. Yoon, and D. S. Park, "Exploring finger vein based personal authentication for secure IoT," *Future Generation Computer Systems*, vol. 77, pp. 149–160, 2017.

[40] W. Guo, N. Xiong, H. C. Chao, S. Hussain, and G. Chen, "Design and analysis of self-adapted task scheduling strategies in wireless sensor networks," *Sensors*, vol. 11, no. 7, pp. 6533–6554, 2011.

[41] X. Wang, Q. Li, N. Xiong, and Y. Pan, "Ant colony optimization-based location-aware routing for wireless sensor networks," in *Wireless Algorithms, Systems, and Applications. WASA 2008*, Y. Li, D. T. Huynh, S. K. Das, and D. Z. Du, Eds., vol. 5258 of Lecture Notes in Computer Science, pp. 109–120, Springer, Berlin, Heidelberg, 2008.

[42] R. Wan, N. Xiong, and N. T. Loc, "An energy-efficient sleep scheduling mechanism with similarity measure for wireless sensor networks," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, pp. 1–22, 2018.

[43] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, Canberra, Australia, November 2015.

[44] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18–31, 2016.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3173995, 8 pages
https://doi.org/10.1155/2022/3173995

Research Article

A Hash-Based Fast Image Encryption Algorithm

Ruifeng Han

Computer Science Department, Xinzhou Teachers University, Xinzhou 034000, China

Correspondence should be addressed to Ruifeng Han; hrf_xztu@163.com

Received 1 March 2022; Accepted 13 July 2022; Published 10 August 2022

Academic Editor: Ruinian Li

Many specialists and academics have recently become interested in the security of digital images in applications for the Internet of Things. Hash-based digital image encryption algorithms with high unified average changing intensity (UACI > 30.96 percent) and only one pixel difference from the plain image would therefore adjust plenty of the pixels in the cipher image and have indeed been suggested to maintain the protection of images in the Internet of Things (NPCR > 98.77 percent). Theoretical study and simulation results show that the suggested approach can fix these issues while retaining all the advantages of the original. The proposed image encryption algorithm has important application value for strengthening the security of the Internet of Things.

1. Introduction

With the continuous development of information technology and computer processing power, cryptography has also been extended. Because the network environment, especially Internet of Things environment is vulnerable to attacks, and digital images contain redundant information, which is closely related to personal privacy, once it is leaked, it is likely to be personally threatened and even affect commercial secrets and national security. Many algorithms exclusively handle with digital encrypted images or text encryption, respectively. Several algorithms, nevertheless, integrate these two tasks. Passwords are simpler and easier for people to recognize than pseudorandom numbers or a long string of codes, making it more practical and appropriate to password-protect digital photographs [1]. Liu and Tan put forth a plan that uses 1D SHA-2 algorithms combined with password protection to encrypt digital images with compound forward transform [2].

2. Defect Analysis of the Original Algorithm

2.1. Lack of Connectivity between Pixels during Encryption. During the encryption of the original image, the key stream generation step and there really is no relationship either between pixel intensity, and indeed, the postprocessing phase solely affects each pixel inside a one-to-one relationship. This characteristic may expose it to selected attacks [3]. An attacker can do this by encrypting only two images that differ by only one pixel. The plain picture and the cipher image could both be found by an opponent because the two encrypted images just vary with one pixel from one another. Figure 1 illustrates an illustration of a 16×16 image. Following encrypted data, point (1, 2) in the original image is transferred to point (12, 5). An opponent can find the permutation rules for a pixel by performing this method again for each pixel in the original image. This is the same as disclosing a fresh random map during postprocessing [4].

2.2. The Only Dependency of the Key Stream. We discover that the participant's password is completely dependent on the secret key during the encryption step of the original technique. The associated stream usually stays the same as long as the cipher is unmodified, regardless of whether this is used mostly for XOR with the original picture or for a postprocessing step. A known-plaintext assault on the original algorithm is hence inevitable. The following formulas are used to express the random key $\{K1, K2, \cdots, KM \times N\}$, KMN that the adversary could retrieve if they are given a pair of pure image needs to set $G = \{G1, G2, \cdots, GM \times N\}$ and a placed of cipher maps $C = \{C1, C2, \cdots, CM \times N\}$.

$$K_i = C_i \oplus G_i, \quad (1 < i \leq 8 \times M \times N). \tag{1}$$

The permutation rules for pixels are recovered when

Figure 1: Mapping process.

used with the chosen-plaintext attack previously discussed. Understanding the key stream produced by a specific key is identical to actually having that key. The equivalent normal image may be retrieved instantly for whichever cipher image encoded with almost the same key.

2.3. Attack Simulation. Let us say there is a password image but no one knows the key. An attacker could trick the encryption system by feeding it a black image (every pixel inside the encrypted image equals 0) to get the cipher image B. B is definitely a different key stream. We could derive the replacement rule, designated by M, using the procedure outlined in Section 2.1. Apply rule M on the replaced key stream to undo it. We could create a new key stream that seems to be identical to the key stream obtained by doing an XOR operation with the original image prior to replacement. There seem to be two procedures required to obtain the original image for encrypted images. Flip the encrypted image using rule M firstly and afterwards XOR the new key stream after that. The outcome is the accurate normal picture. Throughout Figure 2, the attack procedure is depicted.

3. Improved Algorithm

In order to overcome the above potential defects, the following improved algorithm is proposed:

(1) For convenience, the initial image post-processing step is modified by us to the following equation, defined as ACM; this includes the three variables a, b, and k. To create the three variables that will substitute the efficient algorithm, we utilize $Ki = H(H($ Key$)$. In our enhanced algorithm, however, k rounds of operations are required in the encryption/decryption stage. On each turn, swap positions between (0,0) and (1,1). Each round takes turns.

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} ab+1 & a \\ b & 1 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} (\text{mod } 1), \quad a, b \in N \tag{2}$$

(2) The improved algorithm is performed by splitting the image into two equal pieces and then independently encrypting each piece. The key stream is determined by the user key and original image features [5, 6]. In other words, when the same key is used, different images are encrypted with the key stream as a variable; it aids in preventing selected and known-plaintext attacks on the encryption algorithm

3.1. Encryption

(1) Identical to steps one and two of the initial step

(2) Generate a new key stream K_0 with image features. First, the original image is divided into two equal parts L_0 and R_0 in the vertical direction, and the two parts are encrypted, respectively, and the formula is as follows. First, encrypt the information of R_0 using L_0 to R_1; the right part has been encrypted. Then, L_0 is encrypted to L_1 using R1's information

(3) Finally, to acquire the cipher picture, combine the 2 encrypted components L_1 and R_1

$$\begin{cases} R_1 = \text{Complex}\left(\text{imresize}\left(\left(K' \oplus g_{L_0}\right), \left[M, \frac{N}{2}\right], 'nearest'\right)\right) \oplus R_0, \\ L_1 = \text{Complex}\left(\text{imresize}\left(\left(K' \oplus g_{R_1}\right), \left[M, \frac{N}{2}\right], 'nearest'\right)\right) \oplus L_0. \end{cases} \tag{3}$$

L_0 and R_0 throughout the example above stand for the left and right sides of something like the pure picture, while L_1 and R_1 stand for the left and right sides of the cipher image, g_{L_0} and g_{R_1} represent the average gray value of L_0 and R_1 which is of variable size, (•, [M, $N/2$], "nearest") represents the expansion operation, the "nearest" method is used to the process of enlarging an image to M rows, and $N/2$ column is known as cosine similarity interpolation. The number of the pixel to which this pertinent is allocated to the output pixel; additional pixels are not taken into account. The compound transformation in the initial encryption is represented by complex(•)

(4) The postprocessing step is ACM

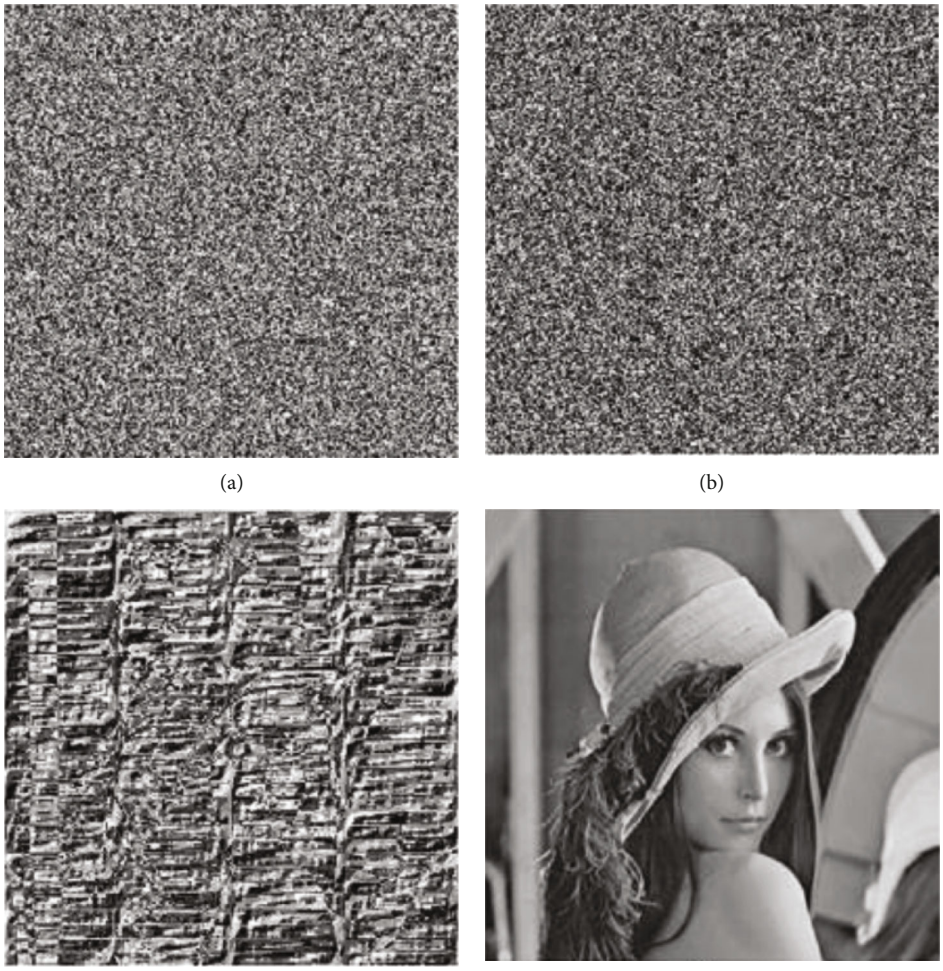
(5) Encrypted images

Figure 3 depicts the encryption process algorithm. Figure 4 depicts the key stream creation algorithm.

3.2. Decryption. To obtain the XOR key stream, firstly invert the ACM.

Encrypt the XOR key stream to produce a plain picture next. L_1 to L_0 first were encrypted using R_1's data. Next, as stated in the following expression, utilize the data from L_0 to decode R_1 to R_0. L_0 and R_0 are the left and right halves of the plain image, whereas L_1 and R_1 are the left and right halves of the encrypted image.

$$\begin{cases} L_0 = L_1 \oplus \text{Complex}\left(\text{imresize}\left(\left(K' \oplus g_{R_1}\right), \left[M, \frac{N}{2}\right], 'nearest'\right)\right), \\ R_0 = R_1 \oplus \text{Complex}\left(\text{imresize}\left(\left(K' \oplus g_{L_0}\right), \left[M, \frac{N}{2}\right], 'nearest'\right)\right). \end{cases} \tag{4}$$



(a) (b) (c) (d)

Figure 2: Attack process.

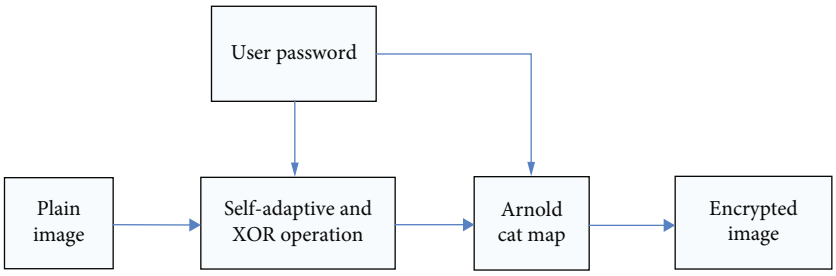


Figure 3: Encryption process algorithm.

Finally, the two decrypted parts L_0 and R_0 are connected together to obtain a decrypted image.

4. Performance Analysis

In this section, a 256×256 Lena gray image [7] is selected to conduct comparative experiments.

All experiments were performed using MATLAB 7.9 on a personal computer (PC) which has a 250 GB hard drive, a 2.0 GHz Intel dual-core microprocessor, and 1.99 GB of storage [8].

4.1. Histogram of Encrypted Image. Every gray level's frequency is shown in the graph, and that every gray level mainly related to the digital image [9]. The histograms of the original and encrypted images are seen in Figure 5.

The graph of the password picture is quite homogeneous and distinct from the actual picture, as shown in the figure.

4.2. Correlation of Two Adjacent Pixels. The correlation of test image pixels includes horizontal correlation and diagonal correlation: first, in the picture, 2500 pairs of adjacent

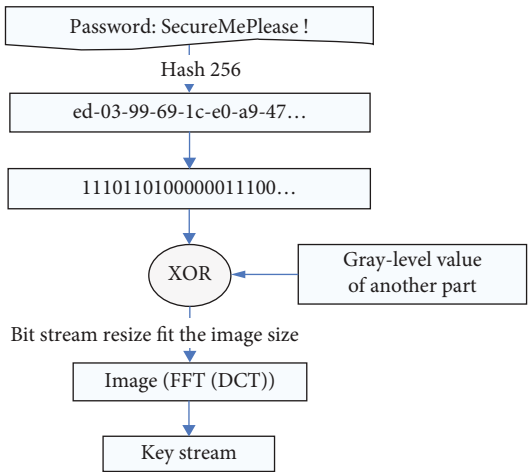
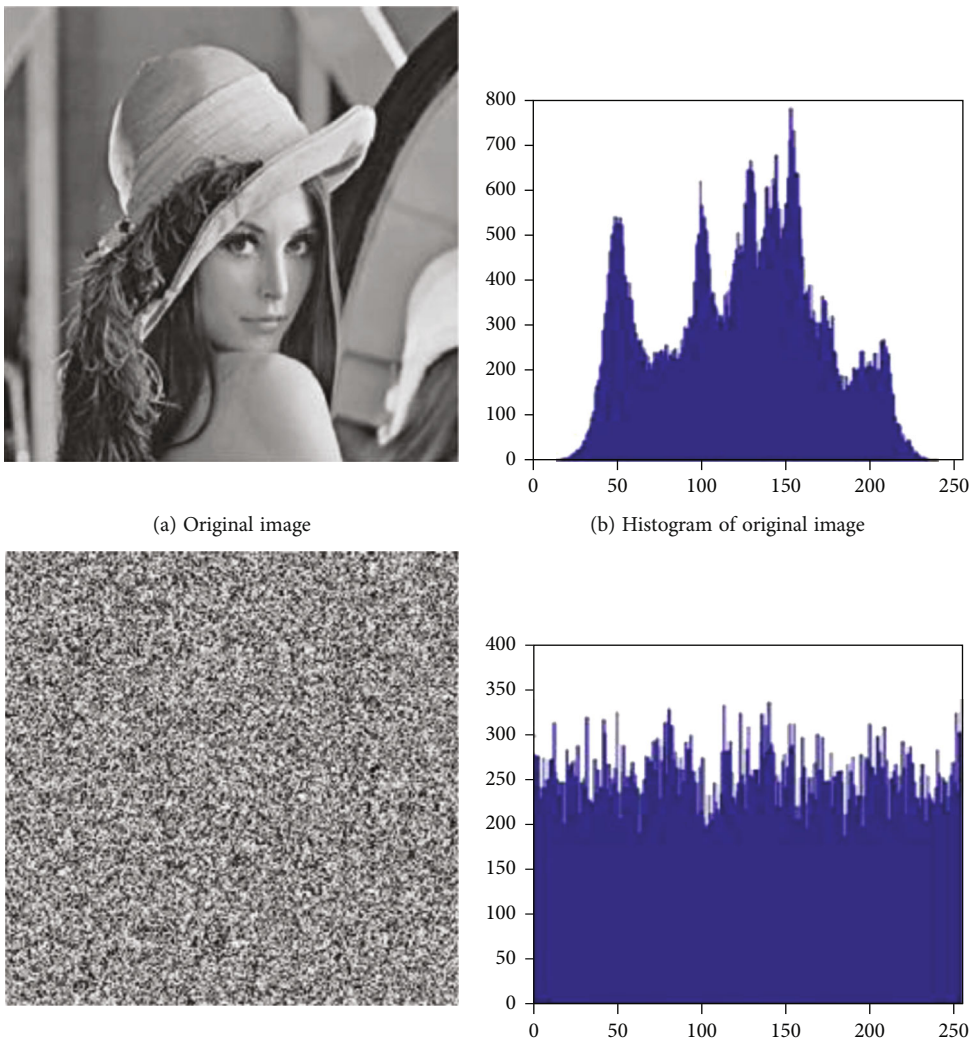


Figure 4: Key stream generation algorithm.



(a) Original image

(b) Histogram of original image

(c) Encrypted image

(d) Histogram of encrypted image

Figure 5: Histogram of original image and password image.

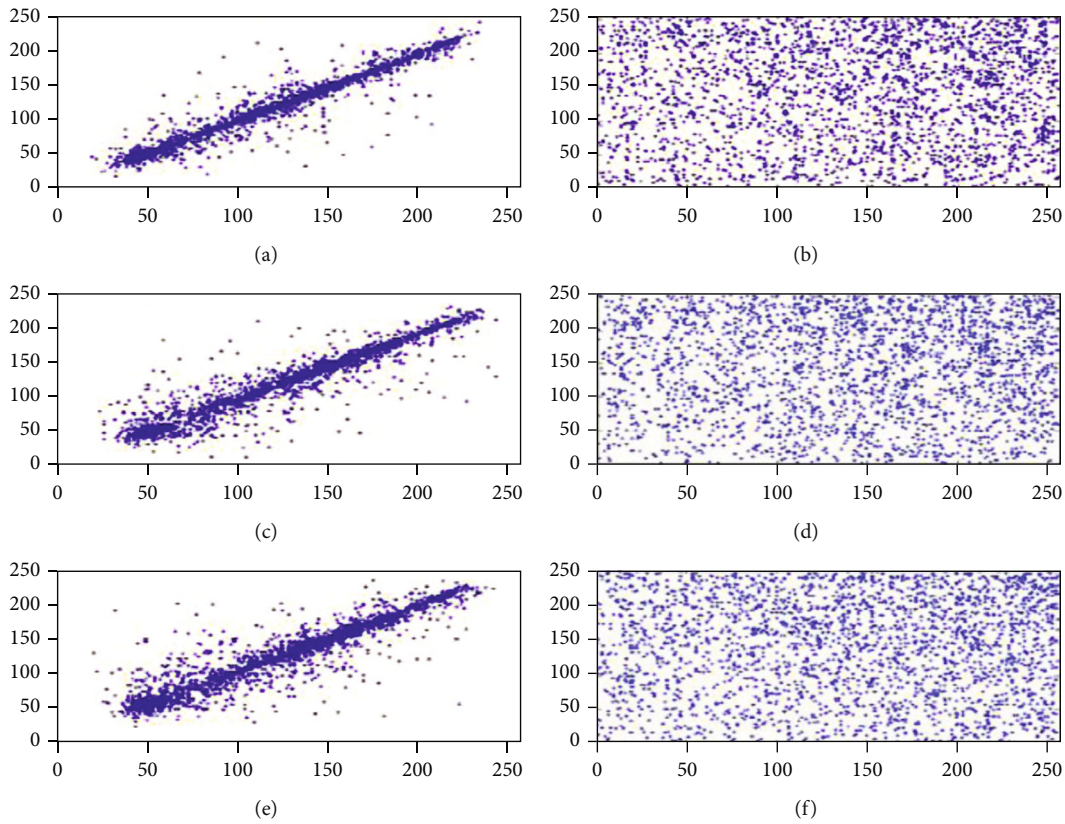


Figure 6: Correlation of two horizontally adjacent pixels.

Table 1: Correlation coefficient of two adjacent pixels.

	Normal image	Password image
Level	0.9431	0.0089
Vertical	0.9725	-0.0215
Diagonal	0.9264	-0.0074

Table 2: NPCR and UACI analyses.

	Normal image	Password image
NPCR	0.0015%	98.7778%
UACI	0.0000%	30.9639%

pixels are chosen at random, and then, the correlation coefficient is calculated by

$$r_{xy} = \frac{|\mathrm{cov}\ (x, y)|}{\sqrt{D(x)} \times \sqrt{D(y)}}, \tag{5}$$

$$\mathrm{cov}\ (x, y) = \frac{1}{N} \sum_{i=1}^{N} x_i, \tag{6}$$

$$D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2, \tag{7}$$

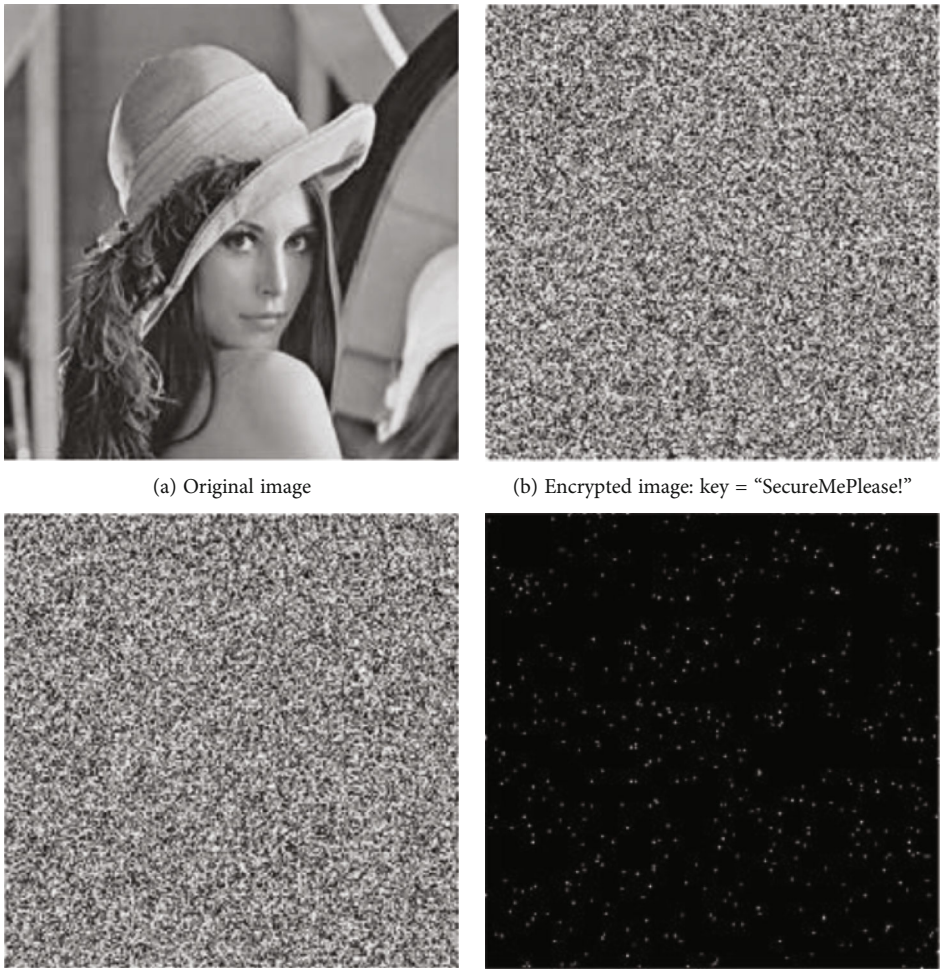
where the gray scale values of the adjacent image pixels are represented by x and y.

Figure 6 depicts, correspondingly, the horizontal, vertical, and diagonal correlations between the two images. Table 1 displays the findings of the regression analysis of adjacent pixels.

4.3. Sensitivity Analysis. Often, an adversary might alter the encrypted image slightly in order to observe the change in results. In doing so, meaningful relationships between plain images and cryptographic images can be found. This is called a differential attack.

4.3.1. NPCR and UACI Analyses. But only when the pictures vary by something like a single pixel, NPCR denotes the rate of difference in the frequency of pixels inside an encryption algorithm. The unified average intensity of change (UACI) metric calculates the average brightness of the variation between the two images. NPCR and UACI both rely on slight adjustments to the two images while maintaining the same key. In reference [10], the original image and key of the NPCR calculated by the author have changed by one bit.

Suppose there are two encrypted pictures, C_1 and C_2, which normal control pictures vary by just one pixel. The gray scale values of the encrypted images C_1 and C_2 are designated as $C_1(i, j)$ and $C_2(i, j)$, accordingly, in the i-th row and j-th column.



(a) Original image

(b) Encrypted image: key = "SecureMePlease!"

(c) Encrypted image: key = "SecureMePlease!"

(d) Difference image

Figure 7: Key sensitivity test.

Table 3: Rate of change.

Original key	New key	Change rate
SecureMePlease!	SecureMePlease.	99.600%
	secureMePlease!	99.161%
	SecuremePlease!	99.245%
	SecureMePlease	99.295%

The definitions of NPCR and UACI are

$$\mathrm{NPCR} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}}{M \times N} \times 100\%,$$
$$\mathrm{UACI} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left|C_i(i,j) - C_2(i,j)\right|}{M \times N \times 256} \times 100\%. \quad (8)$$

The sizes of the picture are M and N, correspondingly.

For our instance, the key "SecureMePlease!" has been used, and the chosen pixel is the position (1,2) of the image pixels. Its number is altered from (10100011)2 to (10100010)2. NPCR and UACI are therefore reported in Table 2.

Table 2 demonstrates how much greater our revised technique is programmatically than the original method in sensitivity.

The responsiveness to the actual picture is quite low in the original method. The whole key picture in the 256 × 256 sample provided has only been modified by roughly 0.0015 percent. The UACI is around 0 [11]. Experiments show that encrypted images are vulnerable to plaintext and known-plaintext attacks.

Consequently, it is evident that our enhanced approach fixes the old system's flaw of being blind to minute alterations in planar images. The encrypted picture's pixels shift whenever a single pixel in the original picture changes (NPCR > 98.77%), and the universal mean shift intensity increases (UACI > 30.96%). The results demonstrate that the effectiveness is satisfactory.

4.3.2. Key Sensitivity Test. Crucial sensitivity is the pace at which a cryptographic picture's pixel count changes even though only one piece of the password is changed. The usual image should first be encrypted with the testing password "SecureMePlease!" before the least-valid password is changed to "SecureMePlease." and the identical image is encrypted. Consider the various password photos once more.

(a) Original image

(b) Encrypted image: key = "SecureMePlease!"

(c) Decrypted image: key = "SecureMePlease!"

(d) Decrypted image: key = "SecureMePlease!"

Figure 8: Key sensitivity test.

Table 4: Encryption time of three encryption algorithms.

Encryption algorithm	Image size	Encryption time
Literature [12]	90*180	2.13
Literature [13]	128*128	1.24
The algorithm in this paper	128*128	0.15

The outcome is in regard to pixel gray levels, the encrypted image is completely different with a slight difference in the key (as shown in Figure 7). Table 3 displays the outcomes of the encryption with different keys, with an average rate of change as high as 99.323%.

Also, if the image is encrypted with a key and decrypted with another, simply modified key, the decryption fails. Figure 8 demonstrates how photos encrypted using the "SecureMePlease!" password cannot be properly decoded with the "SecureMePlease" password.

4.4. Other Security Analysis

4.4.1. Resist Plaintext Attack. We may observe a significant change at the crucial stream creation when evaluating the original method with the revised approach. The user's passcode is the single factor that determines how the key stream is utilized in the original file, whether it is for an XOR operation with the actual picture or for postprocessing. It is separated from typical visuals. The key stream in our enhanced technique is based on the original picture's properties as well as the user's passcode. That is, although just use the same passcode, if various photos are encoded, the crucial stream of the XOR stage is unpredictable. This person assists with encrypted photographs defend against selected plaintext attacks and known plaintext attacks.

Also, a cryptanalyst sending a dark image through into encryption method seems to have no impact on the operation because the key stream is varied when encrypting various images with the same cipher. The "chosen plain text assault" discussed above can be eliminated by our enhanced algorithm.

4.4.2. Diffusion and Chaos. Obfuscation and diffusion are two characteristics of secure cryptographic procedures that

were first established by Claude Shannon in the field of cryptography. NPCR demonstrates that when just one pixel of the plaintext is altered, nearly every pixel in the cryptographic picture is altered, as illustrated in Section 4.3.1 (NPCR > 98.77 percent). The revised algorithm's diffusion characteristics are excellent.

Through the correlation analysis of the gray histogram and adjacent pixels, the proposed improved algorithm has good chaos.

4.4.3. Brute Force Attack. The suggested enhanced algorithm depends on the required space length and critical sensitivities for brute force attack evaluation.

Essential space: the SHA-2 method and the FFT-DCT composite transform are irreparable processes, as was already stated; in addition, ARM will arrange the pixels, which is very secure for common commercial applications

Key sensitivity: as shown in Figures 7 and 8, even a small key change causes almost all pixels to change the corresponding cryptographic image

Therefore, our algorithm is highly resistant to brute force attacks.

4.5. Comparison of Similar Algorithms. The algorithm proposed in this paper has superiority by comparing and analyzing the algorithm of literature [12] and literature [13]. Reference [12] suggested a Feistel network-based picture encryption technique, which has high security and can be comparable to the algorithm in this paper, but there is a gap with this paper in terms of sensitivity. This paper also introduces in Section 4.3. The algorithm in this paper Fewer iteration rounds are required.

In addition, although the literature [13] has been improved, the encryption speed of the algorithm in this paper is significantly higher than that of the literature [13]. Table 4 shows the image encryption algorithms proposed in this paper, literature [12] and literature [13]. The time required to encrypt the same image, it can be seen that the encryption speed of this paper is the fastest that is because the literature [12] uses a large number of iterations, resulting in a slow operation, while the literature [13] is because there are too many rounds. This results in increased computation time, which in turn slows down encryption.

5. Conclusion

In this research, we suggested a hash-based fast picture encryption algorithm for Internet of Things (IoT) applications, where the image is split into equivalent left and right portions, and the data from one half is used to encrypt the other part in turn. Theoretical study and computer simulation demonstrate the robustness of our suggested approach against chosen plaintext assaults as well as chosen plaintext attacks.

Data Availability

All the data used to support the findings of this study are available in the article.

Conflicts of Interest

No contradictions exist, according to the researchers, with the publication of this research.

References

[1] H. Dai, W. Dong, and S. Zhong, "Design and analysis of a class of SHA-x improved hash algorithms," *Computer Engineering*, vol. 35, no. 6, pp. 181-182+185, 2009.

[2] R. Liu and T. Tan, "A review of research on digital image watermarking," *Journal of Communications*, vol. 21, no. 8, pp. 40–49, 2000.

[3] J. Wu and S. Song, "An improvement of text encryption method," *Journal of Chongqing University of Science and Technology: Natural Science Edition*, vol. 6, no. 2, pp. 55-56, 2004.

[4] A. Zhou and Y. Yu, "Robust speech recognition by adopting random projection in feature space," *Computer Applications*, vol. 32, no. 7, pp. 2070–2073, 2012.

[5] X. Liao, S. Lai, and Q. Zhou, "A novel image encryption algorithm based on self-adaptive wave transmission," *Signal Processing*, vol. 90, no. 9, pp. 2714–2722, 2010.

[6] D. Xiao and F. Y. Shih, "Using the self-synchronizing method to improve security of the multi chaotic systems-based image encryption," *Optics Communications*, vol. 283, no. 15, pp. 3030–3036, 2010.

[7] R. Liu, F. Li, and L. Su, "Bilateral filtering based image restoration for multiple grayscale images," *Computer Applications*, vol. 30, no. 4, pp. 902–904, 2010.

[8] Y. Li, "Simulation calculation of control system—MATLAB," *Computer Measurement and Control*, vol. 12, no. 4, pp. 40–43, 1996.

[9] Y. Tang, Z. Wwang, and J. A. Fang, "Image encryption using chaotic coupled map lattices with time-varying delays," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 9, pp. 2456–2468, 2010.

[10] S. Deng, Y. Zhan, and D. Xiao, "Analysis and improvement of a hash-based image encryption algorithm," *Communications in Nonlinear Science & Numerical Simulation*, vol. 16, no. 8, pp. 3269–3278.

[11] A. Manikond and P. Mangalampalli, *UACI: Uncertain Associative Classifier for Object Class Identification in Images*, 2012.

[12] F. Li and J. Xu, "Image encryption algorithm based on hash function and multi-chaotic system," *Computer Engineering and Design*, vol. 31, no. 1, pp. 141–144, 2010.

[13] G. Chen, X. Zhao, and J. Li, "A self-adaptive algorithm on image encryption," *Journal of Software*, vol. 16, no. 11, p. 1975, 2005.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1903197, 17 pages
https://doi.org/10.1155/2022/1903197

Research Article

Certificateless Group to Many Broadcast Proxy Reencryptions for Data Sharing towards Multiple Parties in IoTs

Won-Bin Kim,[1] Su-Hyun Kim,[2] Daehee Seo,[3] and Im-Yeong Lee[1]

[1]*Department of Software Convergence, Soonchunhyang University, Asan 31538, Republic of Korea*
[2]*National IT Industry Promotion Agency, Jincheon 27872, Republic of Korea*
[3]*Faculty of Artificial Intelligence and Data Engineering, Sangmyung University, Seoul 03016, Republic of Korea*

Correspondence should be addressed to Im-Yeong Lee; imylee@sch.ac.kr

Received 31 March 2022; Accepted 11 June 2022; Published 29 June 2022

Academic Editor: Yan Huo

Proxy reencryption delegates encrypted data stored in a proxy to a third party. This proxy reencryption takes the form of one sender providing data to one receiver. However, this method incurs a significant overhead for both the sender and proxy as the number of users receiving the same data increases. In addition, in a large-scale environment, such as an Internet of Things or big data environment, a scenario where several workers jointly create and own an output may exist. In such an environment, ownership disputes can arise when only one operator owns a piece data used by other operators. In this study, to solve this problem, we propose a technique in which multiple users can jointly own one piece of data, and multiple recipients can receive the same data through proxy reencryption.

1. Introduction

The development of information technology has brought about numerous changes to data storage and utilization technology. The Internet, which is the most widely used network, has made it possible to transmit and use data anytime and anywhere without restrictions in time and place. Internet technologies have been developed to achieve improved speeds, allowing more data to be transmitted concurrently. In addition, the Internet can be used in a wireless form. Storage media that allow more data to be stored and used in a unit area have also been developed. Because more data can be stored in a smaller space, removable storage devices have emerged, and removable storage media have provided an environment in which data can be held and utilized more efficiently. The development of such network technologies and storage media has recently achieved a rapid growth and has taken on various forms, reaching the stage of virtual storage spaces such as cloud computing. We believe that this change in the environment is a transition from an environment using a storage medium to an environment using a storage space, and that the change in such an environment is accelerating.

Gartner, an American information technology research and advisory firm, publishes the Top Strategic Technology Trends and Hype Cycles [1]. Cloud computing is an important strategic technology to the extent that it is selected by this publication every year. However, despite the growing awareness and importance of cloud computing, many companies and institutions are hesitant to adopt it for security reasons. Because cloud computing technology is always connected to a network, it is continuously exposed to data leakage and multiple foes using the network. Therefore, security technology is essential when introducing cloud computing. The secure storage and transmission of data are essential for a secure cloud computing environment. In addition, cloud storage, a subclass of cloud computing technology, stores data and must provide availability for future use. Therefore, cloud computing must consider more security factors than portable storage media.

Cloud storage is a representative technology for storing data using cloud computing technology. As described above, cloud storage can be used as storage space by utilizing network technology, and in this way, the digital data can be stored and used without a physical storage medium. Using the advantages of cloud storage, one can not only store and use one's own data, such data, and also be shared with other users. Data sharing in this manner increases the efficiency because data can be passed through the cloud storage without being passed directly between the data owner and recipient. In addition, even when sharing the same data with multiple recipients, it achieves the advantage of being able to transmit data from cloud storage without the need for the owner to transmit the data each time the data are accessed. However, as described above, the cloud computing technology used over a network is continuously exposed to data leakage and security threats. Therefore, the security factor must be considered in data-sharing methods using cloud storage.

To securely share data using cloud storage, protection of both the data and transmission process must be considered. In general, a cloud storage server is a remote server managed by a data owner and other administrators. Such a server has an honest-but-curious characteristic, which processes the user's request accurately but always incurs the possibility of exposing the data. Therefore, if an owner's sensitive data are stored in cloud storage, there is a possibility that the content of the data will be exposed. Data encryption must be applied to solve this problem. Data encryption technology refers to a technology in which only a user who possesses a decryption key corresponding to the encryption key of the data can view the content of the encrypted data. Therefore, only a user who has a decryption key corresponding to the encryption key used for the data uploaded by the owner can view the content of the data. Two encryption algorithms may be primarily used for this encryption method, and a total of four encryption methods may be used by combining the two encryption algorithms. However, these four encryption methods cannot be applied to data-sharing methods using cloud storage because each of them has certain problems such as a key distribution, computational inefficiency, and exposure to the data source. To solve this, a proxy reencryption technique has been proposed.

Proxy reencryption technology securely shares data using a proxy server, as proposed by Blaze et al. in 1998 [2]. Proxy reencryption technology refers to a technology that stores data encrypted with the owner's encryption key in the proxy and then converts the encrypted data into a specified number of cipher texts. During this process, because the proxy does not decrypt the encrypted data, the contents of the data cannot be known, and the receiver can decrypt the data using its own private key. Therefore, the data are not exposed during the process of data storage and delivery. With this proxy reencryption technology, the proxy may be represented by cloud storage, and if such technology is used, data can be shared securely and efficiently in the cloud storage environment.

As large-scale network environments such as IoT, secure e-mail, and connected cars become more common, cases of data sharing between multiple users are increasing [3–5]. In such an environment, data sharing using cloud storage can be an effective way to deliver data securely and efficiently to multiple users. However, because general proxy reencryption technology uses a 1 : 1 data transmission method, it cannot support multiple data owners or multiple data receivers. In this case, to provide the same data to multiple recipients, it is necessary to generate a reencryption key and conduct as many reencryption operations as the number of recipients. In addition, even when multiple workers collaborate to create a single data point, only one worker can be the owner. In this case, because the data cannot be efficiently owned or shared in a large-scale data ownership and reception environment, an appropriate method that considers these issues is required. This study was conducted to provide a method that considers multiple owners and recipients simultaneously. Thus, it provides a method for flexibly and efficiently carrying out the ownership and sharing of data using proxy reencryption technology.

2. Related Works

This section describes related studies for a proper understanding of this study.

2.1. Secure Data Sharing. As a basic concept of data-sharing technology, data owners give permission for their data to be available to other users. In existing systems, such as Linux or Windows, ownership of data is provided in the same form as RWX, and the meanings of readable, writable, and executable are the same. This indicates that data ownership is further subdivided and provided as a logical form of usage rights. By contrast, from a cryptographic perspective, data ownership can be accessed in the form of determining whether data can be decrypted. That is, if one has a decryption key corresponding to a key having encrypted data, it can be determined that one has ownership of the data because the data source can be obtained through decryption. Therefore, the method of sharing data through such a cryptographic concept can be accessed by delegating the decryption authority of the encrypted data [6].

A method of providing the decryption rights of encrypted data to another user can be approached in four major ways using a symmetric key encryption algorithm, and a public key encryption algorithm is shown in Figure 1

(1) *Use of only symmetric key encryption:* with this method, the data that the sender uploads to the proxy are first encrypted with the sender's own symmetric key and uploaded. When the receiver requests data, the proxy delivers a ciphertext of the sender to the receiver, and the sender must deliver its symmetric key to the receiver. When this method is applied, both the sender and receiver can conduct encryption/decryption using a symmetric key. However, this process requires a symmetric key distribution process. Symmetric key eavesdropping by an attacker may occur during the process of symmetric key distribution. In addition, because the symmetric key delivered to the recipient cannot be delivered to

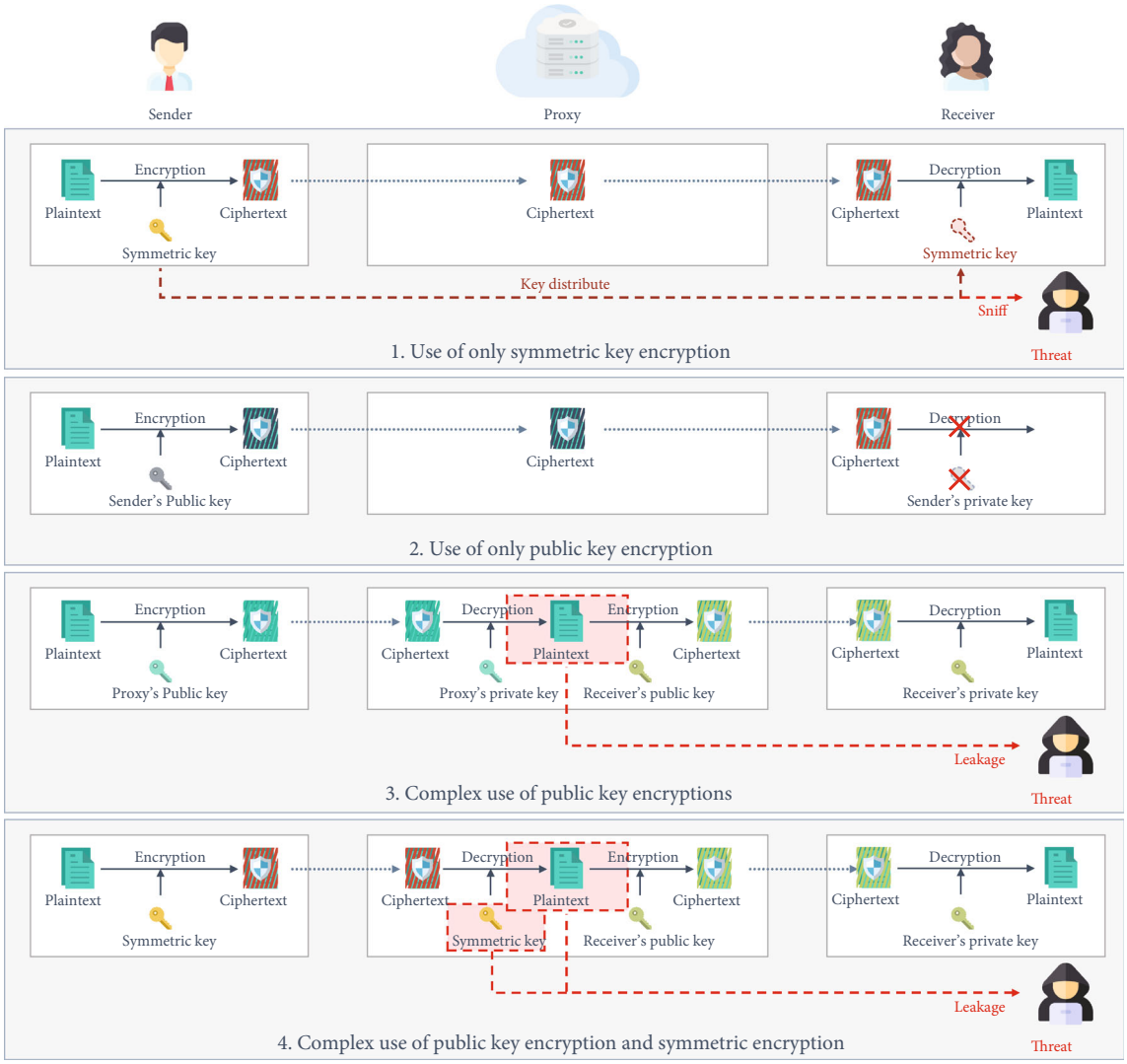


Figure 1: Problems of data sharing method using encryption.

another recipient, reusing the ciphertext uploaded to the proxy becomes impossible. Therefore, the data sharing method using symmetric key encryption is unsuitable in terms of security and efficiency

(2) *Use of only public key encryption:* with this method, the data that the sender uploads to the proxy are first encrypted with the sender's public key and then uploaded. When the receiver requests data, the proxy delivers the sender's ciphertext to the receiver. However, because this method can only be decrypted using the sender's private key, the sender must deliver his or her private key to the receiver. However, in this case, the sender's private key is exposed by other users, which can lead to serious security problems. Consequently, the receiver cannot decrypt the ciphertext of the sender without lowering the level of security

(3) *Complex use of public key encryptions:* with this method, the data uploaded by the sender to the proxy are first encrypted and uploaded with a symmetric key shared between the sender and the proxy. Upon receiving the data, the proxy decrypts the ciphertext of the sender using a symmetric key to obtain the original data. After that, just like the *2. Use of only public key encryption* method, the data source is encrypted with the recipient's public key and delivered to the recipient, who can decrypt it. As with the method that uses public key encryption multiple times, the data source is encrypted with the recipient's public key and delivered to the recipient, and the recipient can decrypt it. In this method, even if there are many recipients, the proxy can directly perform encryption with the public key of each recipient, so that the computational burden on the sender is not increased. As with the method of using public key encryption multiple times, even if the number of recipients increases, the computational burden on the sender does not increase because the proxy can conduct encryption directly

using the public key of each recipient. However, this process allows the proxy to know the list of recipients, exposing the contents of the data source to threats both inside and outside the proxy. Therefore, the method of using public key encryption and symmetric key encryption together has the efficiency of data sharing but without guaranteeing security

(4) *Complex use of public key encryption and symmetric encryption:* with this method, the data that the sender uploads to the proxy are first encrypted with the sender's public key and then uploaded. When the receiver requests data, the proxy delivers the sender's ciphertext to the receiver. However, because this method can only be decrypted using the sender's private key, the sender must deliver his or her private key to the receiver. However, in this case, the sender's private key is exposed by other users, which can lead to serious security problems. Consequently, the receiver cannot decrypt the ciphertext of the sender without lowering the level of security

As described above, use of the symmetric and public key encryption methods to securely share data through cloud storage does not provide sufficient security. Therefore, a method that can provide both security and efficiency throughout the data sharing process is required. Various studies have been conducted to satisfy such requirements, and proxy reencryption technology has been proposed.

2.2. Proxy Reencryption. In 1998, Blaze et al. proposed proxy reencryption (PRE) [2], which is a technology that transforms data through a proxy and delivers them securely to the receiver. This technology converts data encrypted using the sender's public key into data encrypted using the receiver's public key at a proxy. Through this process, the private keys of the sender and receiver, as well as the original data, are not exposed because data decryption is not applied. Using proxy reencryption, data can be securely stored in cloud storage and shared efficiently by converting the data into the recipient's ciphertext at the request of the recipient. The basic form of such a proxy reencryption is shown in Figure 2, and research on various sharing methods using proxy reencryption technology is currently underway.

Proxy reencryption comprises five steps: encryption, reencryption key generation, reencryption, decryption, and redecryption. The details of each step are as follows:

(i) *Encryption:* in this step, the data owner encrypts the data and uploads them to a proxy. To this end, the data owner encrypts the data using his or her own encryption key, such that the source of the data cannot be known. The encrypted data are then delivered to the proxy through the public network and stored. In this case, the proxy cannot know the contents of the data stored in the proxy, and even if the encrypted data are exposed or leaked, decryption corresponding to the encryption key is applied, and a user without a key cannot know its contents

(ii) *Reencryption key generation:* in this step, the data owner provides the receiver with the authority to decrypt his or her data stored in the proxy. For this, the data owner first receives the information of the recipient who requested the data. The data owner then creates a reencryption key by combining the information of the recipient with his or her own decryption key and secret information. The data owner can control the reencryption by passing the generated reencryption key to the proxy. In this case, the proxy and attacker should not be able to obtain the secret information of the data owner through the reencryption key

(iii) *Reencryption:* this step refers to the process of converting the encrypted data of the data owner into receiver data. To this end, the proxy applies a reencryption algorithm using the cipher text and reencryption key received from the data owner, and as a result, can obtain a reencrypted cipher text. In this case, the reencrypted cipher text is the cipher text in which the decryption authority is delegated from the data owner to the receiver, and the proxy cannot know the contents of the data during the reencryption process. The reencrypted ciphertext is then sent to the receiver

(iv) *Decryption:* in this step, the data owner decrypts the ciphertext. This step is conducted to obtain the data source by downloading the ciphertext uploaded by the data owner to the proxy during the encryption step again by the data owner. Accordingly, the data owner represents the data decryption process using a decryption key that corresponds to the encryption key used for data encryption. This process represents a typical encryption-decryption relationship and shows that data owners can reuse their data at will

(v) *Redecryption:* in this step, the receiver decrypts the reencrypted ciphertext. To this end, the receiver receives the reencrypted cipher text from the proxy and performs a process of decrypting the received cipher text using its decryption key. At this time, if the recipient is not the correct recipient, the data cannot be decrypted even if the reencrypted cipher text is received

Most proxy reencryption structures are as above, and various methods can be used to configure the above steps. Currently, most proxy reencryption studies use public-key encryption methods [7–16]. Because PKC performs encryption using a public key, it offers excellent accessibility and usability. However, additional computations and certificate management problems occur because procedures such as the generation of a certificate for the public key are essential. To solve this problem, identity-based PKC (IB-PKC) using a key issuance method through a key generation center (KGC) has been proposed [17]. Since IB-PKC was first proposed,

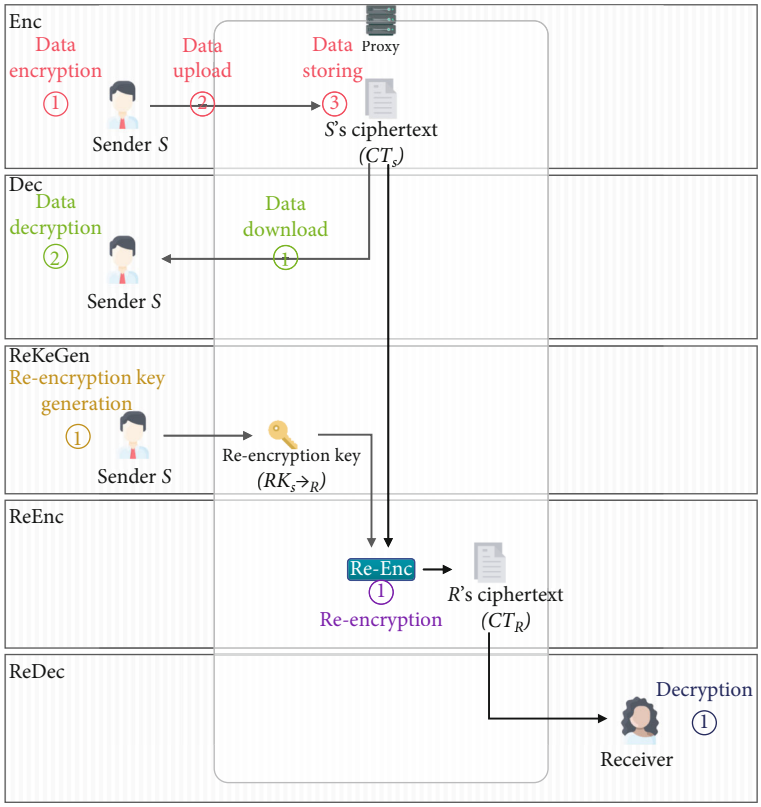


Figure 2: Basic proxy reencryption method.

various proxy reencryption studies based on IB-PKC have been conducted [7, 18–22]. However, in IB-PKC, because KGC directly issues the user's key, the problem of a key escrow by the KGC arises. To solve this problem, CL-PKC, a method in which a complete key is not generated by the KGC without the use of a certificate, has been proposed [23]. CL-PKC follows a method in which KGC issues only a partial secret key to each user, and the users then combine their secret information to complete a private key. Therefore, the key escrow problem of KGC does not occur. Accordingly, studies on certificateless proxy reencryption (CL-PRE) have recently been conducted using CL-PKC [24–27].

2.3. Multireceiver Encryption. Multireceiver encryption (MRE) is a technology that grants the same data decryption authority to multiple recipients with only a single encryption. MRE has been utilized in various studies based on PKC as shown in Figure 3 [28–36]. However, the existing MRE method has the problem of receiver identification. This is because the recipient can be identified by extracting the recipient information included in the ciphertext. To solve this problem, a method for specifying the receiver using a polynomial has been proposed [37]. Using this method, the receiver's information cannot be extracted by combining it with a polynomial. However, other studies have demonstrated that this scheme can obtain the recipient's identity [38, 39]. Fan et al. proposed an improved version of this scheme [40]. In addition, Zhang and Takagi proposed a method in which both the sender and receiver are anonymous [41]. However, Zhang and Mao found that this scheme does not provide complete anonymity; therefore, they proposed a new type of identity-based MRE (IB-MRE) [42]. However, after the key escrow problem of IB-PKC was presented, a study was conducted on applying CL-PKC to MRE.

Based on research conducted on CL-MRE, Sur et al. improved the implicit certificate-based MRE proposed in 2007 [43] and proposed CL-MRE in 2011 [44]. In addition, Islam et al. proposed a CL-MRE, which achieved confidentiality and anonymity in a random oracle model [45]. However, Hung et al. pointed out a large number of computations, similar to that indicted by Islam, which takes a lengthy computation time [46]. However, Hung et al. also had a problem in that the map-to-point (MTP) hash operation, which requires a lengthy operation time, increases linearly in proportion to the number of users. He et al. [47] proposed a method that does not use a map-to-point (MTP) hash to solve this problem. Although Deng et al. [48] and Zhu et al. [49] proposed CL-MRE to solve the key escrow problem, a considerable computational load was incurred using bilinear pairing, and the scheme developed by Zhu et al. did not provide additional receiver anonymity. Although Win et al. [50] did not use bilinear pairing, they also did not provide receiver anonymity or decryption fairness.

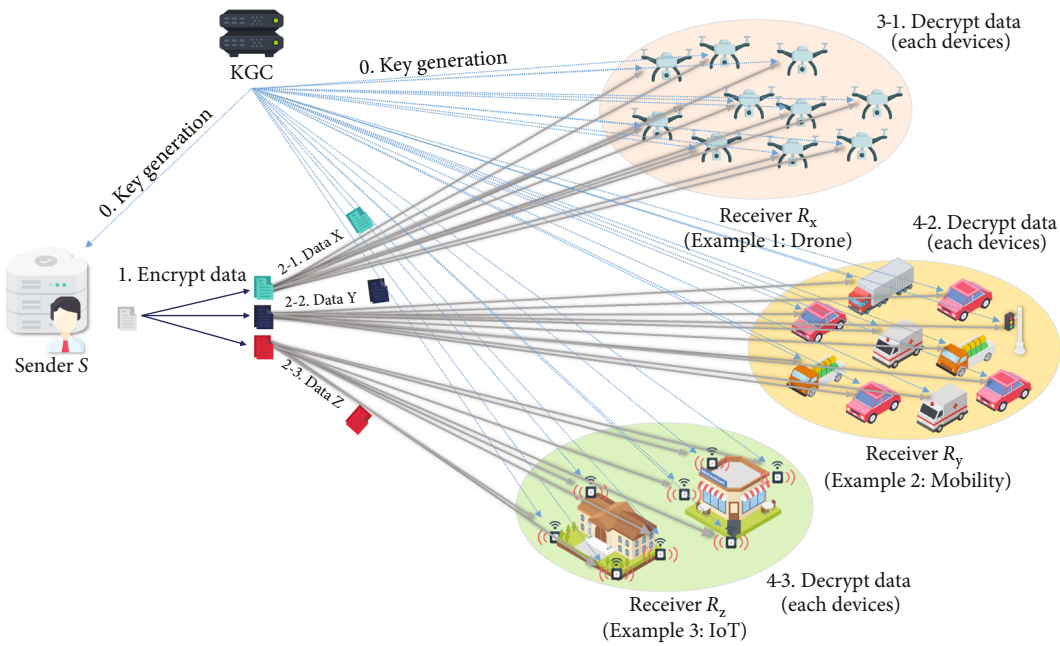


Figure 3: Flow of multireceiver encryption.

3. Preliminaries

This section describes the basic environment and settings for understanding the scheme proposed in this study.

3.1. System Model. This section describes the system model used in the present study. The participants in this system model are divided into KGC, proxy, user, owner, and receiver, and the description of each participant is as follows.

(i) *Key generation center* (*KGC*): with this model, KGC plays a role in managing the system administrator or users in the system. KGC manages all users in the system and registers and manages users through preset settings. In addition, common parameters are created and disclosed such that all participants can conduct the operations of a predetermined algorithm. Using these parameters, all participants can generate their own keys or conduct such predetermined algorithm operations. At this time, to avoid the key escrow problem caused by the KGC, the KGC cannot know the user's complete key

(ii) Proxy: with this model, a proxy indicates a remote server that can store and distribute data between users. The most representative form of a proxy is cloud storage, which can store, transmit, and calculate data according to the user's request. With this model, because the proxy is considered a semitrusted environment, there is a possibility that the contents of the unencrypted data may be exposed or leaked

(iii) *User:* using this model, a user means all users including the owner and receiver. Each user has his/her own public and private keys and can encrypt and decrypt data using these keys

(iv) *Owner group*: with this model, the owner means the group of users who own the data. It is assumed that ownership of one piece of data is shared by several users. Examples of such environments include operations, organizations, and the military. Under this environment, because each user has equal ownership, decryption and reencryption keys can be generated using the threshold method to prevent abuse of authority by one owner

(v) *Receiver*: with this model, the receiver means all receivers who receive the data decryption right from the owner. These recipients may consist of one or more individuals, and multiple recipients who have been granted the same data rights have the same rights. In addition, each authorized recipient can decrypt the data using their own private keys

3.2. Security Requirements. This study consists of seven security requirements. The details are as follows:

(i) *Confidentiality*: the data that are kept in the proxy, and the data delivered through the proxy, shall not be unknown other than to the authorized user. To do this, the data must be encrypted using the encryption key, and a user who does not have a legitimate decryption key should not be able to decrypt the contents

(ii) *Integrity*: data uploaded and shared by the sender must not be changed without permission in the process of being delivered to the cloud and the receiver and stored in the proxy. If the content is changed at all, the sender or receiver who shares the data must be made aware of the change

(iii) *Key escrow problem*: all users who want to use the proxy must communicate with the KGC to generate a private key and public key pair. During this process, the KGC generates a user's full private key, and the KGC may increase the user's authority. This problem is called the key escrow problem, and a method for solving this problem is required

(iv) *Partial key verifiability*: to solve the previously described key escrow problem, a key generation method in the form of a partial key can be used. In this case, each user must be able to verify whether the partial key generated and issued by the KGC to each user is generated legitimately by the correct KGC

(v) *Receiver anonymity*: the reencrypted ciphertext in proxy storage can be decrypted by a number of designated receivers. For this purpose, the reencryption key and reencrypted ciphertext include the information generated by the public key of each receiver. However, privacy issues arise when such information allows a particular recipient or a third party to identify another receiver

(vi) *Decryption fairness*: each legitimate receiver designated by the sender can decrypt the reencrypted ciphertext. However, through this process, a specific receiver should not be discriminated against or disadvantaged during the decryption by a specific receiver or third party

3.3. Algorithms. This section describes the algorithm used for the proposed scheme. Eleven algorithms were used in this study: Setup, Set-Secret-Value, Partial-Key-Extract, Set-Private-Key, Set-Public-Key, Set-Owner-Group, Enc, Re-Key-Gen, Re-Enc, Dec, and Re-Dec. The description of each algorithm is as follows.

(i) *Setup*: this algorithm is executed by inputting a security parameter. With this algorithm, the KGC generates public parameters and master secret keys and publishes the public parameters, which are made available for all users and proxies

(ii) *Set-Secret-Value*: this algorithm is applied by the user. With this algorithm, user i calculates T_i using a randomly selected t_i and sends T_i and ID_i to the KGC

(iii) *Partial-Key-Extract*: this algorithm is performed by KGC. Using this algorithm, the KGC generates the partial key (R_i, k_i) of user i using (T_i, ID_i) and mpk received from user i and sends it to user i

(iv) *Set-Private-Key*: this algorithm is applied by the user. With this algorithm, the user calculates private key sk_i using partial key (R_i, k_i) received from the KGC. The sk_i obtained is kept confidential

(v) *Set-Public-Key*: this algorithm is applied by the user. Using this algorithm, the user calculates the public key pk_i by using the partial key (R_i, k_i) received from the KGC and the secret value t_i generated by user i. The pk_i values obtained are disclosed

(vi) *Initialization, Group Agreement:* this algorithm is run by users to be included in the owner group. With this algorithm, users $\mathcal{G}_j$ that are to be included in the owner group $\mathcal{G}$ exchange the public key $\mathrm{gpk}_{\mathcal{G}}$ with each other to generate the group key

(vii) *Enc:* this algorithm is applied by users included in the owner group. In this algorithm, member $\mathcal{G}_j$ of owner group $\mathcal{G}_j$ encrypts plaintext m with public key $\mathrm{gpk}_{\mathcal{G}}$ of owner group $\mathcal{G}$ to obtain ciphertext CT. Subsequently, the obtained ciphertext, CT, is transmitted to the proxy and stored

(viii) *Re-Key-Gen:* this algorithm is applied by users included in the owner group. With this algorithm, member $\mathcal{G}_j$ of the owner group $\mathcal{G}$ uses the group private key $\mathrm{gsk}_{\mathcal{G}}$ and calculates the reencryption key $\mathrm{RK}_{\mathcal{G}\longrightarrow\mathcal{R}}$ using the receiver's public key $\mathrm{pk}_{\mathcal{R}}$. In this case, the receiver consists of one or more persons. Member $\mathcal{G}_j$ of 1owner group $\mathcal{G}$ passes the reencryption key $\mathrm{RK}_{\mathcal{G}\longrightarrow\mathcal{R}}$ to the proxy

(ix) *Re-Enc:* this algorithm is conducted by a proxy. Using this algorithm, the proxy applies reencryption using the cipher text CT uploaded by the owner group $\mathcal{G}$ and reencryption key $\mathrm{RK}_{\mathcal{G}\longrightarrow\mathcal{R}}$. The reencrypted ciphertext CT_R is then obtained. Subsequently, the acquired CT_R is broadcast

(x) *Dec:* this algorithm is applied by a user included in the owner group. Using this algorithm, a member $\mathcal{G}_j$ of the owner group $\mathcal{G}$ can download ciphertext CT stored in the proxy. Subsequently, members $\mathcal{G}_j$ may obtain plaintext m by decrypting the ciphertext CT with their group private key $\mathrm{gsk}_{\mathcal{G}}$

(xi) *Re-Dec:* this algorithm is conducted using the receiver. With this algorithm, the recipient r_j included in the receiver set $\mathcal{R}$ decrypts the reencrypted ciphertext CT_R received from the proxy with its private key sk_{r_j}, and the plaintext m can thus be obtained

4. Proposed G2M Broadcast Proxy Reencryption

This section describes the proposed scheme. For this purpose, a technical overview, system parameters, and algorithm construction are described.

4.1. Technical Overview. The basic model of the proposed scheme, as shown in Figure 4, can be broadly divided into five phases: *a Setup Phase*, *Key Generation Phase*, *Group Agreement Phase, Data Storage Phase*, and *Data Broadcast*

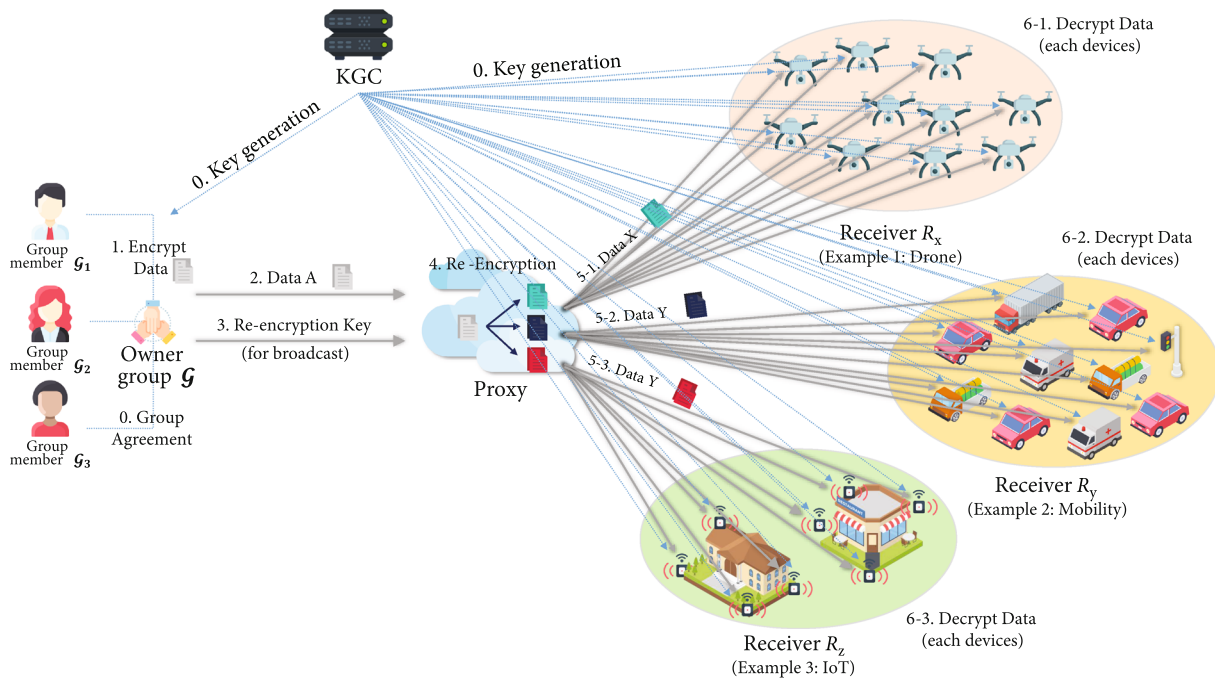


Figure 4: System model of proposed scheme.

Phase. More details regarding these phases are presented in Sections 4.2 and 4.3.

4.2. System Parameters. The system parameters used in the proposed scheme are as follows:

(i) $*$ Participants (KGC, user i, owner group $\mathcal{G}$, owner group member $\mathcal{G}_j$, receiver set $\mathcal{R}$, receiver r_j)

(ii) p, q: λ-bits prime integer

(iii) $\mathbb{E}$: elliptic curve

(iv) $\mathbb{F}_q$: finite field for q

(v) λ: security parameter

(vi) l_1, l_2: length of the message space (determined by the λ)

(vii) P: random generator in $\mathbb{G}_q$ ($P \in \mathbb{G}_q$)

(viii) $\mathbb{G}$: additive group on the elliptical curve, $\mathbb{E}$

(ix) $\mathbb{G}_q$: subgroup of $\mathbb{G}$ with prime order q

(x) ID_*: identity of the participant $*$ ($\mathrm{ID}_* \in \{0,1\}^*$)

(xi) msk: KGC system master secret key

(xii) mpk: KGC system master's public key

(xiii) sk_i: user i's private key

(xiv) pk_i: user i's full public key

(xv) $\mathrm{RK}_{\mathcal{G}\longrightarrow\mathcal{R}}$: reencryption key (owner group $\mathcal{G}$ delegates to receiver set $\mathcal{R}$)

(xvi) M: message space

(xvii) m: plaintext (message) ($m \in M$)

(xviii) CT: ciphertext

(xix) CT_R: reencrypted ciphertext

(xx) H_1 : one-way hash function, $\{0,1\}^* \longrightarrow \mathbb{Z}_q^*$

(xxi) H_2 : one-way hash function, $\{0,1\}^* \times \mathbb{Z}_q^* \longrightarrow \mathbb{Z}_q^*$

(xxii) H_3 : one-way hash function, $\{0,1\}^* \times \mathbb{G}_q \times \mathbb{G}_q \times \mathbb{G}_q \times \mathbb{G}_q \longrightarrow \mathbb{Z}_q^*$

(xxiii) H_4 : one-way hash function, $\mathbb{G}_q \longrightarrow \mathbb{Z}_q^*$

(xxiv) H_5 : one-way hash function, $\mathbb{G}_q \times \{0,1\}^* \longrightarrow \{0,1\}^{l_2}$

(xxv) H_6 : one-way hash function, $\mathbb{G}_q \times \mathbb{G}_q \longrightarrow \{0,1\}^{l_1+l_2}$

(xxvi) H_7 : one-way hash function, $\mathbb{G}_q \times \mathbb{G}_q \times \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$

4.3. Main Algorithm. The scheme was designed based on Kim et al. [51] and Braeken [52]. This scheme is mainly composed of five phases, each of which comprises a *Setup Phase*, *Key Generation Phase*, *Group Agreement Phase*, *Data Storage Phase*, and *Data Broadcast Phase* as shown in Figure 5. A detailed description of each phase is given.

4.3.1. Setup Phase. This phase includes a *Setup* algorithm. This phase is performed by the KGC in advance so that each user can use the proxy. Here, a master public key that can be

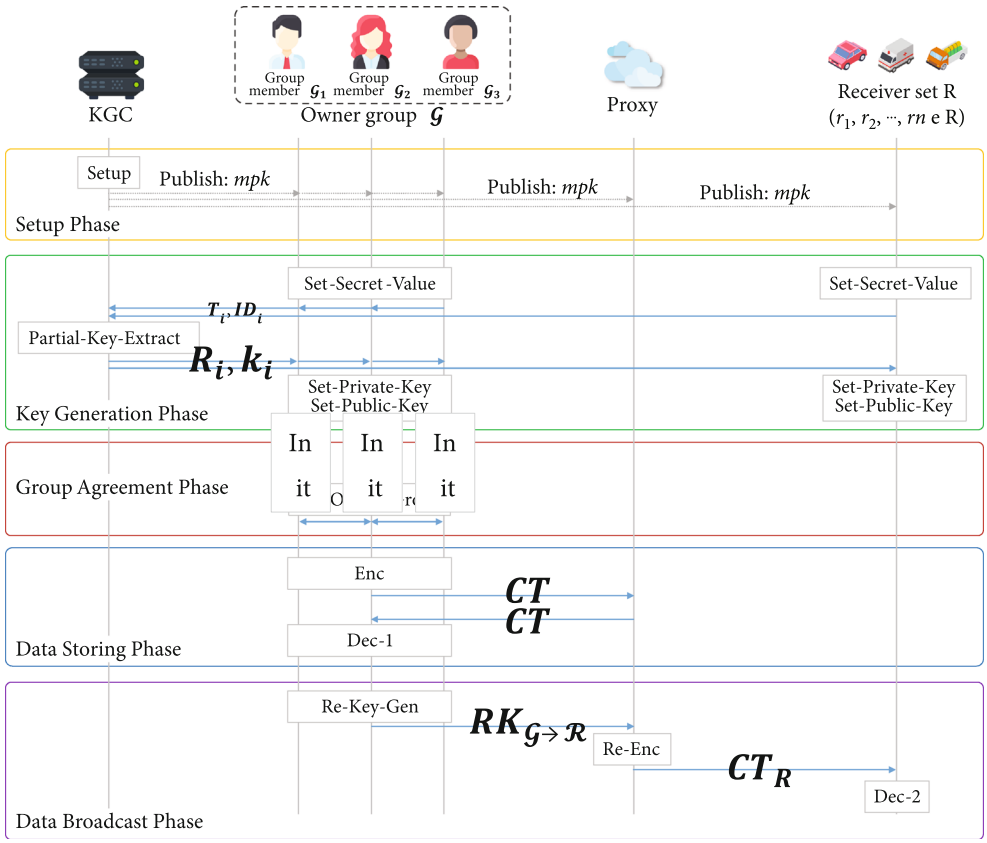


Figure 5: Overview of proposed scheme.

commonly used by each user and a master secret key known only to the KGC are generated.

(i) *Setup* $(\lambda) \longrightarrow$ (msk, mpk): this algorithm is executed by the KGC. With security parameter λ as the input, the KGC performs the following process:

(1) Choose two λ-bits prime integers p, q and elliptic curve $\mathbb{E}$ defined on $\mathbb{F}_p$. Let $\mathbb{G}$ be an additive group on the elliptic curve $\mathbb{E}$ and $\mathbb{G}_q$ be a subgroup of $\mathbb{G}$ with prime order q

(2) Select randomly a generator $P \in \mathbb{G}_q$

(3) Randomly choose $d \in \mathbb{Z}_q^*$ as the *msk* and calculate $P_{\text{pub}} = d \bullet P$ which is part of mpk

Select five secure one-way hash functions as follows:

$$H_1 : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$$

$$H_2 : \{0,1\}^* \times \mathbb{Z}_q^* \longrightarrow \mathbb{Z}_q^*$$

$$H_3 : \{0,1\}^* \times \mathbb{G}_q \times \mathbb{G}_q \times \mathbb{G}_q \times \mathbb{G}_q \longrightarrow \mathbb{Z}_q^*$$

$$H_4 : \mathbb{G}_q \longrightarrow \mathbb{Z}_q^*$$

$$H_5 : \mathbb{G}_q \times \{0,1\}^* \longrightarrow \{0,1\}^{l_2}$$

$$H_6 : \mathbb{G}_q \times \mathbb{G}_q \longrightarrow \mathbb{Z}_q^*$$

$$H_7 : \mathbb{G}_q \times \mathbb{G}_q \times \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$$

Here, l_1 and l_2 are the lengths of the bit string and are determined by the security parameter λ.

(4) Publish the system's maser public key mpk $= \{p, q, l_1, l_2, E, G, G_q, P, P_{\text{pub}}, H_1, H_2, H_3, H_4, H_5, H_6, H_7, H_8, H_9, H_{10}\}$ and message space $= \{0,1\}^{l_1}$

4.3.2. Key Generation Phase. In this phase, the *Set-Secret-Value*, *Partial-Key-Extract*, *Set-Private-Key*, and *Set-Public-Key* algorithms are executed. Each user generates his/her own private key and public key pair so that he/she can use the proxy. Furthermore, each user communicates with the KGC to receive a partial key and uses the partial key to generate his/her own public and private key pair, as shown in Figure 6.

(ii) *Set-Secret-Value*: this algorithm is executed by user i. User i randomly selects $t_i \in \mathbb{Z}_q^*$ and maintains security. User i computes $T_i = t_i \bullet P$ as the public key, and user i sends (T_i, ID_i) to the KGC

(iii) *Partial-Key-Extract*: this algorithm is performed by the KGC. According to the identity ID_i of user i, the KGC performs the following steps.

(1) Randomly select $r_i \in \mathbb{Z}_q^*$ and compute $R_i = r_i \bullet P$

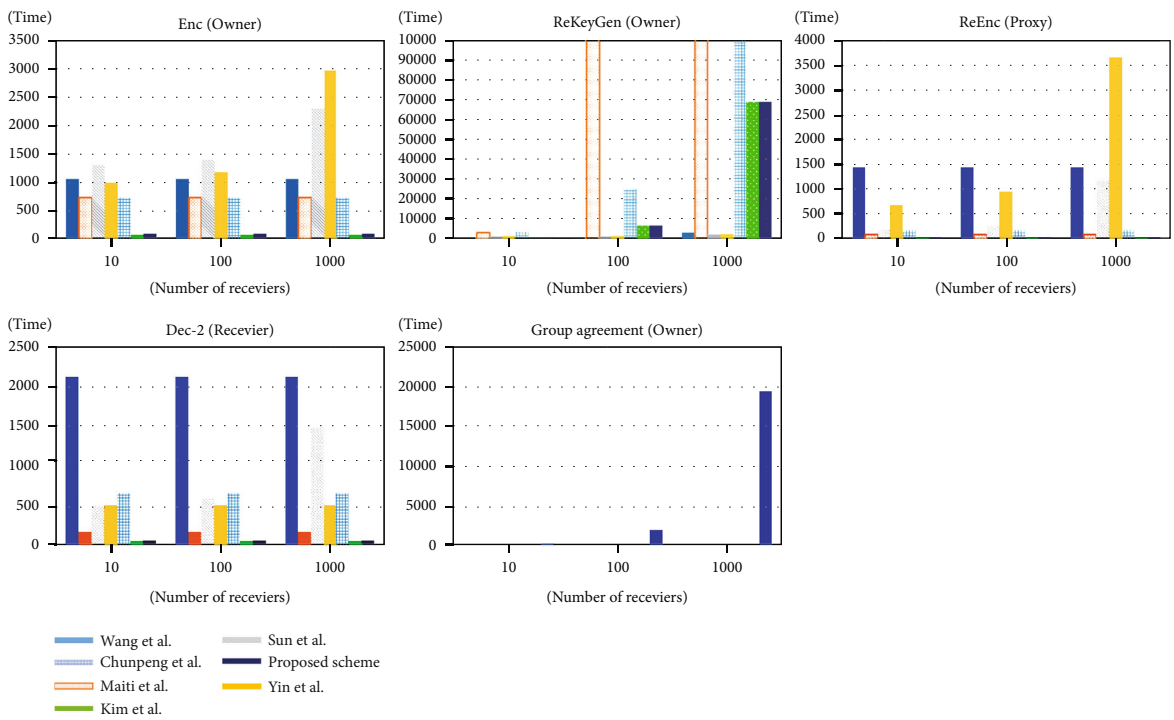


Figure 6: Comparison with other schemes.

(2) Calculate a part of the partial private key k_i as follows:

$$k_i \longleftarrow r_i + dH_{10}(R_i, T_i, \mathrm{ID}_i) + H_8(dT_i, \mathrm{ID}_i)(\mathrm{mod}\ q). \tag{1}$$

(3) After that, partial key (R_i, k_i) is delivered to user i through a public channel

(iv) *Set-Private-Key*: this algorithm is executed by user i. After receiving the partial key (R_i, k_i) from the KGC, user i verifies it as shown in Eqs. (2) and (3). If the key is verified, user i computes the private key $\mathrm{sk}_i = (s_i, t_i)$ as follows:

(4) Verify whether the following equation holds:

$$k_i \bullet P \stackrel{?}{=} R_i + H_7(R_i, T_i, \mathrm{ID}_i)P_{\mathrm{Pub}} + H_5(t_i P_{\mathrm{Pub}}, \mathrm{ID}_i)P. \tag{2}$$

(5) If not, return $\perp$; otherwise, user i compute s_i.

$$s_i \longleftarrow k_i - H_7(t_i P_{\mathrm{Pub}}, \mathrm{ID}_i). \tag{3}$$

(6) Subsequently, user i keeps secret $\mathrm{sk}_i = (s_i, t_i, k_i)$ as his/her full private key

(v) *Set-Public-Key*: this algorithm is performed by user i . User i keeps $\mathrm{pk}_i = (R_i, T_i)$ as a full public key

4.3.3. Group Agreement Phase. This phase includes the *Initialization* and *Group Agreement* algorithms. It represents the process of forming a group of users who jointly own data. Through this process, all users belonging to a group have equal ownership.

(vi) *Group Agreement*: this algorithm is performed by all group members $\mathcal{G}_i$ who will form group $\mathcal{G}$. Each member creates a secret to share with other members using their private sk_i and public keys pk_i. Each member transmits the generated shared secret to other members and generates a group public key $\mathrm{gpk}_{\mathcal{G}}$ and a group private key $\mathrm{gsk}_{\mathcal{G}}$ using the shared secret sent by other members and their own shared secret as follows:

(7) Group member $\mathcal{G}_i$ computes S_i using $sk_i = (s_i, t_i, k_i)$

$$S_i \longleftarrow s_i \cdot P. \tag{4}$$

(8) Group member $\mathcal{G}_i$ computes h_1 and h_2 for each other group member $\mathcal{G}_j$ $(1 \le j \le n, j \ne i)$

$$\begin{aligned} h_1 &\longleftarrow H_3(\mathrm{ID}_i, \mathrm{ID}_j, R_i, R_j, S_i, S_j), \\ h_2 &\longleftarrow H_3(\mathrm{ID}_j, \mathrm{ID}_i, R_j, R_i, S_j, S_i). \end{aligned} \tag{5}$$

(9) Group member $\mathcal{G}_i$ chooses $a_i \in \mathbb{Z}_q^*$ and computes session key ssk_{ij} between $\mathcal{G}_i$ and $\mathcal{G}_j$ and encrypts a_i using a symmetric encryption algorithm

$$\begin{aligned} \mathrm{ssk}_{i,j} &\longleftarrow H_4\big((h_1 t_i + s_i)(h_2 T_j + S_j)\big), \\ x_{i,j} &\longleftarrow E_{\mathrm{ssk}_{i,j}}(a_i). \end{aligned} \tag{6}$$

(10) Group member $\mathcal{G}_i$ sends $x_{i,j}$ $(1 \le j \le n, j \ne i)$ to each group member and receives $x_{j,i}$ $(1 \le j \le n, j \ne i)$ from the other members

(11) All group members of group $\mathcal{G}$ obtain the a_i $(1 \le j \le n)$ generated by each group member through the following operation:

$$\begin{aligned} \mathrm{ssk}_{i,j} &\longleftarrow H_4\big((h_1 T_i + S_i)(h_2 t_j + s_i)\big), \\ a_i &\longleftarrow D_{\mathrm{ssk}_{i,j}}(x_{i,j}). \end{aligned} \tag{7}$$

(12) Group member $\mathcal{G}_i$ computes group private key $gsk_{\mathcal{G}} = t_{\mathcal{G}}$ and group public key $gpk_{\mathcal{G}} = T_{\mathcal{G}}$

$$\begin{aligned} t_{\mathcal{G}} &\longleftarrow a_1 + a_2 + \cdots + a_n, \\ T_{\mathcal{G}} &\longleftarrow t_{\mathcal{G}} \cdot P. \end{aligned} \tag{8}$$

4.3.4. Data Storing Phase. The *Enc* and *Dec*-1 algorithms are executed in this phase. This phase represents the process of group member $\mathcal{G}_i$ encrypting his/her data with the group public key $\mathrm{gpk}_{\mathcal{G}}$ and storing it in a proxy. In addition, group member $\mathcal{G}_i$ downloads his/her own data stored in the proxy, and a decryption process is included using the group private key $\mathrm{gsk}_{\mathcal{G}}$ to obtain the data source again.

(vii) *Enc*: this algorithm is performed by group member $\mathcal{G}_i$. Group member $\mathcal{G}_i$ encrypts message m with ciphertext CT by entering the group public key $\mathrm{gpk}_{\mathcal{G}} = T_{\mathcal{G}}$ and message $m \in M$. Then, the ciphertext CT is uploaded to the proxy

(13) Group member $\mathcal{G}_i$ computes w, z, and Z using given message $m \in M$ and $gpk_{\mathcal{G}} = T_{\mathcal{G}}$

$$\begin{aligned} w &\longleftarrow H_5(T_{\mathcal{G}}, ID_{\mathcal{G}}), \\ z &\longleftarrow H_1(m \| w), \\ Z &\longleftarrow zP. \end{aligned} \tag{9}$$

(14) Group member $\mathcal{G}_i$ chooses $\alpha \in \mathbb{Z}_q^*$ and calculates β, θ, and C as follows:

$$\begin{aligned} \beta &\longleftarrow \alpha \cdot P, \\ \theta &\longleftarrow T_{\mathcal{G}} \cdot \alpha, \\ C &\longleftarrow H_6(Z, \theta) \oplus (m \| w). \end{aligned} \tag{10}$$

(15) Group member $\mathcal{G}_i$ generates the ciphertext $\mathrm{CT} \longleftarrow (C_1, C_2, C_3) = (C, Z, \beta)$. The generated CT is then uploaded and stored as a proxy

(viii) *Dec*-1: this algorithm is performed by group member $\mathcal{G}_i$. Group member $\mathcal{G}_i$ can download the ciphertext $\mathrm{CT} \longleftarrow (C_1, C_2, C_3) = (C, Z, \beta)$ from the proxy. Group member $\mathcal{G}_i$ who has downloaded the ciphertext CT can obtain the plaintext m by decrypting the ciphertext CT with his/her group private key $\mathrm{gsk}_{\mathcal{G}} = t_{\mathcal{G}}$

(16) Group member $\mathcal{G}_i$ calculates θ' by inputting $\mathrm{gsk}_{\mathcal{G}}$ and C_3

$$\theta' \longleftarrow C_3 \cdot t_{\mathcal{G}}. \tag{11}$$

(17) Group member $\mathcal{G}_i$ computes m by inputting C_1, C_2, θ'

$$\begin{aligned} (m \| w) &\longleftarrow C_1 \oplus H_6\left(C_2, \theta'\right), \\ \because \mathrm{C}_1 \oplus \mathrm{H}_6\left(\mathrm{C}_2, \theta'\right) &= \mathrm{H}_6(\mathrm{Z}, \theta) \oplus (\mathrm{m} \| \mathrm{w}) \oplus \mathrm{H}_6\left(\mathrm{C}_2, \theta'\right) \\ &= \mathrm{H}_6(\mathrm{Z}, \theta) \oplus (\mathrm{m} \| \mathrm{w}) \oplus \mathrm{H}_6\left(\mathrm{Z}, \theta'\right) \\ &= (\mathrm{m} \| \mathrm{w}). \end{aligned} \tag{12}$$

(18) Verify whether the following equation holds. If not, return $\perp$; otherwise, group member $\mathcal{G}_i$ keeps the plaintext m

$$\begin{aligned} C_2 &\stackrel{?}{=} H_2(m \| H_5(T_{\mathcal{G}}, ID_{\mathcal{G}}))P, \\ \because \mathrm{C}_2 &= H_1(m \| H_5(T_{\mathcal{G}}, \mathrm{ID}_{\mathcal{G}}))P \\ &= H_1(\mathrm{m} \| \mathrm{w})P = \mathrm{z}P = Z. \end{aligned} \tag{13}$$

4.3.5. Data Broadcast Phase. This phase includes the *Re-Key-Gen*, *Re-Enc*, and *Dec*-2 algorithms. In this phase, group member $\mathcal{G}_i$ generates a reencryption key for a set of recipients and passes it to the proxy. After receiving the reencryption key, the proxy reencrypts the encrypted data and broadcasts them to the recipients. A receiver that has received the broadcast ciphertext can obtain the message by decrypting the ciphertext with its private key.

(ix) *Re-Key-Gen*: in this algorithm, group member $\mathcal{G}_i$ specifies a set of recipients $\mathcal{R} = (r_1, r_2, \cdots, r_n)$ and generates a reencryption key $\mathrm{RK}_{\mathcal{G} \longrightarrow \mathcal{R}}$ to delegate the ciphertext CT

(19) Group member $\mathcal{G}_i$ computes U_j for all receiver r_j $(r_j \in \mathcal{R})$

$$U_j \longleftarrow z \bullet \left(R_j + H_7\left(R_j, T_j, \mathrm{ID}_j\right) P_{\mathrm{pub}} + T_j\right). \tag{14}$$

(20) Group member $\mathcal{G}_i$ computes a polynomial $f(x)$ with degree n using $\gamma \in \mathbb{Z}_q^*$ as follows:

$$\mu \longleftarrow \alpha \cdot \gamma,$$
$$f(x) = \prod_{i=0}^{n}(x - U_j) + \mu \pmod q \quad (15)$$
$$= x^n + \varphi_{n-1}x^{n-1} + \cdots + \varphi_1 x + \varphi_0,$$

where $\varphi_i \in \mathbb{Z}_p^* \ (i = 0, 1, \cdots, n-1)$

(21) Group member $\mathscr{G}_i$ computes ζ using $\text{gsk}_{\mathscr{G}} = t_{\mathscr{G}}$ and γ as follows:

$$\zeta \longleftarrow (\gamma + 1) \cdot t_{\mathscr{G}}. \quad (16)$$

(22) Group member $\mathscr{G}_i$ generates a reencryption key $\text{RK}_{\mathscr{G} \longrightarrow \mathscr{R}} = (\text{rk}_1, \text{rk}_2) = (\zeta, \{\varphi_0, \varphi_1, \cdots, \varphi_{n-1}\})$ and sends $\text{RK}_{\mathscr{G} \longrightarrow \mathscr{R}}$ to the proxy

(x) *Re-Enc*: this algorithm is executed using a proxy. This algorithm reencrypts the ciphertext $\text{CT} \longleftarrow (C_1, C_2, C_3) = (C, Z, \beta)$ into ciphertext CT_R using the reencryption key $\text{RK}_{\mathscr{G} \longrightarrow \mathscr{R}}$

(23) Compute CT_R using ciphertext CT and reencryption key $\text{RK}_{\mathscr{G} \longrightarrow \mathscr{R}}$

$$\begin{aligned} C_1' &\longleftarrow C_1, \\ C_2' &\longleftarrow C_2, \\ C_3' &\longleftarrow C_3 \cdot \text{rk}_1, \\ C_4' &\longleftarrow \text{rk}_2. \end{aligned} \quad (17)$$

(24) Output $\text{CT}_R = (C_1', C_2', C_3', C_4')$ and send CT_R to receivers $\mathscr{R}$

(xi) *Dec*-2: this algorithm is executed by the selected receiver r_j to extract plaintext from the received ciphertext $CT_R = (C_1', C_2', C_3', C_4')$. Receiver r_j performs the following steps:

(25) Compute U_j

$$U_j' \longleftarrow (s_j + t_j) \bullet C_1'. \quad (18)$$

(26) Generate polynomial $f(x)$ and compute β'

$$\begin{aligned} f(x) &= x^n + \varphi_{n-1}x^{n-1} + \cdots + \varphi_1 x + \varphi_0, \\ \mu' &= f\left(U_j'\right). \end{aligned} \quad (19)$$

(27) Compute θ' as an input C_3' and β'

$$\theta' = C_3' - \mu' \cdot T_{\mathscr{G}},$$

$$\begin{aligned} \because C_3' - \mu' \cdot T_{\mathscr{G}} &= C_3 \cdot rk_1 - \alpha \cdot \gamma \cdot T_{\mathscr{G}} \\ &= \beta \cdot \zeta - \alpha \cdot \gamma \cdot T_{\mathscr{G}} \\ &= \alpha \cdot P \cdot (\gamma + 1) \cdot t_{\mathscr{G}} - \alpha \cdot \gamma \cdot T_{\mathscr{G}} \\ &= \alpha \cdot P \cdot \gamma \cdot t_{\mathscr{G}} + \alpha \cdot P \cdot t_{\mathscr{G}} - \alpha \cdot \gamma \cdot T_{\mathscr{G}} \\ &= \alpha \cdot P \cdot t_{\mathscr{G}} = \theta. \end{aligned} \quad (20)$$

(28) Compute m as an input C_1', C_2', θ'

$$(m \| w) \longleftarrow C_1' \oplus H_6\left(C_2', \theta'\right),$$

$$\begin{aligned} \because C_1' \oplus H_6\left(C_2', \theta'\right) &= (m \| w) \oplus H_6(Z, \theta) \oplus H_6\left(C_2', \theta'\right) \\ &= (m \| w) \oplus H_6(Z, \theta) \oplus H_6\left(Z, \theta'\right) \\ &= (m \| w), \end{aligned} \quad (21)$$

where $C_1' = C_1 = Z$

(29) Verify message m. If not, return $\perp$; otherwise, receiver i outputs the plaintext m

$$\begin{aligned} C_2' &\stackrel{?}{=} H_1(m \| w)P, \\ \because C_2' &= H_1(m \| w)P = zP = Z, \end{aligned} \quad (22)$$

where $Z = zP$ and $z = H_1(m \| w)$

5. Analysis of the Proposed G2M BPRE Scheme

In this section, we perform a security analysis and computational analysis of the security requirements of the proposed scheme.

5.1. Analysis of the Security Requirements. In this section, we analyze the security requirements presented in Section 3.2. Here, we analyze the security of the seven security requirements, as shown in Table 1.

(i) *Confidentiality*: this proposed method performs an encryption operation based on elliptic curve encryption. Because elliptic curve encryption provides high security, even with a short key, efficient encryption is possible. The proposed method uses this elliptic curve encryption method such that a user without a decryption key cannot know the contents of the data. First, the proposed method encrypts a message using a public key:

Table 1: Comparison of the security requirements.

	Group ownership	Bilinear pairing	Key escrow problem	Receiver anonymity	Re-Key genneration
Wang and Wang [53]	Not provided	Used	Insecure	Offer	KGC/BC
Maiti and Misra [54]	Not provided	Used	Insecure	Offer	Sender
Sun et al. [55]	Not provided	Used	Insecure	Offer	Sender
Yin et al. [56]	Not provided	Used	Insecure	Offer	Sender
Chunpeng et al. [57]	Not provided	Used	Insecure	Offer	Sender
Kim et al. [51]	Not provided	Not used	Secure	Offer	Sender
Proposed scheme	Provided	Not used	Secure	Offer	Sender

$$\begin{aligned} w &\longleftarrow H_5(T_{\mathscr{G}}, ID_{\mathscr{G}}), \\ z &\longleftarrow H_1(m\|w), \\ Z &\longleftarrow zP, \\ \beta &\longleftarrow \alpha \cdot P, \\ \theta &\longleftarrow T_{\mathscr{G}} \cdot \alpha, \\ C &\longleftarrow H_6(Z, \theta) \oplus (m\|w). \end{aligned} \tag{23}$$

Here, message encryption is performed by the XOR operation, and θ in the XOR operation is created with the owner's public key. In addition, the owner's private key is required to create θ using the ciphertext C_3. Accordingly, the ciphertext of the proposed method can only be decrypted with the group private key $\text{psk}_{\mathscr{G}}$ paired with the group public key $\text{gpk}_{\mathscr{G}}$ used for encryption.

(ii) *Integrity*: recipients who decrypt the data can verify the integrity of the data using the values contained in the integrity ciphertext and parameters of the public KGC. The proofing methods are as follows.

$$\begin{aligned} C_2' &\stackrel{?}{=} H_1(m\|w)P, \\ \because C_2' &= H_1(m\|w)P = zP = Z, \end{aligned} \tag{24}$$

where $Z = zP$ and $z = H_1(m\|w)$.

The receiver that decrypts the ciphertext CT_R can obtain message m and verification value w. Here, $H_1(m\|w)$ is equal to z; thus, the integrity of the message can be verified by determining whether $H_1(m\|w)P$ is equal to $C_2 = Z$.

(iii) *Key escrow problem*: in the certificate-based public key encryption method, a certificate corresponding to the public key must be issued and stored. To solve this problem, a certificateless public-key encryption method may be used. However, in the general certificate public-key encryption method, the KGC generates and delivers the user's private key. Thus, because the KGC user's complete private key is known, the key escrow problem of the KGC may occur. In this study, an algorithm is designed using the partial-key method to solve this problem

First, the user creates his/her secret value t_i, converts it into T_i, and transmits it to the KGC. Upon receiving T_i, KGC generates a secret value r_i for the user, generates k_i through the following calculation process, and delivers (R_i, k_i) to the user.

$$\begin{aligned} R_i &= r_i \bullet P, \\ k_i &\longleftarrow r_i + dH_7(R_i, T_i, \text{ID}_i) + H_5(dT_i, \text{ID}_i)(\text{mod } q). \end{aligned} \tag{25}$$

The user who receives (R_i, k_i) from the KGC calculates s_i using k_i and t_i known only to the user as follows:

$$s_i \longleftarrow k_i - H_7(t_i P_{\text{Pub}}, \text{ID}_i). \tag{26}$$

Thereafter, the user uses (s_i, t_i, k_i) as private keys and (R_i, T_i) as public keys.

Finally, T_i generated by the user and R_i generated by the KGC are used as public keys. Consequently, the partial key known to the KGC and the unknown partial key are as follows:

KGC only knows $pk_i = (T_i, R_i)$ and k_i

KGC cannot knows $sk_i = (s_i, t_i)$

(iv) *Partial key verifiability*: the proposed scheme uses a partial key in the key generation process to solve the key-escrow problem. However, it is possible for the malicious KGC to deliver the generated partial key with a value other than the T_i passed to the KGC by the user. To solve this problem, the proposed scheme provides a partial key verification function through the following operation:

$$\begin{aligned} k_i \bullet P &\stackrel{?}{=} R_i + H_7(R_i, T_i, \text{ID}_i)P_{\text{Pub}} + H_5(t_i P_{\text{Pub}}, \text{ID}_i)P, \\ \because k_i \bullet P &= r_i \bullet P + H_7(R_i, T_i, \text{ID}_i) \cdot \text{d} \cdot P + H_5(t_i P_{\text{Pub}}, \text{ID}_i)P \\ &= (r_i + H_7(R_i, T_i, \text{ID}_i) \cdot d + H_5(t_i \cdot \text{d} \cdot P, \text{ID}_i))P \\ &= (r_i + \text{d} \cdot H_7(R_i, T_i, \text{ID}_i) + H_5(T_i \cdot d_i, \text{ID}_i))P = (k_i)P, \end{aligned} \tag{27}$$

where $k_i = r_i + dH_7(R_i, T_i, \text{ID}_i) + H_5(dT_i, \text{ID}_i)$,

$$R_i = r_i P, T_i = t_i P, P_{\text{pub}} = dP. \tag{28}$$

Table 2: Comparison of the computation efficiency.

	Enc	Re-Key gen	Re-Enc	Dec-2	Group agreement
Wang and Wang [53]	$(2)T_M + (4)T_e + (1)T_P$	$(10 + 3n)T_M + (1)T_e$	$(6)T_e$	$(7)T_M + (7)T_e + (5)T_P$	—
Maiti and Misra [54]	$(4)T_M + (3)T_e$	$(3 + n^2 + n)T_M + (3 + n)T_e$	$(1)T_M + (1)T_P$	$(1)T_M + (2)T_P$	—
Sun et al. [55]	$(2 + \mathrm{n})T_M + (5)T_e + (1)T_P$	$(3 + n)T_M + (6)T_e + (1)T_p$	$(1 + \mathrm{n})T_M + (2)T_P$	$(4 + \mathrm{n})T_M + (2)T_e$	—
Yin et al. [56]	$(4 + 2n)T_M + (4)T_e$	$(4 + 2n)T_M + (4)T_e$	$(4 + 3n)T_M + (2)T_e + (2)T_p$	$(7)T_M + (1)T_e + (3)T_P$	—
Chunpeng et al. [57]	$(2)T_M + (3)T_e$	$(5 + n)T_M + (5 + n)T_e + (1)T_p$	$(1)T_M + (2)T_p$	$(6)T_M + (2)T_e + (2)T_p$	—
Kim et al. [51]	$(2)T_{\mathrm{EM}} + (2)T_{\mathrm{EA}}$	$(2 + n)T_M + (2n)T_{\mathrm{EM}} + (2n)T_{\mathrm{EA}}$	$(1)T_{\mathrm{EM}}$	$(2)T_{\mathrm{EM}}$	—
Proposed scheme	$(3)T_{\mathrm{EM}}$	$(1 + n)T_M + (2n)T_{\mathrm{EM}} + (2n)T_{\mathrm{EA}}$	$(1)T_{\mathrm{EM}}$	$(2)T_{\mathrm{EM}} + (1)T_{\mathrm{EA}}$	$(1 + 6n))T_{\mathrm{EM}} + (4(n - 1))T_{\mathrm{EA}}$

T_M: computation time of modular multiplication operation; T_{EM}: computation time of ECC multiplication operation; T_{EA}: computation time of ECC point add operation; T_e: computation time of exponent operation; T_p: computation time of bilinear pairing operation.

(v) *Receiver anonymity*: in the proposed scheme, the public key and ID of the recipient are used to designate multiple recipients. This method was designed based on multireceiver encryption. However, in the existing multireceiver encryption, other users can identify the recipient because the ciphertext contains information that can identify the recipient. To solve this problem, in this study, a receiver identification process was designed using a polynomial, as follows:

$$\begin{aligned} f(x) &= \prod_{i=0}^{n}(x - U_i) + \beta \pmod q \\ &= (x - U_1)\bullet(x - U_2)\bullet\cdots\bullet(x - U_n) \\ &\quad + \mu \pmod q \\ &= x^n + \varphi_{n-1}x^{n-1} + \cdots + \varphi_1 x + \varphi_0, \\ \mathrm{U}_j' &= \left(\mathrm{s}_j + \mathrm{t}_j\right)\bullet \mathrm{C}_1'. \end{aligned} \tag{29}$$

It is possible to generate U_i' of the receiver to identify a specific recipient in the above polynomial. However, as in the confidentiality item above, an attacker cannot forge U_i'.

$$U_i' \longleftarrow \overset{?}{z} \bullet \left(R_i + H_7(R_i, T_i, ID_i)P_{\mathrm{pub}} + T_i\right). \tag{30}$$

As a result, the attacker cannot identify the recipient.

(vi) *Decryption fairness*: as described in the receiver anonymity section, each receiver's public key and ID are used to designate multiple receivers. However, in the design process, there is a threat that a specific receiver performs more operations during the decoding process or makes decoding impossible. This is known as the decryption fairness problem. Such problems can be caused by removing or changing some elements in the data that specify and validate the recipient. In the proposed scheme, an algorithm is designed using polynomials to address this problem. These polynomials, which can only be changed and falsified by the user who created them, are as follows:

$$\begin{aligned} f(x) &= \prod_{i=0}^{n}(x - U_i) + \beta \pmod q \\ &= (x - U_1)\bullet(x - U_2)\bullet\cdots\bullet(x - U_n) + \mu \pmod q \\ &= x^n + \varphi_{n-1}x^{n-1} + \cdots + \varphi_1 x + \varphi_0. \end{aligned} \tag{31}$$

5.2. Analysis of Computational Efficiency. The scheme proposed in this study was designed to provide extended functions based on the method proposed by Kim et al. Accordingly, its overall structure is similar to that reported by Kim et al., but its detailed calculations are different. As shown in Figure 6 and Table 2, the computation time of the proposed scheme is almost the same as that of Kim et al. There are differences in some calculations; however, they are not so large in terms of the total number of calculations. In addition, compared with other schemes, the reencryption key generation algorithm requires a relatively larger amount of computation time than the other algorithms in the scheme. In addition, in the proposed scheme, a group agreement algorithm is additionally used to provide a group joint ownership function. Accordingly, although its total computation time is greater than that of other schemes, the proposed method is able to perform group-owned functions that cannot be executed by other schemes.

6. Conclusion

This study examined the extended form of proxy reencryption. Existing proxy reencryption technology provides a data delegation method that assumes one owner and one receiver. It provides an intuitive and clear form of data communication. However, owing to recent technological developments, an environment in which multiple devices exchange data, such as device-to-device communication, rather than human-to-human communication, is becoming common. A typical example of this is the IoT environment. The IoT environment is an environment in which multiple devices communicate with each other and share and use data for various purposes. However, in this environment, existing proxy reencryption for 1 : 1 communication is inevitably inefficient. In an IoT environment, where the same data must be delivered to multiple devices in the same way, when using the existing proxy reencryption, the same data must be reencrypted several times. This method inevitably reduces the data transfer efficiency. In addition, in a large-scale communication environment, an environment in which multiple users form a group to create and own data can be presented. However, because the existing proxy reencryption is a form in which only one user can be the owner, data ownership disputes may arise. To solve this problem, this study proposes proxy reencryption, which can support multiple owners and recipients. In addition, to increase the security and efficiency of the proposed technology, only elliptic curve encryption is used, and security is improved using the partial key form. However, because the proposed scheme uses a group key method that has not been used in other existing schemes, the group agreement algorithm is additionally applied and requires a relatively large amount of computation time. As a result, the proposed method provides more functions than the existing proxy reencryption and improved security; however, it requires additional computation. This method can be used more effectively in environments in which scalability is more important than computational efficiency.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Won-Bin Kim and Su-Hyun Kim contributed equally to this work.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A2B5B01002490) and Soonchunhyang University Research Fund

References

[1] Gartner, "Gartner top strategic technology trends for 2022," Technical report, Gartner, 2022.

[2] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 127–144, Espoo, Finland, 1998.

[3] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[4] Z. Cai, X. Zheng, J. Wang, and Z. He, "Private data trading towards range counting queries in Internet of Things," *IEEE Transactions on Mobile Computing*, pp. 1–17, 2022.

[5] J. Byabazaire, G. O'Hare, and D. Delaney, "Data quality and trust: review of challenges and opportunities for data sharing in iot," *Electronics*, vol. 9, no. 12, p. 2083, 2020.

[6] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, 2019.

[7] G. Ateniese, F. Kevin, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security (TISSEC)*, vol. 9, no. 1, pp. 1–30, 2006.

[8] R. H. Deng, J. Weng, S. Liu, and K. Chen, "Chosen- ciphertext secure proxy re-encryption without pairings," in *International Conference on Cryptology and Network Security*, pp. 1–17, Hong-Kong, China, 2008.

[9] B. Libert and D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy re-encryption," in *11th International Workshop on Practice and Theory in Public-Key Cryptography*, pp. 360–379, Barcelona, Spain, 2008.

[10] J. Shao and Z. Cao, "Cca-secure proxy re-encryption without pairings," in *International Workshop on Public Key Cryptography*, pp. 357–376, Irvine, CA, USA, 2009.

[11] G. Ateniese, K. Benson, and S. Hohenberger, "Key-private proxy re-encryption," in *Cryptographers' Track at the RSA Conference*, pp. 279–294, San Francisco,CA, USA, 2009.

[12] S. S. M. Chow, J. Weng, Y. Yang, and R. H. Deng, "Efficient unidirectional proxy re-encryption," in *International Conference on Cryptology in Africa*, pp. 316–332, Stellenbosch, South Africa, 2010.

[13] J. Shao, P. Liu, G. Wei, and Y. Ling, "Anonymous proxy re-encryption," *Security and Communication Networks*, vol. 5, no. 5, p. 449, 2012.

[14] H. Wang and Z. Cao, "More efficient cca-secure unidirectional proxy re-encryption schemes without random oracles," *Security and Communication Networks*, vol. 6, no. 2, p. 181, 2013.

[15] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.

[16] G. Hanaoka, Y. Kawai, N. Kunihiro, T. Matsuda, J. Weng, R. Zhang, and Y. Zhao, Eds., "Generic construction of chosen ciphertext secure proxy re-encryption," in *Cryptographers' Track at the RSA Conference*, pp. 349–364, San Francisco, CA, USA, 2012.

[17] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop on the Theory and Application of Cryptographic Techniques*, pp. 47–53, Springer, 1985.

[18] C.-K. Chu and W.-G. Tzeng, "Identity-based proxy re- encryption without random oracles," in *International Conference on Information Security*, pp. 189–202, Valparaiso, Chile, 2007.

[19] M. Green and G. Ateniese, "Identity-based proxy re- encryption," in *International Conference on Applied Cryptography and Network Security*, pp. 288–306, Zhuhai, China, 2007.

[20] K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing," in *European symposium on research in computer security*, pp. 257–272, Wroclaw, Poland, 2014.

[21] A. Paul, S. Varshika Srinivasavaradhan, S. D. Selvi, and C. P. Rangan, "A ca-secure collusion-resistant identity-based proxy re-encryption scheme," in *International Conference on Provable Security*, pp. 111–128, Jeju, South Korea, 2018.

[22] L. Wang, L. Wang, M. Mambo, and E. Okamoto, "New identity-based proxy re-encryption schemes to prevent collusion attacks," in *International Conference on Pairing-Based Cryptography*, pp. 327–346, Yamanaka Hot Spring, Japan, 2010.

[23] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *International conference on the theory and application of cryptology and information security*, pp. 452–473, Taipei, Taiwan, 2003.

[24] X. Lei, W. Xiaoxin, and X. Zhang, "Cl-pre: a certificateless proxy re-encryption scheme for secure data sharing with public cloud," in *Proceedings of the 7th ACM symposium on information, computer and communications security*, pp. 87-88, Seoul Korea, 2012.

[25] W. Xiaoxin, X. Lei, and X. Zhang, "Poster: a certificateless proxy re-encryption scheme for cloud-based data sharing," in *Proceedings of the 18th ACM conference on computer and communications security*, pp. 869–872, Chicago Illinois USA, 2011.

[26] K. Yang, X. Jing, and Z. Zhang, "Certificateless proxy re-encryption without pairings," in *International Conference on Information Security and Cryptology*, pp. 67–88, Seoul, Korea, 2014.

[27] X. Zheng, Y. Zhou, Y. Ye, and F. Li, "A cloud data deduplication scheme based on certificateless proxy re-encryption," *Journal of Systems Architecture*, vol. 102, article 101666, 2020.

[28] J. Baek, R. Safavi-Naini, and W. Susilo, "Efficient multi-receiver identity-based encryption and its application to broadcast encryption," in *International Workshop on Public Key Cryptography*, pp. 380–397, Springer, 2005.

[29] S. Chatterjee and P. Sarkar, "Multi-receiver identity-based key encapsulation with shortened ciphertext," in *International Conference on Cryptology in India*, pp. 394–408, Kolkata, India, 2006.

[30] P. Vijayakumar, S. Bose, A. Kannan, and L. J. Deborah, "Computation and communication efficient key distribution protocol for secure multicast communication," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 4, pp. 878–894, 2013.

[31] I. Kim and S. O. Hwang, "An optimal identity-based broadcast encryption scheme for wireless sensor networks," *IEICE Transactions on Communications*, vol. E96.B, no. 3, pp. 891–895, 2013.

[32] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[33] J. Kim, W. Susilo, A. Man Ho, and J. Seberry, "Adaptively secure identity-based broadcast encryption with a constant-sized ciphertext," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 679–693, 2015.

[34] F.-C. Zhou, M.-Q. Lin, Y. Zhou, and Y.-X. Li, "Efficient Anonymous broadcast encryption with adaptive security," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 11, pp. 4680–4700, 2015.

[35] J. Li, Y. Qihong, and Y. Zhang, "Identity-based broadcast encryption with continuous leakage resilience," *Information Sciences*, vol. 429, pp. 177–193, 2018.

[36] J. Lai, M. Yi, F. Guo, P. Jiang, and S. Ma, "Identity- based broadcast encryption for inner products," *The Computer Journal*, vol. 61, no. 8, pp. 1240–1251, 2018.

[37] C.-I. Fan, L.-Y. Huang, and P.-H. Ho, "Anonymous multireceiver identity-based encryption," *IEEE Transactions on Computers*, vol. 59, no. 9, pp. 1239–1249, 2010.

[38] Y. Huaqun Wang, Z.,. H. Xiong, and B. Qin, "Cryptanalysis and improvements of an anonymous multi-receiver identity-based encryption scheme," *IET Information Security*, vol. 6, no. 1, pp. 20–27, 2012.

[39] H.-Y. Chien, "Improved anonymous multi-receiver identity-based Encryption," *The Computer Journal*, vol. 55, no. 4, pp. 439–446, 2012.

[40] C.-I. Fan, P.-J. Tsai, J.-J. Huang, and W.-T. Chen, "Anonymous multi-receiver certificate-based encryption," in *2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 19–26, Beijing, China, 2013.

[41] M. Zhang and T. Takagi, "Efficient constructions of anonymous multireceiver encryption protocol and their deployment in group e-mail systems with privacy preservation," *IEEE Systems Journal*, vol. 7, no. 3, pp. 410–419, 2013.

[42] J. Zhang and J. Mao, "An improved anonymous multi-receiver identity-based encryption scheme," *International Journal of Communication Systems*, vol. 28, no. 4, pp. 645–658, 2015.

[43] C. Sur, C. D. Jung, and K. H. Rhee, "Multi-receiver certificate-based encryption and application to public key broadcast encryption," in *2007 ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security (BLISS 2007)*, pp. 35–40, Edinburgh, UK, 2007.

[44] C. Sur, Y.-H. Park, and K.-H. Rhee, "A multi-receiver certificateless encryption scheme and its application," *Journal of Korea Multimedia Society*, vol. 14, no. 6, pp. 775–784, 2011.

[45] S. K. Hafizul Islam, M. K. Khan, and A. M. Al-Khouri, "Anonymous and provably secure certificateless multireceiver encryption without bilinear pairing," *Security and Communication Networks*, vol. 8, no. 13, p. 2231, 2015.

[46] Y.-H. Hung, S.-S. Huang, Y.-M. Tseng, and T.-T. Tsai, "Efficient anonymous multireceiver certificateless encryption," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2602–2613, 2017.

[47] D. He, H. Wang, L. Wang, J. Shen, and X. Yang, "Efficient certificateless anonymous multi-receiver encryption scheme for mobile devices," *Soft Computing*, vol. 21, no. 22, pp. 6801–6810, 2017.

[48] L. Deng, "Anonymous certificateless multi-receiver encryption scheme for smart community management systems," *Soft Computing*, vol. 24, no. 1, pp. 281–292, 2020.

[49] J. Zhu, L.-L. Chen, X. Zhu, and L. Xie, "A new efficient certificateless multi-receiver public key encryption scheme," *International Journal of Computer Science Issues*, vol. 13, no. 6, pp. 1–7, 2016.

[50] E. K. Win, T. Yoshihisa, Y. Ishi, T. Kawakami, Y. Teranishi, and S. Shimojo, "A lightweight multi-receiver encryption scheme with mutual authentication," in *2017 IEEE 41st annual computer software and applications conference (COMPSAC)*, pp. 491–497, Turin, Italy, 2017.

[51] W.-B. Kim, S.-H. Kim, D. Seo, and I.-Y. Lee, "Broadcast proxy reencryption based on certificateless public key cryptography for secure data sharing," *Wireless Communications and Mobile Computing*, vol. 2021, 16 pages, 2021.

[52] A. Braeken, "Pairing free certified common asymmetric group key agreement protocol for data sharing among users with different access rights," *Wireless Personal Communications*, vol. 121, no. 1, pp. 307–318, 2021.

[53] X. Wang and X. Yang, "Identity based broadcast encryption based on one to many identity based proxy re-encryption," in *2009 2nd IEEE International Conference on Computer Science and Information Technology*, Beijing, China, 2009.

[54] S. Maiti and S. Misra, "P2B: privacy preserving identity-based broadcast proxy re-encryption," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5610–5617, 2020.

[55] M. Sun, C. Ge, L. Fang, and J. Wang, "A proxy broadcast re-encryption for cloud data sharing," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 10455–10469, 2018.

[56] S. Yin, H. Li, and L. Teng, "A novel proxy re-encryption scheme based on identity property and stateless broadcast encryption under cloud environment," *International Journal of Network Security*, vol. 21, no. 5, pp. 797–803, 2019.

[57] G. Chunpeng, Z. Liu, J. Xia, and F. Liming, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, pp. 1214–1226, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4518599, 12 pages
https://doi.org/10.1155/2022/4518599

Research Article

A Data-Secured Intelligent IoT System for Agricultural Environment Monitoring

Qing Zhou,[1] Minghua Xiao,[1] Lei Lu,[1] Jun Zeng,[1] Wenting He,[1] Chao Li,[2] and Yulun Shi[3]

[1]*School of Information Engineering, Jiangxi College of Applied Technology, Ganzhou, China*
[2]*Zhijiang College, Zhejiang University of Technology, Hangzhou, China*
[3]*Sunyard System Engineering Co., Ltd., Hangzhou, China*

Correspondence should be addressed to Chao Li; alexlee779@outlook.com

Received 29 January 2022; Accepted 14 June 2022; Published 28 June 2022

Academic Editor: Liran Ma

Collecting environmental information of crop growth and dynamically adjusting agricultural production has been proved an effective way to improve the total agricultural yield. Agricultural IoT technology, which integrates the information sensing equipment, communication network, and information processing systems, can support such an intelligent manner in the agricultural environment. Traditional agricultural IoT could meet the service demand of small-scale agricultural production scenarios to a certain extent. However, the emerging application scenario of the agricultural environment is becoming more and more complicated, and the data nodes of the underlying access to IoT backend system are increasing in large number, while the upper-layer applications are requiring high quality of data service. Hence, the traditional architecture-based (i.e., centralised cloud computing) IoT systems suffer from the problems such as small network coverage, data security issue, and limited power supply time while attempting to provide high-quality services at the edge of the network. Emerging edge computing offers the opportunity to solve these issues. This paper builds an intelligent IoT system for agricultural environment monitoring by integrating edge computing and artificial intelligence. We conducted an experiment to validate the proposed system considering the reliability and usability. The experimental results prove the system's reliability (e.g., data packet loss rate is less than 0.1%). The proposed system achieves the concurrency of 500TPS and the average response time of 300 ms, which meet the practical requirements in agricultural environment monitoring.

1. Introduction

Recently, the Internet of Things (IoT) is being applied to several fields, such as agriculture, logistics, and transportation [1–3]. Using various types of integrated microsmall sensors, wireless sensor networks (WSNs) can achieve real-time detection, acquisition, and sense of various objects [4]. The transmission of various information in physical space (such as temperature and humidity, moisture, and pressure) can make users intuitively understand the information. In China, the establishment of agricultural IT infrastructures is still in its infancy. Therefore, it is impossible to obtain timely information on all farms and different locations of crop growth environment parameters. In order to detect the temperature and suffocation environment, the environmental instruments equipped in the farms are manually operated on-site, which is time-consuming and inefficient. This situation makes the farmers monitor and control the climatic conditions are adversely affected, which in turn affects the improvement of crop yield and quality [5]. Therefore, the method to improve the accuracy and effectiveness of the crop growth environment in the acquisition of various environmental parameters is expected.

On the other hand, the operation of various types of environmental control equipment to achieve intelligent operation and remote control has become a growing concern. The use of the IoT, cloud computing, big data, and other information technology promotes the transformation and upgrading of the entire agricultural industry chain and vigorously drives the development of intelligent agriculture. IoT can effectively

reduce human consumption and accurately capture crop environment and other information and timely carry out actions. But traditional IoT platforms usually adopt a centralised architecture, in which the network bandwidth and processing capacity of the central node could be the bottlenecks for the horizontal expansion of the system [6]. The IoT network edges of large-scale heterogeneous generate massive amounts of heterogeneous data. The long network links for data access operations reduce the performance and efficiency of centralised data storage architectures. The resource constraints of IoT nodes make them dependent on the IoT platform to provide rich services to the external, and the long network links for edge nodes to access IoT services under the centralised IoT architecture make the network latency high, which makes it difficult for edge IoT nodes to get real-time services [7, 8]. By incorporating the computing model of edge computing (EC), an edge processing layer is employed at the near-device end, effectively reducing the workload of network and computation. Edge clouds rely on shorter network links with service invokers, making it possible to provide low-latency network services and low-cost resource access compared to the traditional cloud computing model. However, the edge cloud itself is limited in resources and can easily become a bottleneck for the platform's services on the edge side of the network.

For agricultural IoT, the introduction of EC means that many tasks that used to need to be processed in the cloud can be done locally with artificial intelligence algorithms and data fusion algorithms and can greatly accelerate the response speed of agricultural information and improve monitoring accuracy and more targeted development of agricultural environment management strategies in the monitoring coverage area. This paper combines the Internet of Things and its artificial intelligence technology to build a wide coverage, low power consumption IoT monitoring system suitable for monitoring agricultural environment. It achieves unified management of IoT resources and cooperative computing for environmental monitoring tasks. In this design, the contextual specificity of the environmental monitoring, coupled with the high requirement of real-time data processing, a new agricultural, environmental monitoring IoT architecture model is highlighted. The main contribution in this paper are summarised as follows:

(1) Techniques related to building an edge computing platform for environment monitoring were investigated. The functional architecture and data communication architecture of the edge computing gateway were studied

(2) A collaborative IoT cloud-edge architecture was proposed to realise the unified heterogeneous resource management. It enables the compatibility of the resource identification and mapping mechanism to various kinds of IoT identification standards and realises the unification of the platform resource description methods to reduce the complexity of resource description

(3) The application of LSTM-based environmental indicator prediction algorithm in environmental monitoring was explored. Besides, a visualization dashboard for agricultural environmental monitoring data is built. The data collected by all monitoring nodes are displayed in a dynamic visualization, and the corresponding decision-making support was enabled by the predicting module

The rest of this paper is organised as follows: Chapter 2 introduces the related work, Chapter 3 presents the system architecture and functional details, Chapter 4 presents the experiment and discussion of the proposed system, and Chapter 5 concludes the work.

2. Related Work

2.1. Internet of Things. The Internet of Things (IoT) was first proposed by Prof. Kevin Ashton of the MIT Auto-ID Lab in 1999. IoT was initially designed to solve the problems in supply chain management [9]. The traditional IoT platform architecture usually consists of a data sensing layer, network layer, and business logic layer [10]. Such a centralised architecture has the advantages of easy construction and efficient resource management when the scale of the system is small-size or medium-size. However, with the expansion of the IoT system, service demand is increasing (e.g., response time, intelligent analysis, data storage, and privacy protection). With the popularity of 5G technology, this demand further increases [11, 12], a trusted computer or cluster of computers deployed at the edge of a network with rich service resources to provide computing and data storage services to nearby mobile devices. In 2011, Bonomi et al. proposed the concept of fog computing [13], which introduces a fog computing layer between the device and the cloud. It uses the local fog devices (e.g., routers, IP video cameras, and switches) to process some task requests in close proximity, thereby reducing the number of tasks transmitted to remote cloud computing centres. In 2013, Ryan proposed the early concept of edge computing [14] to address the problem of the rapidly growing number of mobile edge devices. In recent years, distributed IoT architecture based on edge computing has attracted the attention from many researchers [15–17]. Existing studies or edge computing platforms only consider a single edge cloud's vertical application in an IoT scenario, without considering multiple edge clouds in heterogeneous scenarios. Guo et al. built the first virtual fencing system based on a wireless sensor network and implemented a research test for automatic grazing of arable cattle [18]. Vijayakumar and Ramya designed a low-cost and real-time water quality monitoring system by a wireless sensor network, which requires no wiring and has the advantages of flexible deployment and low cost [19]. [20] explored the ZigBee technology in significant field conditions to ensure stable operation of wireless transmission in irrigation area environment. LoRaWAN is designed for long-range communication and networking devices using LoRa (Long Range Radio) technology and can be independently networked from wireless operators [21]. Sendra

et al. develop a monitoring system for large-scale farming, which can measure temperature, relative humidity, wind speed, and carbon dioxide in the farming environment [22]. Valente et al. developed a LoRaWAN-based terminal node equipped with a cluster of asynchronous serial protocol sensors that can measure environmental parameters (including atmospheric pressure, lightning strike count, and soil conductivity) [23].

2.2. Agricultural IoT and Intelligence Computing. Agricultural IoT is the domain application of IoT technology in the whole industry chain of production, management, operation, and service in agriculture [24, 25]. The most common agricultural IoT is used for production environment monitoring [19]. IoT technology is used to collect and obtain information of different elements in the agricultural production environment, including temperature and humidity, light, carbon dioxide, soil water content, and soil fertility. Early IoT applications focused on agricultural information sensing. For example, Wang et al. built a mobile observation system for bovine animals using pulse oximeters, respiratory sensors, body temperature sensors, environmental sensors, and GPS modules [26], which provided a monitoring tool to prevent the spread of diseases in the herd. González et al. developed a method to perform unsupervised behavioural classification by installing GPS sensors and movement collars on cattle to observe and record foraging [27]. Gill et al. propose a cloud-based information system that provides agriculture-as-a-service using cloud and big data technologies [28]. It collects information from different users through preconfigured devices and IoT sensors and processes it in the cloud using big data analytics. Zhu et al. design a dedicated IoT platform in precision agriculture and ecological monitoring [29]. The massive amount of data generated by agricultural, environmental monitoring exhibits complex and dynamic characteristics, and it usually involves multiple sectors, regions, and domains.

With the advancement of artificial intelligence, advanced scientific data processing algorithms are applied in multiple aspects such as air quality prediction and pollution source location. Environmental data are often a series of observations obtained from various physical quantities observed in temporal order, reflecting the characteristics of entity attributes over time, i.e., a multidimensional time series [30]. Time series usually carry a specific law of variation, which is determined by the intrinsic physical properties of the monitored indicators. Time series prediction refers to the process of mining the intrinsic law of change through a large amount of series data and predicting the next point in time based on this law. Popular time series algorithms, including ARIMA [31], etc. [32], used wavelet decomposition and reconstruction to smooth the time series and demonstrated its feasibility for atmospheric pollutant concentration analysis. The change of environmental information involves multiple factors and has nonlinear characteristics, which significantly impacts the accuracy of environmental information prediction. Neural network-based methods have the advantages of self-learning and self-evolving neurons, which are good at dealing with nonlinear models. Artificial neural networks have shown better performance in environmental prediction, and the first one used by related scholars was BP neural network [33], which is with simple structure. Still, it cannot record the features of the previous moment or multiple moments to be used as learning, which leads to the poor prediction and poor generalisation. RNN [34] and many other neural networks have been applied to environmental information prediction, and certain improvements have been achieved. The various environmental monitoring parameters are affected by many factors such as climatic conditions and geographic conditions and do not show linear characteristics.

3. Data-Secured Intelligent Edge Cloud Architecture for IoT

As shown in Figure 1, the workflow is clearly described. This system integrates various kinds of sensors, RFID, video, and other sensing and monitoring devices to collect specific information of farm. This system introduces a cloud-edge collaboration mechanism to process, analyse, and store data to improve the efficiency of network bandwidth utilisation and guarantee the high quality of the platform's external services. The system integrates wireless sensor networks and achieves stable and reliable data transmission through 5G networks. The system fuses and processes the obtained massive agricultural data and realises fully automated monitoring and intelligent analysis of agrarian environment in combination with intelligent terminals. The system's main objectives are as follows: (1) to realise unified management of heterogeneous IoT resources. The proposed architecture uses mapping technology to achieve compatibility with various IoT identity standards and adopts customised identity within the system. A unified resource descriptor is realised by abstracting the behaviour and attributes of resources. (2) To realise a cloud-edge computing services. Data resources are exchanged to each edge cloud through the cloud computing centre. The services of each edge cloud are integrated to provide intelligent decision support for agricultural environment monitoring through intelligent algorithms for hierarchical processing and computing massive data.

3.1. Cloud-Edge Collaborative Architecture. The platform provides comprehensive IoT perceptive ability. The terminal is equipped with ULG series collection and transmission integrated equipment and a set of crop growth related sensors (e.g., soil moisture sensor, soil pH sensor, air temperature and humidity sensor, soil temperature and humidity sensor, light intensity sensor, and carbon dioxide concentration sensor). The platform's architecture consists of an intelligent sensing layer, cloud-edge collaboration layer, heterogeneous network layer, business logic layer, and human-machine interface layer (as shown in Figure 2).

The sensing layer contains sensor monitoring nodes. The monitored data are transmitted to the edge computing gateway through wireless transmission protocols such as LPWAN and 802.11 g. The transport layer adopts heterogeneous networking rules based on LoRaWAN and Wi-Fi and

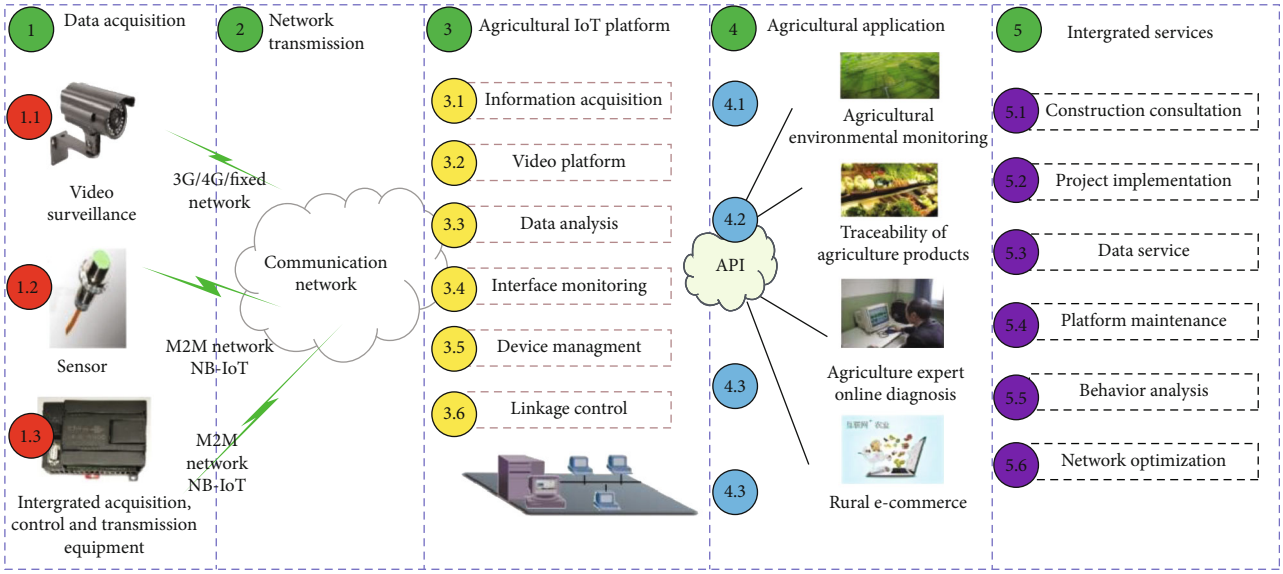


Figure 1: The overview of the agricultural IoT platform workflow.

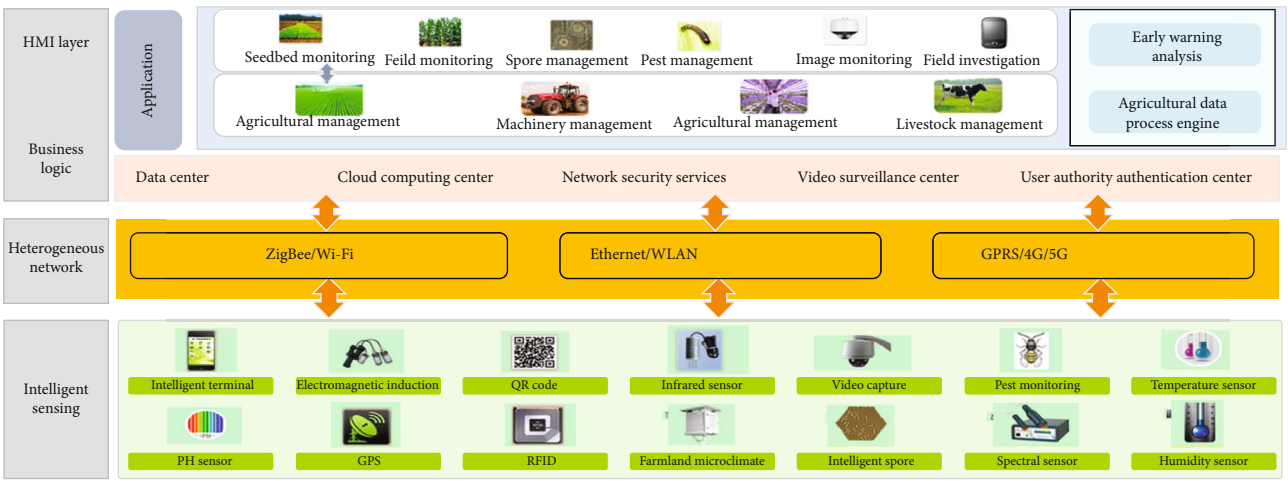


Figure 2: The hierarchical business logic architecture of the proposed system.

applies star topology with low energy consumption, wide coverage, and high bandwidth. The edge computing layer is able to perform online real-time data processing on each measurement parameter of agricultural environment using artificial intelligence and data fusion algorithms at the information collection site. Each functional module is encapsulated through container technology, and information is transmitted between each functional module and between the cloud and the edge through the edge messaging middleware server. Compared with the traditional cloud architecture monitoring system, it can reduce the transmission delay, improve the system response speed, and reduce the pressure on the server side. The application layer implements an agricultural environment monitoring visualization platform to effectively manage and apply the data collected by multiple types of nodes and to make expert decisions and early warnings based on environmental factors. It is designed for the platform resource management and collaboration. Resource sharing and resource control policy can be made among edge cloud systems to achieve controlled sharing of resources and service convergence.

More precisely (as illustrated in Figure 3), the cloud computing centre adapts to access all edge clouds and provides unified services to the external through the conversion of resources. The sensing devices sense the information of the physical world (such as temperature, humidity, and pressure) and transmit the collected data to the corresponding servers through the network. The execution devices execute the received instructions from the upper layer or their control logic to realise the operation of the physical world. The cloud-edge collaboration refers to the access to the corresponding edge cloud system according to the functional requirements and geographic location. The edge system shields the differences of sensing devices to realise the

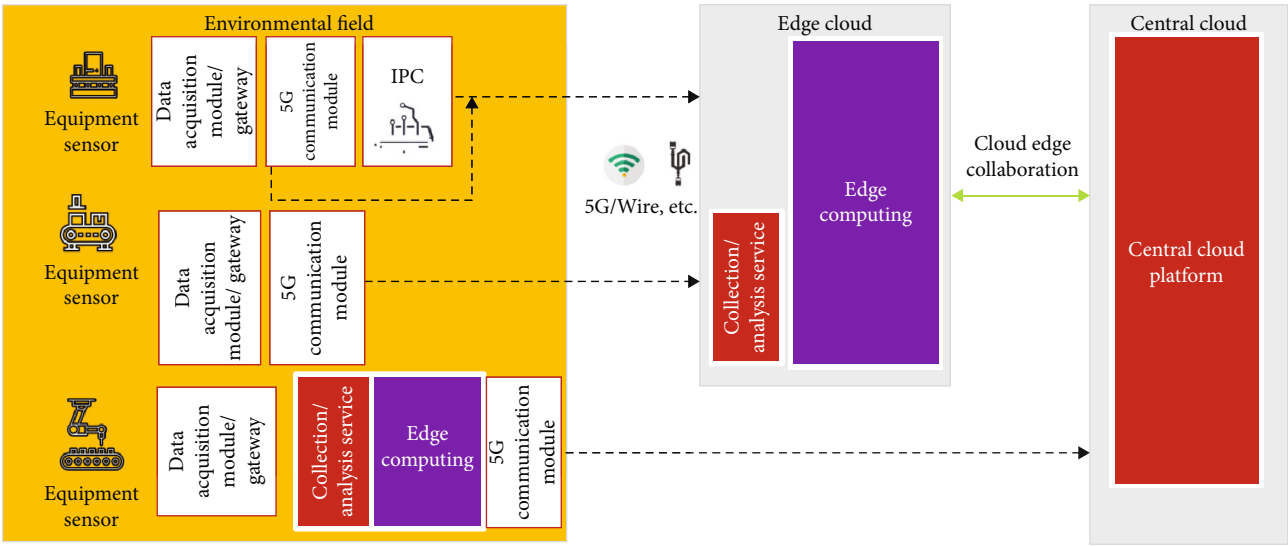


Figure 3: The workflow of cloud-edge collaborative management.

unified management and control of multisource heterogeneous devices. The edge cloud system provides device access and management services to the perception layer and provides a unified resource open interface to the business logic layer. The customised services of the cloud-edge collaboration meet the diversity of the underlying device access, and the edge cloud system deployed at the data source can effectively utilise the network bandwidth to provide high-quality services for the sensing layer devices and IoT applications. The heterogeneous network layer adopts different network communication technologies (such as WLAN, 5G, NB-IoT, and LoRa).

The data stored in the platform are mainly (1) text data collected (such as data collected by temperature and humidity sensors), (2) audio and video data, and (3) system data (such as users, service call permissions, system configuration). It is difficult to store these massive heterogeneous data using a structured database, so we use Redis (https://redis.io) as the persistent data repository. A cloud-based file storage system (https://os.iot.10086.cn) is employed to store IoT audio and video data and uses a content distribution network to distribute the resources according to the regional nature of the data. We employed SQL Server 2015 to store entity data. The structured database supports complex conditional statement queries to meet the platform application requirements. The platform uses read-write separation to operate the database to improve access performance, making data read and write in different instances.

3.2. Cloud-Edge Service and IoT Data Security. The IoT edge cloud collaboration applies to large-scale heterogeneous scenarios, using the cloud computing centre to connect edge cloud systems at the edge of the distributed network and manage and control the edge cloud systems to achieve cross-edge cloud service collaboration. The relationship between the cloud computing centre and the edge cloud systems is shown in Figure 3. The cloud computing centre is a key component of the edge cloud collaborative architecture, which connects each edge cloud system through the cloud computing centre, links the data resources between each edge cloud system, and realises cross-edge cloud collaborative services. The cloud computing centre provides management and control functions for each edge cloud, such as resource access policy, resource identification, instantiation deployment of cloud-edge microservices, security monitoring, and user management. Most of the data is stored in the edge cloud, and some resources frequently requested are stored in the cloud computing centre to reduce the dependency. In addition, the cloud computing centre integrates the platform resources and provides a unified service interface to the developers, making the underlying heterogeneous system transparent to the developers.

The edge cloud system is customised according to the application and device requirements of the scenario where it is located and can be adapted to various network communication protocols and data formats in the underlying layer. It uses a unified external interface in the upper layer to communicate with the business logic layer. The deployment of the edge cloud system requires digital certificates issued by the cloud computing centre as the legal proof of identity and the key for resource sharing. The service interface uses RESTful specification (https://restfulapi.net/), HTTP protocol for synchronous data exchange, and RabbitMQ (https://www.rabbitmq.com/) message queue for asynchronous data exchange, etc. The data generated by the edge cloud system is processed, analysed, and stored locally. The edge cloud system administrator has the highest management control over the data in the edge cloud and can decide which data are open to the public and which data can only be used in the current edge cloud system. The edge cloud system is an integral part of the IoT edge cloud collaborative architecture, which is implemented according to the service requirements of the scenario, mainly consisting of device access middleware, data storage centre, data analysis and

the processing module, and service provision centre and resource sharing and exchange module, and the edge cloud architecture is shown in Figure 3.

The heterogeneity of edge clouds for IoT resources leads to the difficulty of resource sharing and service collaboration among edge clouds. The resource management module was aimed at managing and utilising heterogeneous resources efficiently. IoT identification is the basis of IoT resource management; due to the lack of unified identification system standards, it is challenging to share resources among systems. We propose an IoT identity mapping method to support various identity technology standards, adopt a customised identity method within the system, and decouple resource identification and resource location. It adopts Uniform Resource Name (URN) to identify platform resources and uses Uniform Resource Locators (URL) to search resources. It realises (1) the compatibility of various IoT identification standards through the resource identification mapping, (2) the unique identification of heterogeneous resources in each edge cloud, and (3) the unification of the platform resource description mode. It reduces the difficulty of resource expression format conversion in the service collaboration operation in each edge cloud and supports the sharing of resources in each edge cloud. The architecture of the identity mapping module is shown in Figure 4.

The identity mapping service is deployed in the cloud computing centre to provide an identity mapping information query service. The identity mapping information includes the platform virtual identification number, the resource's physical identification number, and the edge cloud system number where the resource is located. Through this module, the platform virtual identification number can be converted to the physical identification number of the resource and the edge cloud system number where the resource is located. When retrieving the platform resources, only the platform virtual identification number of the resource needs to be passed in to discover the edge cloud system where the resource is located. Its physical identification number achieves the resource search. The identity mapping management module is deployed in the cloud computing centre to provide services for creating, modifying, and deleting identity mapping information. An edge cloud resource, which could be shared externally, must be registered in this module to obtain the platform virtual identification number before it can be accessed externally. When the physical identification number of the resource changes, only the physical identification number of the resource in the identity mapping information needs to be modified, while the platform virtual identification number of the resource does not need to be changed, which avoids the modification of applications developed based on the resource and simplifies the work of platform application developers. The identity mapping cache is deployed in each edge cloud system to cache the resource mapping information to reduce redundant data requests.

3.3. Agricultural-Oriented Service Management. The platform provides the information collection function, with which we can view the real-time status of the plot on the map. The platform transmits the images collected by high-definition cameras to the data centre through the network and enables real-time preview and playback. The interface counter can monitor the total number of interface calls and the success rate of interface calls. The system manager can browse and manage the information of base stations in each block (including base station name, base station level, base station serial number, and base station map markers) and sensor information (such as sensor name, category, health value, display type, and setting the period of sensor upload data). The system can control the equipment on the farm, such as turning on/off the devices, including the fan, the fill light, the shade screen, and the automatic sprinkler irrigation. The system can also support the facilities' performance monitoring, such as the management of M2M cards and SIM card management.

Some of the functional interfaces of the system are shown in Figure 5. Figure 5(a) shows the environmental factor functional interface. In this interface, the left area shows the sensor's name and the current value obtained from the sensor. Below the sensor value is the selection of parameters such as soil moisture, air humidity, and light level. The middle of the interface is shown more visually by displaying the locations where the platform has been used on a map, marked by red dots.

On the right side, information such as device name, device information, status, and switches is displayed, giving an intuitive and convenient display. The video monitoring function is shown in Figure 5(b). The left side of the interface shows the completed campus monitoring. You can select the location or camera you need to view. The area's video monitoring will be displayed in the middle of the page after double-clicking. You can also adjust the direction of the surveillance camera in multiple paths through the operation area below to see the surrounding images without leaving any dead angle. You can also adjust the number of images displayed in the main interface by the number of window segments, up to 16 cameras simultaneously. Click the history video button in the upper left corner to query the history video; after selecting the camera location, enter the query period in the operation area below to query the history video; the function also has the parts of pause, resume, fast playback, slow playback, etc. Growth report real-time query as shown in Figure 5(c) is to query the real-time growth monitoring data returned by each sensor; the menu is divided into three levels, menu level 1 for the agricultural industry, menu level 2 for the company, and menu level 3 for the sensor. Intelligent control engine function as shown in Figure 5(d), the left menu of the interface to select the regional node, after selecting the middle of the interface shows the existing equipment, you can also enter the query conditions above to filter. The primary displayed device information is device name, device information, status, node name, and switch. The switch button can be used to adjust the operation and stop of the device. The system can set the threshold value to achieve automatic control. Suppose the air temperature and humidity are greater than 30 degrees. The fan will be automatically turned on to cool down and automatically turned off when the temperature is less than 25 degrees.

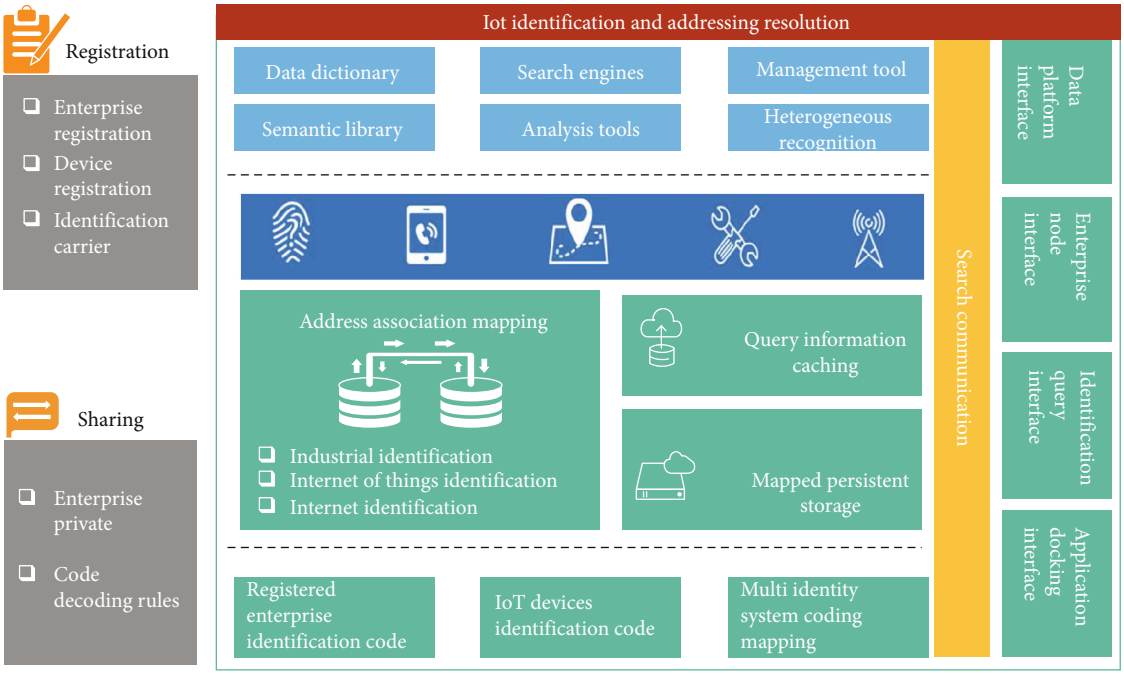


Figure 4: The architecture of IoT identity mapping service.

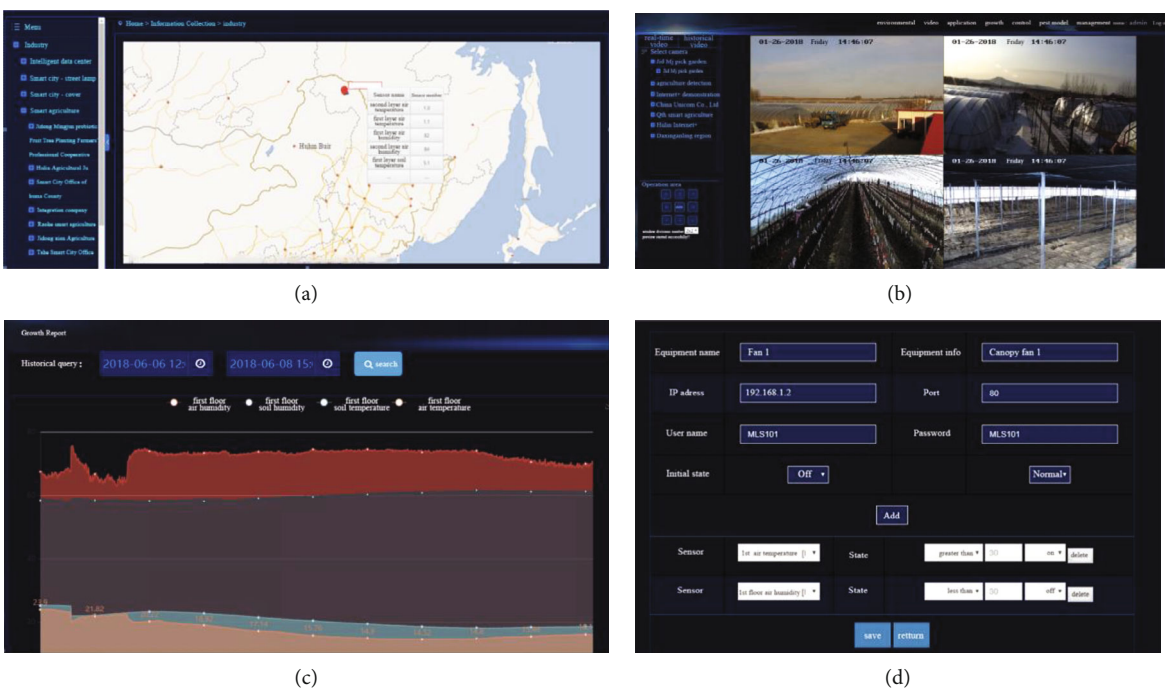


Figure 5: The snapshots of the system functions: (a) Environmental factor function. (b) Video monitoring function. (c) Growth report query. (d) Intelligent control engine function.

3.4. Environmental Predictive Computing Service. RNNs are mainly used to process temporal data, such as speech and text. In temporal data, the output of the current time point is related to the input of the previous time point, and traditional neural networks cannot capture this back-and-forth dependency, while RNNs learn this relationship by adding periodic connections to the neurons in the hidden layer. Figure 6(a) shows the basic structure of an RNN, where x and y denote the input and output vectors, respectively, H denotes the hidden layer, W_1, W_2, and W_3 are the weight matrices, respectively, and h denotes the output vector of the hidden layer. Figure 6(a) shows the circular structure of RNN, and Figure 6(b) shows the expanded structure of Figure 6(a). We find that the output y_t at the current time

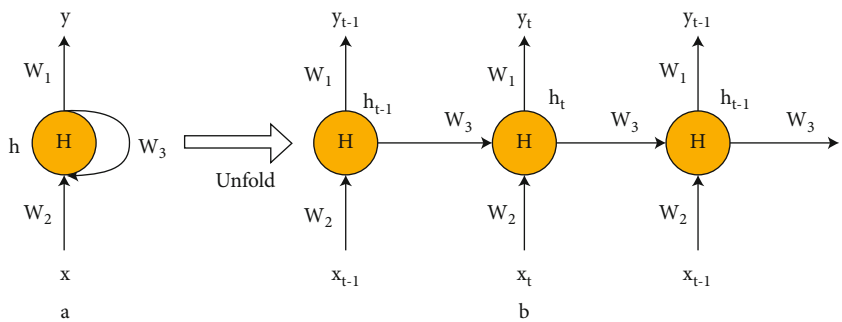


Figure 6: The structure of RNN.

point is jointly determined by the output h_{t-1} of the hidden layer at the previous time point and the current input x_t, and h_{t-1} contains the output information of the hidden layer at all previous time points. The RNN at time point t can be represented by the following equation.

$$h_t = \tanh\left(W_3 h_{t-1} + W_2 x_t\right), \quad y_t = W_1 h_t. \tag{1}$$

The expanded RNN structure can be regarded as a feed-forward neural network with N intermediate layers, so it can be trained using the backpropagation algorithm. However, in the process of backpropagation, the continuous multiplication of W_3 and h_{t-1} tends to cause gradient disappearance and gradient explosion, which makes it difficult for the RNN to learn the forward and backward information dependence at a long distance.

To solve this problem, long short-term memory networks (LSTM) are proposed. Similar to RNN, LSTM also consists of a set of repeated neural network modules, and such modules are called memory blocks. As shown in Figure 7, each memory block contains three gates, i.e., forgetting gate, input gate, and output gate. In contrast to RNN, which has only one state for recurrent transmission (output of the hidden layer), LSTM has two, namely, the hidden layer state (h_t) and the cellular state (s_t), that runs through the entire memory block. In Figure 2, s_{t-1} and h_{t-1} are the cell state and the hidden layer state at the previous time point, respectively, and x_t and y_t are the input and output at the current time point, respectively. The roles of the three gates are described in detail below.

The forgetting gate determines how much information is discarded from the cell state. The hidden state h_{t-1} at the previous time point and the input x_t at the current time point are fed into the memory block and after the activation function Sigmoid outputs a proportional value from 0 to 1, which represents the proportion of information retained from the cell state. Finally, the proportional value is multiplied by s_{t-1} to achieve the forgetting function. The forgetting gate can be represented as

$$f_t = \text{Sigmoid}\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \tag{2}$$

where W_f and b_f denote the weight and bias, respectively.

The input gate determines how much information from the current input is added to the cell state. First, consistent with the forgetting gate, the hidden state h_{t-1} from the previous time point and the input x_t from the current time point are input to the memory block, and after the activation function Sigmoid outputs a scale value from 0 to 1, which represents the proportion of information retained from the current input. At the same time, $h_{t-1}$and x_t are input to the memory block, and $x_t^{'}$ is output after the activation function tanh. Finally, the proportional value is multiplied with $x_t^{'}$ to realise the input function. The input gate can be represented as

$$i_t = \text{Sigmoid}\left(W_i \cdot [h_{t-1}, x_t] + b_i\right), \quad x_t^{'} = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right), \tag{3}$$

where W_i and W_C denote weights and b_i and b_C denote biases.

The output gate determines the output information for the current point in time. This is also done first by feeding the memory block with the hidden state h_{t-1} of the previous time point and the input x_t of the current time point, which is then passed through the activation function Sigmoid to obtain a proportion. Then, the output value of the cell state after the activation function tanh is multiplied by the proportion to get the output at the current time point. The output gate can be expressed as follows.

$$o_t = \text{Sigmoid}\left(W_o \cdot [h_{t-1}, x_t] + b_o\right), \quad h_t = o_t * \tanh\left(s_t\right), \tag{4}$$

where W_o and b_o denote the weight and bias, respectively.

4. Experiment

The experiment was designed to verify the overall availability of the proposed system and the accuracy of its environmental prediction function.

The packet loss, throughput, and response time are frequently used metrics to describe the system availability. We used a group of data to test the packet loss (between transmission layers) of the system data transmission and in the environmental information data collection. The terminal

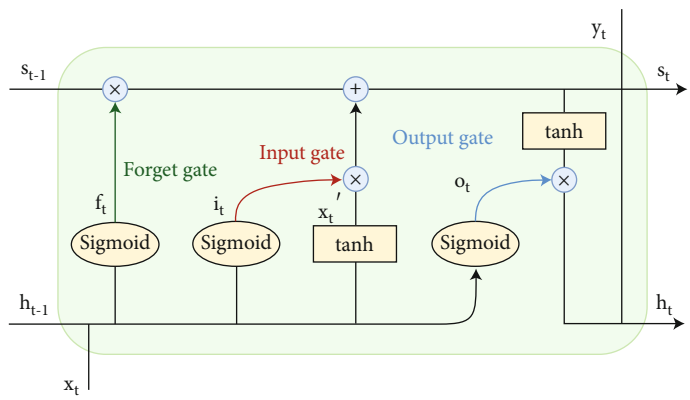


Figure 7: The structure of LSTM.

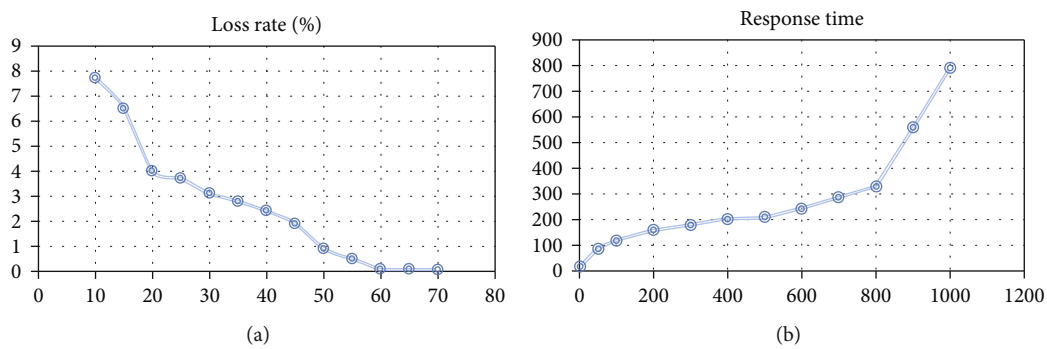


Figure 8: The experimental result regarding the performance metrics.

Table 1: Availability testing result.

Transactions (hits)	Availability	Elapsed time (secs)	Response time (secs)	Transaction rate (trans/sec)	Concurrency	Successful transactions	Failed transactions
10000	100%	6.7	0.05	1492.5	200	10000	0
10000	100%	8.1	0.89	1234.5	300	10000	0
10000	100%	12.1	1.93	826.4	400	10000	0
10000	100%	15.4	2.01	649.3	500	10000	0
10000	100%	24.55	2.54	407.3	600	10000	0
10000	99%	47.06	2.87	210.0	700	9989	11
10000	96%	78.1	3.13	128.0	800	9602	398

Table 2: Samples from the selected dataset.

Date	Maximum temperature (°C)	Minimum temperature (°C)	Average temperature (°C)	Average humidity (%RH)	Yield (kg/mu)
2015-01-01	1.9	-0.4	0.7875	75	907.177044
2015-01-02	6.2	-3.9	1.7625	77.25	747.835779
2015-01-03	7.8	2	4.2375	72.75	740.097015
2015-01-03	8.5	-1.2	3.0375	65.875	760.081199

device collects data in a given period (e.g., an hour) and sends it to the automatic monitoring base station for aggregation at regular intervals (e.g., every 2 hours). The environmental monitoring wireless node collects data once a day and sends them to the monitoring base station for aggregation, and finally, all data are transmitted to the IoT platform. As shown in Figure 8(a), under different sampling frequencies, the computing unit performs 5000 data sends and

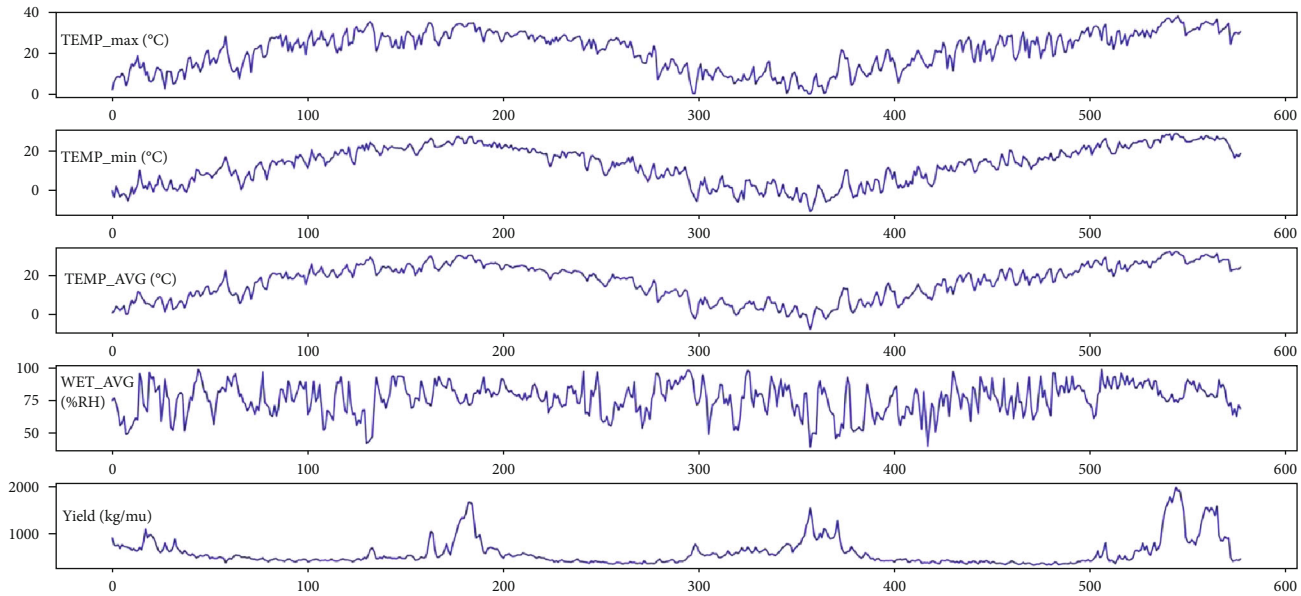


Figure 9: The visualisation of the selected experimental dataset.

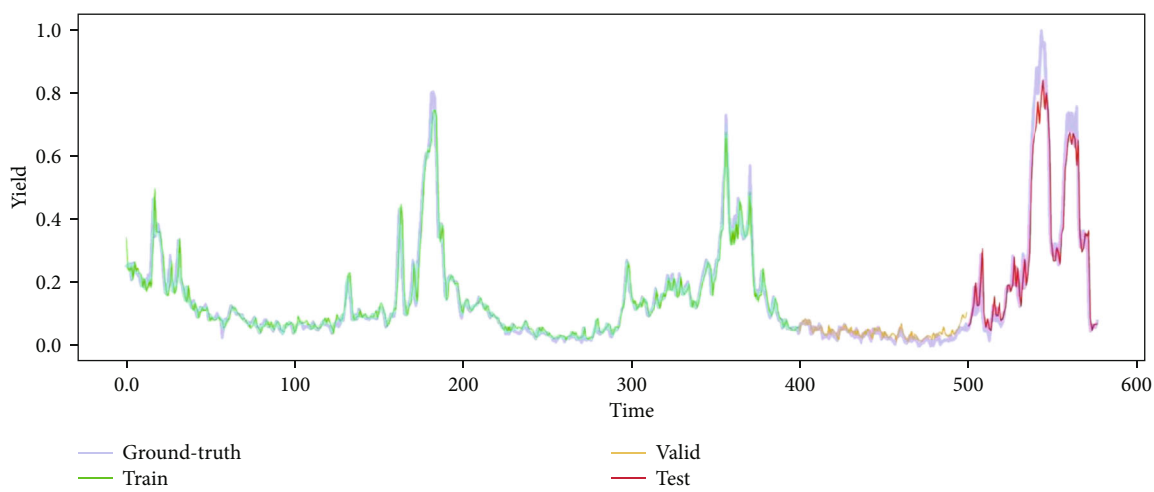


Figure 10: The visualized result of training data, valid data, and test data.

counts the number of packet losses, with the increase of sampling frequency, the overall packet loss rate declines sharply, and when the sampling frequency reaches 60 ms, the packet loss rate drops to 0.1%. The result falls in our expectation, and it indicates the proposed system could fulfil the requirement in practice.

The system is designed for large-scale heterogeneous scenarios, and the platform needs to guarantee service availability during high concurrent scenarios. The throughput and response time of the platform are verified by analysing the log data of the platform under different concurrency. We use the Pulsar tool (https://pulsar.apache.org/) to simulate the service requests to the platform, with different values of concurrent clients. The test result of workload is shown in Table 1, in which, the task queue indicates the percentage of backlogged tasks in the task queue. For each group in the experiment, 10000 transactions were executed with different size of terminals (i.e., concurrency). In the 6th group, 11 transactions failed; that means the concurrency maximum is between 600 and 700. The relationship between the average response time and the number of concurrent clients is shown in Figure 8(b). The average response time increases with the number of concurrent threads. When the number of concurrent threads surpasses the maximum number of threads supported by the edge cloud, there is a significant increase in the average response time. As shown in Figure 8(b), the system throughput of the edge cloud system deployed with a single service reaches the maximum when the concurrency is 300, indicating that the system does not saturate with the number of tasks when the concurrency is less than 300; while the number of tasks saturates when the concurrency is more significant than

Table 3: Part of the result of the crop yield prediction.

Date	Maximum temperature (°C)	Minimum temperature (°C)	Average temperature (°C)	Average humidity (%RH)	Yield (kg/mu)	
					Actual	Predicted
2016-06-15	27.6	22.2	24.4875	72.5	408.97309	434.70498
2016-06-16	30.2	17.8	24.45	69.125	419.502002	427.01839
2016-06-17	32.8	20.1	26.875	61.125	463.757745	492.51727
2016-06-18	33.2	21.7	28.175	64.25	507.862592	553.84575
2016-06-19	32.8	23.9	27.6375	79.625	637.323462	684.95569
2016-06-20	31	21.5	26.425	81.875	492.604463	544.86529
2016-06-21	25.9	24.2	25.275	99	507.48092	541.65983
2016-06-22	32.6	25.1	28.6875	87.875	637.718308	711.57310
2016-06-23	34	26.3	29.625	84.5	809.730649	871.13022
2016-06-24	24.6	22.5	23.45	93.625	488.359556	499.60946

300, but the system throughput does not decrease significantly when it is less than 800. When the concurrency reaches 300, the task queue starts to have tasks piling up, but at this time, the system can still process in time, and the task processing error rate is 0. When the concurrency reaches 800, request processing exceptions start to occur.

We designed an experiment based on the computing service (detailed in Section 3.5) to predict crop yield based on meteorological, environmental factors. The data were collected from February 1, 2015, to August 31, 2016. They contained four environmental factors, i.e., the maximum temperature of the day, minimum temperature of the day, the average temperature of the day and average humidity of the day, and crop yield. The dataset contains 46 data from different data sources, and each data contains 578 datasets. Some of the data in the dataset are shown in Table 2.

The visualization of the dataset is shown in Figure 9. The five images from top to bottom in Figure 9 show the data changes of maximum temperature, minimum temperature, average temperature, average humidity, and yield of the day in chronological order. The horizontal axis represents the time, totalling 578 days, and the vertical axis represents temperature, humidity, and yield. As shown from Figure 9, the temperature varied with the change of the season, the humidity factor had no obvious pattern, and the work reached its highest in the harvest period (in late July).

We first normalized the temperature, humidity, and yield, dividing the dataset into a training set, validation set, and test set, containing 400, 100, and 78 sets of data, respectively, and trained 100 rounds with the mean square error as the loss function. At the end of the training, the training loss reached 0.0016, the validation loss reached 0.0004, and the test loss reached 0.0176. Figure 10 shows the performance of the model. The horizontal axis represents time and the vertical axis represents the yield (after normalisation). The blue curve indicates the actual yield value, the green curve indicates the predicted yield value output from the training set, the orange curve indicates the predicted yield value output from the validation set, and the red curve indicates the predicted yield value output from the test set. As shown in Table 3, the prediction results of LSTM on the training, validation, and test sets are consistent with the true values, but the predicted values are slightly higher than the actual values in the nonharvest period and slightly lower than the actual values in the harvest period.

5. Conclusion

In this paper, an agricultural environment monitoring system is built by integrating edge computing and artificial intelligence. This paper investigates the traditional architecture of agricultural IoT system, proposes a cloud-edge collaboration framework for agricultural environment monitoring, and implements agricultural environment prediction function based on LSTM. The system has been deployed in more than 10 large-scale farms. There are still some shortcomings in the research; e.g., for sensors, improper installation location may lead to inaccurate data acquisition, and instability could result in data collection changes. Moreover, there are some wireless sensors transmission signal distance is limited. The power supply for equipment is not easy-to-obtain: solar power supply may not provide sufficient power, and the adoption of AC power required relocating power wires on the site. Advancement of sensor technology is expected in future. In other hand, for LSTM structure, the training cost of its model is relatively high. In the future, we will introduce several improved versions of LSTM (e.g., coupled LSTM) and other methods on the agricultural environment prediction model proposed in this paper to enhance the training performance of the prediction model and improve the prediction accuracy.

Data Availability

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] G. Premsankar, M. D. Francesco, and T. Taleb, "Edge computing for the Internet of Things: a case study," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1275–1284, 2018.

[2] S. Gill, I. Chana, and R. Buyya, "IoT based agriculture as a cloud and big data service: the beginning of digital India," *Journal of Organizational and End User Computing*, vol. 29, no. 4, pp. 1–23, 2017.

[3] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019.

[4] X. Zhou, X. Xu, W. Liang et al., "Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1377–1386, 2022.

[5] S. Li, L. Xu, and S. Zhao, "5G Internet of Things: a survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1–9, 2018.

[6] R. K. Singh, M. Aernouts, M. De Meyer, M. Weyn, and R. Berkvens, "Leveraging LoRaWAN technology for precision agriculture in greenhouses," *Sensors*, vol. 20, no. 7, 2020.

[7] W. Qiu, K. Saleem, M. Pham et al., "Robust multipath links for wireless sensor networks in irrigation applications," in *The 3rd Intelligent Sensors, Sensor Networks and Information Processing Conference*, Melbourne, VIC, Australia, 2007.

[8] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks: a survey towards private and secure applications," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.

[9] K. Ashton, "That 'Internet of Things' thing," *RFID Journal*, vol. 22, no. 7, pp. 97–114, 2009.

[10] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[11] X. Zhou, X. Yang, J. Ma, and K. Wang, "Energy efficient smart routing based on link correlation mining for wireless edge computing in IoT," *IEEE Internet of Things Journal*, vol. 8, 2021.

[12] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[13] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *The proceedings of the first edition of the MCC Workshop on Mobile Cloud Computing*, pp. 13–16, Helsinki Finland, 2012.

[14] L. Ryan, "Edge Computing [EB/OL]," 2013, https://mafiadoc.com/edge-computingpacificnorthwest-national-laboratory_59d648481723dd08e35b7b77.html.

[15] W. Liang, Y. Hu, X. Zhou, Y. Pan, and K. Wang, "Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5087–5095, 2021.

[16] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, "Deep-learning-enhanced multitarget detection for end-edge-cloud surveillance in smart IoT," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12588–12596, 2021.

[17] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[18] Y. Guo, P. Corke, G. Poulton, T. Wark, G. Bishop-Hurley, and D. Swain, "Animal behaviour understanding using wireless sensor networks," in *2006 31st IEEE Conference on Local Computer Networks*, Tampa, FL, USA, 2006.

[19] N. Vijayakumar and R. Ramya, "The real-time monitoring of water quality in IoT environment," *International Journal of Science and Research (IJSR)*, vol. 4, no. 3, 2015.

[20] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[21] J. Chen, T. Cai, W. He et al., "A blockchain-driven supply chain finance application for auto retail industry," *Entropy*, vol. 22, no. 1, p. 95, 2020.

[22] S. Sendra, L. García, J. Lloret, I. Bosch, and R. Vega-Rodríguez, "LoRaWAN network for fire monitoring in rural environments," *Electronics*, vol. 9, no. 3, p. 531, 2020.

[23] A. Valente, S. Silva, D. Duarte, F. Cabral Pinto, and S. Soares, "Low-cost LoRaWAN node for agro-intelligence IoT," *Electronics*, vol. 9, no. 6, p. 987, 2020.

[24] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[25] S. K. Grange and D. C. Carslaw, "Using meteorological normalisation to detect interventions in air quality time series," *Science of the Total Environment*, vol. 653, pp. 578–588, 2019.

[26] N. Wang, N. Zhang, and M. Wang, "Wireless sensors in agriculture and food industry–recent development and future perspective," *Computers and Electronics in Agriculture*, vol. 50, no. 1, pp. 1–14, 2006.

[27] L. A. González, G. J. Bishop-Hurley, R. N. Handcock, and C. Crossman, "Behavioral classification of data from collars containing motion sensors in grazing cattle," *Computers and Electronics in Agriculture*, vol. 110, pp. 91–102, 2015.

[28] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, "Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5790–5798, 2021.

[29] D. Zhu, G. Shen, D. Liu, J. Chen, and Y. Zhang, "FCG-ASpredictor: an approach for the prediction of average speed of road segments with floating car GPS data," *Sensors*, vol. 19, no. 22, p. 4967, 2019.

[30] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing multidimensional time-series," *The VLDB Journal*, vol. 15, no. 1, pp. 1–20, 2006.

[31] M. Iqbal and A. Naveed, "Forecasting inflation: autoregressive integrated moving average model," *European Scientific Journal*, vol. 12, no. 1, pp. 83–92, 2016.

[32] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. Wang, "Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9310–9319, 2021.

[33] B. Sadeghi, "A bp-neural network predictor model for plastic injection molding process," *Journal of Materials Processing Technology*, vol. 103, no. 3, pp. 411–416, 2000.

[34] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, 1998.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6476274, 13 pages
https://doi.org/10.1155/2022/6476274

Review Article

A Survey on Zero Trust Architecture: Challenges and Future Trends

Yuanhang He,[1] Daochao Huang,[2] Lei Chen,[1] Yi Ni,[1] and Xiangjie Ma[1]

[1]*No.30 Research Institute of China Electronics Technology Group Corporation, Chengdu, China*
[2]*National Computer Network Emergency Response Technical Team/Coordination Center of China (CNCERT/CC), Beijing, China 100029*

Correspondence should be addressed to Lei Chen; chenl_ccsc@163.com

Received 29 March 2022; Accepted 9 May 2022; Published 15 June 2022

Academic Editor: Yan Huo

The traditional perimeter-based network protection model cannot adapt to the development of current technology. Zero trust is a new type of network security model, which is based on the concept of never trust and always verify. Whether the access subject is in the internal network or the external network, it needs to be authenticated to access resources. The zero trust model has received extensive attention in research and practice because it can meet the new network security requirements. However, the application of zero trust is still in its infancy, and enterprises, organizations, and individuals are not fully aware of the advantages and disadvantages of zero trust, which greatly hinders the application of zero trust. This paper introduces the existing zero trust architecture and analyzes the core technologies including identity authentication, access control, and trust assessment, which are mainly relied on in the zero trust architecture. The main solutions under each technology are compared and analyzed to summarize the advantages and disadvantages, as well as the current challenges and future research trends. Our goal is to provide support for the research and application of future zero trust architectures.

1. Introduction

Since human beings entered the information age, the problem of information security has been perplexing with the further development and practical application of information technology. In 2010, the Stuxnet virus attacked the supervisory control and data acquisition (SCADA) equipment designed to destroy Iran's nuclear fuel enrichment process. The attack successfully destroyed SCADA equipment in many parts of Iran. The "Stuxnet" virus has achieved the destruction of a single piece of information and data to the actual physical facilities, which marks the entry into a new stage of cyber warfare. In 2015, a cyberattack on Ukraine's power grid caused a massive blackout in western Ukraine, leaving a fifth of the Ukrainian capital without power. This is the first-ever cyberattack posing a danger to a nation's critical infrastructure. Intuitively, network security issues have threatened industrial control and infrastructure.

To solve this problem, the perimeter-based security architecture is proposed, which divides the network into internal network and external network with a firewall, intrusion detection system (IDS), or intrusion prevention system (IPS) as the border. According to the physical location of the object, it is judged whether it is located in the internal network, and the object in the internal network is regarded as trusted by default. External objects, on the other hand, must be authenticated before they can be trusted. In a perimeter-based security architecture, once an object is authenticated, it is trusted for a long time. Therefore, if a malicious object is authenticated, it can continue to attack and sabotage the internal network. At the same time, with the continuous development of cloud computing and Internet of Things technologies and the popularization of telecommuting, especially since the outbreak of the new crown, telecommuting has become an indispensable way of working. Therefore, based on the

physical location of the object, it is no longer possible to judge whether it is located in the internal network, let alone give it corresponding trust.

To address this new challenge and problem, the National Institute of Standards and Technology (NIST) proposed the concept of zero trust architecture (ZTA) [1]. It is different from the perimeter-based security architectures; the trust of an object is independent of its physical location and all objects are untrusted by default. The trust of an object can only be obtained by identity authentication and trust evaluation. After the system assigns the relative permissions to the object, the object can perform related operations. In recent years, zero trust architecture has been initially applied, and the most typical example is Google's BeyondCorp model [2]. In this model, first, users need to perform location-based identity authentication. For example, in the public network, single-point SSO is used for authentication. Authorization is also required after authentication, and the access authority can be obtained only after the authorization is successful, and the authority is obtained by granting the authorization information to the access agent through the access control engine. BeyondCorp will associate the results (people and devices) to the vnet network segment constructed based on specific services after identity authentication, forming isolation domains with different security levels. If users want to access cross-domain, they must abide by relevant security policies. In addition, zero trust also has some preliminary applications in civil aviation airport network security and virtual power grids [3].

Identity authentication, access control, and trust evaluation algorithms are the technical cornerstones of ZTA. Among them, the identity authentication mainly realizes the identification of the object in the ZTA, the access control mainly realizes the safe and efficient access of the ZTA object to the resources, and the trust evaluation algorithm realizes the evaluation of the trust degree of the ZTA object and is used as the main credential for identity authentication and access control. At present, the research of ZTA is still in the preliminary stage, and the research on the architecture, identity authentication, access control, and trust evaluation algorithm of ZTA is the key field of it. Therefore, this paper introduces the current research status of ZTA development from these four aspects and discusses the main problems they face and future research directions. Our main contributions include:

(i) This paper makes a detailed analysis and summary of the current status of zero trust research. It focuses on the current research status of ZTA architecture, identity authentication, access control, and trust evaluation algorithms, and makes a specific analysis and summary, so as to grasp the overall situation of zero trust research

(ii) Comparing the current main ZTA, identity authentication, access control, and trust evaluation algorithms, and summarize their advantages and main problems

(iii) According to the main advantages and disadvantages of current zero trust in architecture, identity authentication, access control, and trust evaluation algorithms, the main challenges they face are summarized, and the main research directions of zero trust in the future are proposed

The next arrangement of this paper: We introduce the ZTA in Section II, including its main components and operation methods. In Section III, we review the relevant literature on the research status of zero trust including zero trust control and trust evaluation algorithms. We summarize the current progress of the main schemes, and give their comparisons, as well as future research directions in Section IV. Finally, we make a summary of this paper.

2. Zero Trust Architecture

ZTA was proposed by Kindervag, principal analyst at Forrester in 2010. In a zero trust architecture, all traffic cannot be trusted, and location cannot be used as a basis for security. Instead, security measures need to be taken for all access, minimum authorization policies and strict access control are adopted, and all traffic needs to be visualized and analyzed. These concepts are significantly different from the traditional perimeter-based security architecture, and the security is stronger.

2.1. Zero Trust Basic Assumptions and Principles. ZTA is built on the following five basic assumptions:

(1) The network is in a dangerous environment all the time

(2) There are external or internal threats in the network from beginning to end

(3) The location of the network is not enough to determine the credibility of the network

(4) All devices, users, and network traffic should be authenticated and authorized

(5) Security policies must be dynamic and calculated based on as many data sources as possible

Based on the above assumptions, the zero trust model is believed to adhere to the following four basic principles

(1) Authenticate users: Assess user security based on location, device, and behavior to determine if the user is who they claim to be. Take appropriate measures (such as multifactor authentication) to ensure user authenticity

(2) Authenticate devices: Whether it is corporate devices, BYOD or public hosts, or laptops or mobile devices, implement access control policies based on device identity and security. Only trusted endpoints are allowed to access company resources

(3) Restrict access and permissions: If users and devices are authenticated, implement a role-based access control model for resources, giving them the minimum permissions to complete the work at the time

(4) Adaptive: Various sources (such as users, their devices, all activities related to them) are always producing information continuously. Leverage machine learning to set context-sensitive access policies, automatically adjust and adapt to policies

2.2. Zero Trust Architecture. The core goal of zero trust is to allow users in untrusted network areas to access trusted areas through authentication and policy control, as shown in Figure 1.

In order to reduce the security risk of the access process, a continuous dynamic security access control technology is required, which is not based on the network location of the access subject, but authorizes the access subject based on the security and trust status before each access object is allowed to access; continuously monitors the security of the entire access process and assesses the trust status; dynamically adjusts access rights and implements fine-grained security access control.

To achieve this goal, more network elements are needed to support the entire zero trust architecture. The ZTA architecture given by NIST is shown in Figure 2.

Among them, the identity management (ID management) system and the enterprise public key infrastructure (PKI) are mainly used for the authentication of personnel and equipment, which is the basis. The data access policy mainly provides resource access policy, and the security information and event management system (SIEM system) provides the security information and event management of the entire architecture. At the same time, to integrate capabilities such as industrial compliance policies and threat detection, more attention needs to be paid to continuous diagnostics and mitigation (CDM) systems.

In general, the ZTA is based on identity, giving digital identities to people and devices, and setting minimum permissions for access subjects; aiming at business security, realizing business concealment, transmission encryption, and fine control; with continuous trust assessment as the guarantee, including user trust assessment, environmental risk determination, and abnormal behavior discovery; using dynamic permission control as a means, including attribute-based access control baseline, trust level-based hierarchical access, and risk-aware dynamic permissions.

The zero trust architecture focuses on the security capabilities of identity, trust, access control, permissions, and other dimensions, and these security capabilities are also an indispensable part of the information-based business system, so zero trust is inherently a kind of "endogenous security." In a sense, it is a spiral sublimation of business and security. From the initial business system to complete business goals, security equipment realizes the mutual independent system of security assurance, and integrates into a close relationship between security and business, and returns to security and application again.

Figure 1: Zero trust access.

3. Literature Review

In this section, we review and analyze the current status of important technical research on zero trust security. It mainly includes ZTA, identity authentication, access control, and trust evaluation algorithms.

3.1. Review on Zero Trust Architecture. As early as 2010, Kindervag [4] proposed the concept of a zero trust architecture model and a method to implement it in a practical environment and innovatively proposed a zero trust architecture based on "Data Acquisition Network" (DAN) in the paper. DAN helps to extract network data to the management center and then realizes inspection and analysis of it in real time, thus realizing the concept of zero trust, but this is also accompanied by problems of higher network complexity and increased user communication delay.

After that, in 2016, DeCusatis et al. [5] proposed a zero trust method based on transport access control. This method is based on steganography and overwriting, and the authentication token is embedded in the TCP request packet and the first authentication packet. Thus realizing the concept of zero trust, this approach increases the security of enterprises in cloud computing environments and prevents unwanted fingerprinting of protected resources; this approach provides protection at layer 3/4 but not at layer 7.

Subsequently, in 2020, Rose et al. [6] summarized the existing basic zero trust architecture schemes and proposed the basic logical components of the zero trust architecture. In addition, the author paid more attention to the implementation of the zero trust architecture, considering the realization of ZTA. Rather than a massive replacement of infrastructure or processes, it is a process that proposes specific steps to apply ZTA to a perimeter-based architecture network.

Sultana [7] et al. proposed a secure medical image sharing system based on the principle of zero trust and blockchain technology. The system combines zero trust with blockchain. The blockchain is used to protect sensitive information. Comprehensive protection of medical data, but this also increases the complexity of the system and needs to be studied in terms of efficiency.

Weever et al. [8] proposed a zero trust network security model in a containerized environment, which solved how to implement zero trust for "east-west" traffic between microservices in a containerized environment, using Kubernetes and Istio service mesh to build. A zero trust model in containerized environments reduces data leakage in containerized environments, but this model does not implement behavioral analysis and data leakage detection.

In 2021, Ramezanpour and Jagannath [9] proposed an artificial intelligence-based zero trust architecture (i-ZTA), which uses artificial intelligence for intelligent detection,

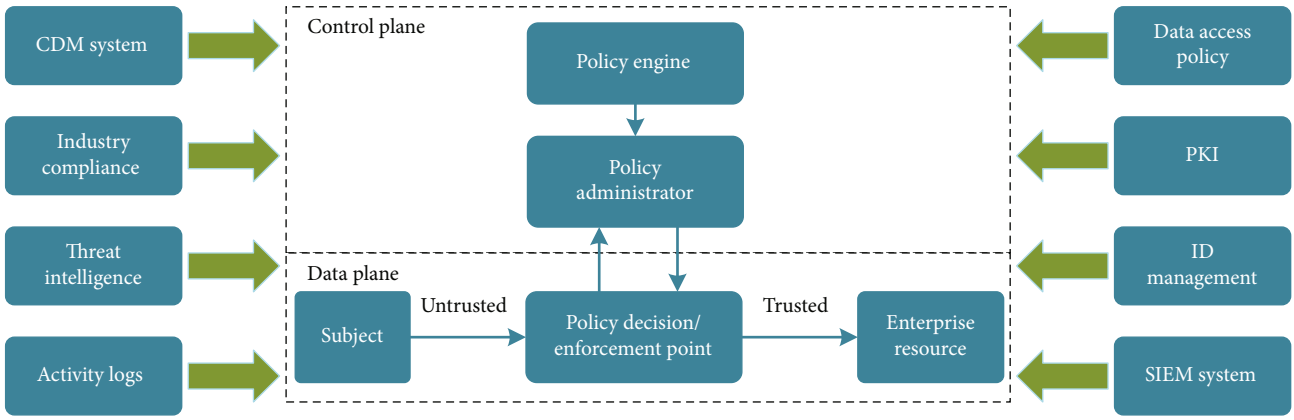


Figure 2: Typical zero trust architecture.

evaluation, and decision-making, which can improve the efficiency of ZTA components in processing big data. The architecture combines artificial intelligence with PEP and PDP in the traditional zero trust architecture. The former uses reinforcement learning with the goal of maximizing guaranteed scores, while the latter uses joint learning to provide users with context-aware scores.

Tian and Song [10] proposed a zero trust approach based on BLP and BIBA models, which conducts comprehensive trust scores for system components such as users, terminals, channels, files, and applications, and requires confidentiality and integrity, setting different weights to achieve better confidentiality and integrity protection of objects, but does not consider the initial trust value granting method of users, terminals, environments, objects, and other entities, and cannot effectively avoid human-factor errors in initial trust. The assignment, as well as the completeness and rationality of the list weight assignment, is for further study.

Ghate et al. [11] proposed an advanced zero trust architecture that leverages generalized attribute relation extraction to automate fine-grained access control to achieve low-cost fine-grained access control, performance, scalability, and security for real enterprise networks. The evaluation is for future work.

A comparison of these ZTAs is shown in Table 1.

3.2. Review on Identity Authentication. Authentication infrastructure is a key supporting component for realizing zero trust architecture's identity-based capabilities. As an access control system based on identity rather than network location, a zero trust system first needs to give digital identities to people and devices in the network, and combines the identifiable people and devices at runtime to construct access subjects. Access subjects need to authenticate themselves before being granted access to specific resources. Identity authentication technology is a method or means used in the process of confirming the real identity of the visitor. Under the ZTA, the system continuously monitors the user's behavior during access, and can adjust policies in real time according to the user's behavior. The process of proving identity is primarily based on information obtained from the user, which can be described by a set of characteristics. The characteristics should be unique and permanent. The multifactor authentication technology based on multifeature identification will be widely used because of its stronger security.

According to the different types of features, we divide authentication into two categories: user-to-device authentication and device-to-device authentication.

User-to-device authentication: Methods of authentication based on biometrics have been widely proposed, and these biometrics can be used to prove the identity of users due to their uniqueness and persistence. Many researchers use human physiological attributes for identity authentication. In 2016, Kindervag et al. [4] proposed to use sensors and applications in Android smartphones to collect fingerprint information for identity authentication, but this method is only for some series of mobile phones and lacks comprehensiveness. The security of single-factor authentication is weak. In 2019, Henderson et al. [12] proposed a two-factor authentication that combines a fingerprint sensor and an LED pulse oximeter, aiming to optimize the shortcomings of a single fingerprint scan with an LED pulse oximeter. In 2015, Matsuyama et al. [13] proposed a method for continuous authentication using low-frequency brain signals, which measured changes in oxyhemoglobin in the brain by near-infrared spectroscopy (NIRS). But the study required placing probes on test subjects' foreheads, which is not suitable for real-life use. In 2016, Mahbub et al. [14] proposed to utilize facial attributes captured by smart devices for continuous identity authentication. There are also studies on identity authentication based on user behavior characteristics. In 2018, Ehatisham-ul-Haq et al. [15] used behaviors such as walking, running, sitting, standing, and going upstairs and downstairs to distinguish different users. In 2020, Abuhamad et al. [16] used sensors in smartphones to capture user behavior patterns such as jogging and exercising while holding a smartphone, combined with contextual information such as text messages, voice, and video chat for authentication. And the method does not require any sensitive software or hardware permissions that could violate user privacy.

Device-to-device authentication: Due to the lack of biometrics, there are relatively few features available for authentication. But this approach assumes that device power consumption is linear and that battery capacity is also affected by the environment. Due to the heterogeneity of

Table 1: Comparison of zero trust architectures.

Literature	Implementation	Advantage	Disadvantage
Kindervag [4]	Extract network data to the management center using DAN	Inspect and analyze data in real time	Network complexity increases and user communication delay increases
DeCusatis et al. [5]	Using steganography and overwriting methods, the authentication token is embedded in the TCP request packet and the authentication header	Increased security for businesses in cloud computing environments and prevents unwanted fingerprinting of protected resources	No protection at layer 7, not comprehensive enough
Rose et al. [6]	Automatically link new APIs to existing service mesh categories by using machine learning-based smart association models	Simplify the creation, management, and monitoring of APIs	Difficult to achieve in real environment
Sultana et al. [7]	The system combines zero trust with blockchain, the blockchain is used to protect sensitive information, and zero trust realizes comprehensive protection of medical data	Combining blockchain with zero trust	Low efficiency
Weever et al. [8]	A zero trust network security model in a containerized environment	Reduced data leakage in containerized environments	Behavioral analysis and data leak detection are not implemented
Ramezanpour and Jagannath [9]	Using artificial intelligence for intelligent detection, assessment, and decision-making	Improve the efficiency of ZTA components in processing big data	Only at the theoretical level
Tian et al. [10]	Zero trust approach based on BLP and BIBA models	Set different weights based on confidentiality and integrity requirements	The weight distribution is not reasonable enough
Ghate et al. [11]	Automate fine-grained access control with generalized attribute relation extraction	Low cost	Failed to measure performance in a real environment

IoT devices, not all devices are battery powered. In 2018, Chuang et al. [17] proposed to utilize the remaining battery capacity of the sensor device as a dynamic feature for authentication. In 2019, Wang et al. [18] proposed the use of electromagnetic radiation (EMR) for identity authentication between devices. However, electromagnetic radiation is easily disturbed by the external environment, so the characteristic factor selected by this scheme also has defects. Meng et al. [19] propose a D2D continuous authentication protocol combining blockchain and trust assessment, where the time interval of authentication is dynamic. The trust assessment center evaluates every device and outputs the trust level; the higher the trust level, the longer the interval it conducts to continuous authentication. Therefore, the scheme is dynamic and flexible for each device with different trust levels.

To sum up, the user-to-device authentication can be verified by collecting the user's biometric information by means of sensors in wearable devices such as mobile phones and watches. And using a variety of information for multifactor authentication can achieve higher security. However, due to the heterogeneity between devices, the identity authentication method suitable for device-to-device has few common features for authentication. It is necessary to further explore the unique and persistent features between devices.

After obtaining the identity information, both parties need to conduct a session to transmit information, and compare the obtained information with the information stored in the database to confirm the identity. According to the different transmission protocols, it can be divided into certificate-based authentication, encryption-based authentication, and nonencryption-based authentication.

Certificate-based Authentication: In 2013, Kothmayr et al. [20] proposed a two-way authentication security scheme for IoT based on the Datagram Transport Layer Security (DTLS) protocol, which uses RSA-based asymmetric encryption and X.509 authentication. However, the protocol requires 8 handshakes to establish a session, so the device needs to have higher computational cost and storage space to implement this solution. In 2016, Verma et al. [21] proposed a certificate-based protocol for node authentication in mobile ad hoc networks. The protocol utilizes trust management mechanisms to keep track of certificate operations and authentication operations and uses digital signatures with hash functions to maintain certificate authenticity. The protocol performs well in terms of robustness. In 2020, Kumar and Gandhi [22] proposed an enhanced DTLS based on smart gateways to overcome denial of service attack servers. The protocol uses packet loss rate to evaluate performance and based on data transfer and handshake time to evaluate protocol efficiency.

Encryption-based Authentication: In 2015, Shivraj et al. [23] proposed one-time password (OTP) authentication for IoT infrastructure. The protocol employs Identity-Based Elliptic Curve Cryptography (IBE-ECC) to provide lightweight end-to-end authentication between devices. The scheme is lightweight with smaller key size and smaller infrastructure, but performs poorly in terms of increased

OTP size, increased computational complexity, and time-consuming performance. In 2016, Kumar et al. [24] proposed a lightweight authentication-based session key establishment protocol for smart home. The protocol requires a security service provider, which is a trusted server. The security service provider assigns important parameters, generates tokens, and distributes the tokens to communication devices. The protocol has high computational and storage efficiency and can defend against a variety of common attack behaviors, but feasibility monitoring is carried out through proof-of-concept implementations. In 2021, Syed et al. [25] proposed a lightweight continuous device-to-device authentication (LCDA) protocol that utilizes communication channel properties and a tunable mathematical function to generate dynamically changing session keys for continuous device-to-device authentication. An evaluation of the protocol using the Scyther tool shows that both the mutual authentication and the continuous authentication phases comply with security properties such as integrity, confidentiality, freshness, and resistance to protocol attacks. However, the effectiveness of this protocol on various constrained devices requires further research in the future.

Nonencryption-based Authentication: In 2015, Gope and Hwang [26] proposed a nontraceable authentication protocol in distributed IoT architectures. This scheme only uses hash functions and bitwise XOR operations to construct a lightweight authentication mechanism. The method is light in calculation and consumes less resources of the device. Taking sensor device movement into account, the scheme not only guarantees the legitimacy of sensor nodes but also supports identity anonymity and untraceability. In 2017, Ying and Nayak [27] proposed an anonymous and lightweight authentication based on the smart card (ASC) protocol to solve the authentication problem in in-vehicle ad hoc networks. ASC uses operations such as hash function and XOR to verify the legitimacy of the user (vehicle) and the verification of data messages. Utilizing a trusted authority to send anonymous certificates and keys to the vehicle, vehicle users must first authenticate and obtain a session key before they can interact securely with each other. The protocol has better efficiency in terms of communication/computational overhead, end-to-end delay, and packet loss rate. However, there are serious security problems in offline identity guessing attacks, session linking attacks, and replay attacks. In 2019, Chen et al. [28] aimed at the problem of offline identity guessing attack and time-consuming authentication phase in literature [27]. To improve security and reduce the time required for authentication, they proposed a patch on the protocol of Ying et al. The patched protocol performs better in terms of security and performance than the original protocol. The comparison of the methods used in the above papers is shown in Table 2.

3.3. Review on Access Control. In the early traditional access control model, role-based access control (RBAC) [29] was viewed as a task performed by a user, assigning him/her one or more roles to indirectly associate permissions with the user, It is considered an alternative to mandatory access control (MAC) and discretionary access control (DAC), which can realize centralized management of role membership and access control, but it only describes the characteristics of the subject and lacks the description of the characteristics of the object, the permissions cannot be dynamically changed, and the constraint particles are large, which can be. The scalability is not strong, and it is difficult to apply in a distributed environment. Attribute-based access control (ABAC) [30] is similar to RBAC in that it mainly grants or denies user requests based on arbitrary attributes of users and globally identifiable attributes of objects, but the disadvantage is that all elements need to be described in the form of attributes. Some relationships are not easily described with basic properties. Thomas et al. designed a task-based access control model (TBAC) [31] and proposed the concept of task-oriented. The model is to establish a secure access mechanism in the workflow, making it widely used in workflow systems and distributed computing systems. However, TBAC is not suitable for complex network environments. It does not involve the issue of user rights assignment, but simply introduces a set of trustees to represent the executors of tasks, and does not discuss how to determine trustees in the actual environment.

Attribute-based encryption (ABE) scheme to achieve access control is mainly studied from three aspects: the first is fine-grained access control, the second is the problem of user attribute revocation, and the third is the multiauthorization center scheme. In 2007, Oh and Park [31] proposed a detailed CP-ABE scheme, which embeds the access structure into the ciphertext and the attribute set of the data user into the private key, only when the attribute set satisfies the access structure, to decrypt the ciphertext. However, the security of this scheme is not ideal, and it is easy to be broken. In the same year, the nonmonotonic access policy ABE of Bethencourt et al. [32] allows the data owner to insert the revoked user ID into the ciphertext in the form of "non" when the data is confidential, so as to realize the revocation of the user's access right to the ciphertext, but this scheme is less flexible. In 2009, in the scheme of Ostrovsky et al. [33], the data owner decides and manages the authorized user list of each attribute, and realizes the revocation of user attributes by sending out the user representation from the authorized list of attributes, using the idea of encrypting the broadcast. Direct revocation is introduced into ABE, which ensures that unrevoked users are not affected and do not need to update keys for users regularly. In 2017, Attrapadung and Imai [34] et al. proposed a generalized software-defined storage (SDS) constrained access control method for cloud storage; the method is based on CP-ABE with hidden access policy. SDS constraints are handled through the participation of additional entities and additional human attributes. But it does not completely hide the constraint policy structure from all entities without affecting SDS constraint enforcement. Constraints such as "and," "or," and "threshold" are not considered. In 2018, Nurmamat et al. [35] proposed a fine-grained access control scheme based on location server for the mobile cloud environment, which takes the dynamic location of the mobile user as the user's information and the location range as the access policy. And will satisfy the demands for the dynamic location of

Table 2: Different technical methods used by authentication protocols.

Literature	Methods	Continuous authentication	Multifactor authentication	Strengths	Weakness
Kothmayr et al. [20]	Datagram Transport Layer Security (DTLS) protocol, RSA-based asymmetric encryption, X.509 authentication	No	No	The system architecture follows the IoT model and inherits the security properties of UDP.	The protocol has eight handshakes, which is computationally expensive
Verma et al. [21]	Certificate	Yes	No	The protocol has better performance in terms of throughput, end-to-end delay, and packet loss. Has a small amount of computation and communication overhead	No discussion of resilience to foreign attacks
Kumar and Gandhi [22]	Certificate Advanced Encryption Standard Counter and Cipher Block Chain Message Authentication Code (AESCCM), Elliptic Curve Digital Signature Algorithm (ECDSA)	No	No	Overcome the denial of service attack server vulnerable to DTLS protocol	This protocol is used in medical and health monitoring, but the collected body information is not used for identity authentication, but only as transmitted data information.
Shivraj et al. [23]	Elliptic Curve Cryptographic (ECC)	No	Two-factor	The protocol is scalable, with small keys and robustness	As the size of the OTP increases, the computational complexity also increases, and the time consumption increases significantly
Kumar et al. [24]	Symmetric key, hash function	No	No	The scheme provides important security properties, including protection against a variety of common attacks, such as denial of service attacks and eavesdropping attacks	Preliminary evaluation and feasibility testing was carried out through the implementation of the proof of concept
Syed et al. [25]	Cryptography	Yes	No	The protocol can be adapted to devices with limited computing and storage resources	Difficulties in measuring Channel State Information (CSI) for heterogeneous IoT devices
Gope and Hwang [26]	Hash function, XOR	Yes	No	The protocol provides more security features under the premise of ensuring less computational overhead, with anonymity and nontraceability	Security analysis is just a proof by means of theoretical analysis
Ying and Nayak [27]	Hash function, XOR	No	No	An efficient password modification phase that does not rely on TA (trusted authority) and third-party servers is proposed, which can resist offline password guessing attacks.	There is no reasonable extension of the protocol, and the protocol is insecure against offline identity guessing attacks, session link attacks, and replay attacks
Chen et al. [28]	Hash function, XOR	No	No	Fixed the security vulnerability found in [27]	The protocol only uses the iPhone as a test platform.

the access policy is authorized to users for privacy protection. However, this solution introduces a third-party location-based services (LBS). If the LBS is maliciously damaged, the entire system will crash.

Due to the diversity of user devices and the consistency of access policies, the zero trust access control model requires support for dynamic network access. The first zero trust architecture was proposed by Google. After Google suffered a highly sophisticated APT attack in 2009, it began to redesign the security architecture for employees and devices to access internal applications, so that employees can achieve secure access at any location. No traditional VPN required.

In order to meet the needs of the company's internal mobile office, Google designed and implemented a relatively stable zero trust network model BeyondCorp [36], in which the authorized access rights after identity authentication need to be obtained by granting the access agent authorization information through the "access control engine." It adopts the ZTN method for access control, but does not describe in detail the implementation of policy language, risk management, or decision-making continuity, and does not fully consider the inheritance and reuse of existing networks, just visualize its security capabilities as a product.

In 2016, Ward and Beyer used transmission access control and first packet authentication to realize zero trust cloud network. Based on the principle of zero trust network, they redesigned the network architecture of data center and demonstrated the principle of zero trust through transmission access control system. One of the most important principle is steganographic overlay, which embeds authentication tokens into TCP packet requests and first packet authentication. The system can be used as part of a defense-in-depth strategy to strengthen the security of protected resources in enterprise computing and cloud environments, preventing protected resources from being unnecessary. However, it has not conducted penetration testing and cannot guarantee the emergence of other vulnerabilities.

In 2016, DeCusatis et al. [5] took advantage of the advantages of software-defined network (SDN) to centrally control traffic, designed a trust-based network access path dynamic authorization technology, established a user's trust degree hierarchy, analyzed the user's trust degree in real time, and based on the user's trust degree. Trust and security make real-time adjustments to defense paths. However, it only studies the target user behavior measurement indicators and measurement algorithms, and does not provide a feasible zero trust system implementation architecture, and the proposed indicators and algorithm measurements are not detailed enough, and do not consider highly concealed foul behaviors.

In 2018, Vanickis et al. [37] developed the FURZE system, a risk-adaptive access control policy implementation framework based on Kandala's policy modeling method, to support future security requirements. In FURZE, the application of decision continuity imposes requirements on control functions to maintain session state information so that access control can adapt to the environment or other influencing factors, thereby changing the balance between operational requirements and security risks, and triggering policy re-opening and evaluate. However, it does not include development language tools and is not integrated into the existing professional dynamitic program (PDP) system, so the stability of the runtime mechanism cannot be guaranteed.

In 2020, Yao et al. [38] proposed a dynamic access control and authorization system based on a zero trust security architecture. The system uses the TBAC model, and its user portrait and user trust are generated according to user behavior. The system adopts real-time hierarchical control in different scenarios to achieve dynamic and fine-grained access control and authorization. However, due to the influence of the TBAC model, tasks and roles cannot be clearly separated, and passive access control and role hierarchy are not supported.

In 2021, da Silva et al. [39] proposed zero trust access control with context awareness and behavior-based continuous authentication for smart homes. A zero-aware smart home system is proposed to provide access control to the smart home system using zero trust continuous identity authentication to continuously verify the authenticity of the user, powered by edge computing to eliminate unreliable service providers and access from any means. However, it has not been applied to the actual environment, the impact of latency and concurrency has not been tested, and the accuracy cannot be guaranteed. In the same year, Hatakeyama et al. [40] proposed a new access control model for zero trust networks, which does not assume trusted properties such as source networks. And to verify and evaluate whether the user requesting access is worth relying on each access request, based on the evaluation results, consider the decision of whether to allow access. However, it does not standardize the format and semantics of the context in ZTF and cannot operate the authorization server and the identifier used when the context cannot be shared. In the same year, Mandal [41] et al. proposed a cloud-based zero trust access control strategy by establishing a MAC spoofing defense mechanism in the SDN framework of the cloud architecture to support the work-from-home approach driven by COVID-19. When changes for the access control strategy of the enterprise structure are required, it shows greater accuracy by examining source TCP/IP traffic and corresponding MAC addresses, collecting individual network traffic from untrusted zones. Its AI-based models help lower thresholds and normalize traffic when the network is growing rapidly. However, under the security threat of advanced attackers, the optimal security of lowering the threshold and cloud resources cannot be guaranteed, and the time-consuming nature of analyzing traffic and removing deceived users has not been resolved. Yang et al. [42] proposed an adaptive dynamic access control model based on blockchain and short-term tokens, introduced user trust assessment into the role-based access control model, and used a deep learning-based user abnormal behavior detection algorithm to dynamically evaluate user behavior and update trust, and realize corresponding access rights adjustment on the basis of dynamically updating short-term tokens, but it also has the common problems of RABC: it is difficult to establish an initial role structure and lack of flexibility in IT technology.

Early security policies were divided into two types: discretionary access control (DAC) and mandatory access control (MAC) [43]. However, with the development of computer and network technology, DAC and Mac can no longer meet the needs of practical applications. Therefore, role-based access control (RBAC), object-based access control (OBAC), and task-based access control (TBAC) have emerged. However, with the emergence of new computing environments such as cloud computing and Internet of things, some of its characteristics have brought great challenges to the application of access control technology, which

makes the traditional access control model for closed environment difficult to apply to the new computing environment. Facing the new computing environment with massive, dynamic, and strong privacy [44], the efficiency is very low. The subsequent access control model based on attribute [30] (ABAC) takes the attributes of the subject and object as the basic decision-making elements, and can flexibly use the attribute set of the requester to decide whether to grant its access rights. In addition, the strong expansibility of ABAC enables it to be combined with data privacy protection mechanisms such as encryption mechanism. On the basis of realizing fine-grained access control, protect user data from analysis and disclosure, such as attribute-based encryption (ABE) [45]. The detailed comparison of common access control models is shown in Table 3.

3.4. Review on Trust Assessment Algorithm. A continuous, fine-grained evaluation model is an important part of implementing a zero trust system. The trust evaluation module accepts all kinds of security data monitored and collected by the auxiliary platform, analyzes and judges the data, and forms the trust value of the access request. This trust value will serve as the key basis for the authorization mechanism. The process of quantifying trust will be time-shifted to meet the requirements of high dynamics. Moreover, the level division of trust assessment is also more refined than the traditional model.

Scholars have greatly improved the performance of various aspects of trust assessment in combination with emerging technologies in recent years.

As early as 2018, Jayasinghe U and others [46] and others applied machine learning technology to the node trust evaluation framework, implemented a data credibility labeling method based on unsupervised learning technology, and based on this node trust labeling method. The corresponding trust prediction model is established, which not only improves the accuracy of the trust evaluation algorithm but also improves the ability of the trust evaluation technology to identify trusted interactions. On the basis of the success of this technology, the author also proposes map reduction and data parallelism as research directions, trying to solve the scalability problem existing in the current model.

In 2019, Gao Z and others [47] designed a multidimensional adaptive trust evaluation mechanism using edge computing for nodes with weak computing power that widely exist in the network (especially local edge computing networks), which improved node trust. The robustness of the evaluation, but the flexibility of the model, is not high and should be improved in future research. In the same year, Boussard M and others [48] applied blockchain technology to trust evaluation and proposed a specific trust evaluation framework suitable for home IoT networks, realizing efficient trust evaluation of smart products in home IoT networks. But the single blockchain that this research relies on may be deployed on multiple chains or implemented through channels, and this technology has not yet been successfully developed, so scalability issues remain to be solved. Using the concept of fuzzy logic, Guleng S and others [49] designed a direct trust evaluation method and further proposed a multiagent trust evaluation scheme, considering the characteristics of the entire trust forwarding chain (trust value, forwarding hops, etc.), which improves the accuracy of trust assessment. However, the survivability of the model under abnormal conditions is weak, especially considering the complexity of routing decisions with various constraints such as mobility, bandwidth, link quality, and reliability, and the robustness of this design needs to be improved. In the same year, Rani R and others [50] used game theory to design a lightweight trust evaluation scheme, which not only improved the success rate of malicious node detection but also improved the energy efficiency of trust evaluation. However, the types of external attacks that can be resisted are relatively single, and the comprehensiveness of the protection of trust assessment is relatively poor.

In 2020, Chuan T et al. [51] proposed a method for implementing the concept of zero trust, which described seven evaluation elements of zero trust evaluation: required procedures (including weak password detection procedures, website detection procedures, configuration detection procedures, host vulnerability detection programs, brute force protection programs, hardening programs, mandatory access control programs, and micro-isolation control programs), operating system security vulnerabilities, network security vulnerabilities, weak passwords, high-risk ports, sensitive information protection, and accounts and passwords.

In 2021, Basta N and others [52] adopted microsegmentation technology, which limited the attacker's ability to move laterally in the network by binding fine-grained security policies to a single workload, and initially realized trust assessment under the concept of zero trust. Furthermore, the authors develop an analytical framework to describe and quantify the effectiveness of microsegmentation in enhancing network security. In the same year, in 2021, Zhang Yi and others [53] proposed a trust evaluation optimization mechanism using the fuzzy network analysis method, which effectively refined the evaluation granularity, scientifically quantified the behavioral trust value of users in the cloud computing environment, and improved the evaluation of the objectivity of trust assessment techniques. In the same year, Papakonstantinou N and others [54] used the concept of zero trust to provide a multidisciplinary early design risk assessment framework for early joint safety and security assessment based on the system interdisciplinary dependency model, which more accurately estimated the probability of successful attacks on system components. In the same year, Ramezanpour K and Jagannath [9] adopted reinforcement learning to perform three tasks of trust assessment (i.e., providing an initial network environment risk assessment, learning unnecessary communication flows in devices, and providing models for device communication patterns) and used graph neural networks. The zero trust assessment is modeled by simulating the state of the 5G network, and the application of artificial intelligence in the zero trust assessment is discussed. The application of the intelligent zero trust architecture mentioned by the author in the

Table 3: Comparison of access control models.

Name	Model introduction	Advantage	Limit
Autonomous access control (DAC)	User centered, allowing users to control file access without specifying rules in advance	It is very flexible and can assign access rights between principals and objects	System maintenance and verification of safety principles are very difficult
Mandatory access control (MAC) [44]	Users cannot customize permissions, and access control policies are managed in a centralized manner	Limitations of customer service DAC model	Rely on trusted components
Role-based access control (RBAC) [29]	Assign multiple roles to users and give them permissions and responsibilities as principals	Central management with role members and ACS	Difficult to establish initial role structure and lack of flexibility in IT technology
Organization-based access control (ORBAC)	A more abstract control strategy. It is designed to address topics, objects, and actions. Policies determine which subjects have some actions to access certain objects	Eliminate conflicts between security rules	Vulnerabilities vulnerable to certain types of attacks
Task-based access control (TBAC) [31]	Implement different access control policies for different workflows or different tasks that agree to workflows	When a task is introduced, it can be authorized actively and represent the change of task status	Tasks and roles cannot be clearly separated, and passive access control and role hierarchy are not supported
Attribute-based access control (ABAC) [30]	Approve or reject user requests based on any attributes of the user and selected attributes of objects that may be globally recognized	Subjects can access a wider range of objects and flexibly assign policies and security features	It is difficult to calculate the final permission set of a given user effectively
Policy-based access control (PBAC)	A method of combining roles and attributes with logic to create flexible dynamic control strategies	Flexibility with fine-grained or coarse-grained	Imperfect conflict detection mechanism
Use control (UCON)	It contains three basic elements: subject, object, authority, and three other elements related to authorization: authorization rules, conditions, and obligations	Support trust management and digital rights management, add subject and object attributes, and control them in the process of topic access	Delegation without permission description, explicit management description, and temporal description

article in providing information security in untrusted networks remains to be developed. A detailed comparison of popular evaluation methods is shown in Table 4.

4. Challenge and Future Trends

Today, the basic ZTAs have been determined, but how to make various technologies meet the standard of ZTA is still a difficult problem. At present, the access control, identity authentication, and trust assessment in ZTA are still in the preliminary research stage. In the future, how to use these technologies to enhance the security protection capability and practicability of ZTA is still a hot topic worthy of research. After proposing a new ZTA, how to apply it to the real enterprise network environment is also a challenging research topic.

In identity authentication, because single-factor authentication has only one unique factor for identity authentication, once the unique password or biometrics is stolen, it will collapse completely, while multifactor authentication can improve the defects of single-factor and greatly reduce the threat of network attacks. Because even if the attacker intercepts the password information, the difficulty of obtaining the authorization of the second or third factor is greatly increased. However, the incomplete authentication information is not enough to access. Continuous authentication changes the way that visitors can access system information for a long time after one-time authentication in the initial stage. It continuously grants user resource access rights before and during the session, reducing the security risks caused by attackers in the middle of the session, to enhance the security of the system. From single-factor authentication to two-factor and multifactor authentication, from one-time authentication to continuous authentication, security continues to improve. In the future, multifactor authentication methods and continuous authentication methods will be widely used in zero trust architectures because of their better security. Whether based on certificates, encrypted authentication protocols, or nonencrypted protocols, different protocols have trade-offs between security and resource consumption. It aims to reduce resource consumption as much as possible in the authentication process under the premise of ensuring the security of the zero trust system, which is also the direction of the identity authentication part of the future zero trust architecture.

In recent years, the number and complexity of security attacks against enterprises have risen sharply and will continue to grow in the next few years, which will only increase the complexity of the computing environment. Therefore, the access control system needs to be dynamically adjusted, and the risk assessment has been incorporated into the access control process. The access control decision will include many factors: such as the trust degree of users and devices and the situational environment of users and

Table 4: Comparison of popular evaluation methods.

Name	Evaluation method	Advantage	Disadvantage
Subjective Logic Model	Express the influence of trust parameter factor on trust value in fact space and idea space	More in line with common sense	Difficulty dealing with massive collaborative cheating and defamation
Information Entropy Model	Constructing a trust evaluation algorithm based on entropy increment	Vulnerability was assessed and characterized	Using entropy to represent trust is not comprehensive enough
Weighted Average Model	Fit the weighted average formula of each parameter factor and directly calculate the trust value	The method is simple to execute and the algorithm is efficient	The weight determination process is complicated
Bayesian Model	Use Bayes' theorem to combine prior probabilities with new evidence to get new probabilities	The method is simple and the algorithm is efficient.	Poor subjectivity
Fuzzy Theory	Updating trust vectors using fuzzy logic	Takes into account the "fuzzy" properties of trust itself	Difficulty making rules
Game Theory	Establish a game model for the acquisition of trust information of nodes	Comprehensive evaluation and high accuracy	The game model building process is complicated
Machine Learning	Build a machine learning model as an evaluation algorithm	Strong intuition, high evaluation accuracy, closer to "human" evaluation	Large amount of calculation and low evaluation efficiency

devices, i.e., location, time, task type, and the current security threat level in the user's direct environment. In addition, the access level assigned to devices or users may change over time. The access control system must be able to judge the current trust level by consulting various data sources and making corresponding decisions. Therefore, risk-based access control should be used in many areas. The main result of this trend is to shift from the traditional perimeter-based security model to the application of the so-called zero trust network security model, and treat the enterprise intranet and the Internet to the same extent, that is, lack of trust. Under the zero trust model, we need to solve the problems of minimizing authorization and dynamic authorization control for users. The current zero trust access control model should not be limited to a certain access control model. Only by using a variety of technologies together can we achieve the required level and requirements of access control. For a period of time, RBAC and ABAC will still play an important role in the zero trust model.

Judging from the history of the development of trust assessment technology, the assessment has gradually changed from one-sided to comprehensive, the trust factors considered are more and more extensive, and the theories used in the assessment are gradually enriched, ranging from subjective logic to Bayesian, entropy theory, etc. It makes the trust evaluation close to the subjective evaluation from all directions. Accuracy and fine-grainedness are eternal topics of trust assessment in a zero trust security environment. Not only that, but in different network environments, trust evaluation algorithms also face various requirements. For example, for weak nodes in the network, improving the evaluation efficiency and saving computing resources are as important as ensuring the accuracy of the evaluation. In many applications in social network environments, it is also a trade-off between ensuring the comprehensiveness and accuracy of the evaluation and protecting the user's private information. To sum up, in future research, not only will it be the consistent direction of efforts to continuously improve the comprehensiveness and subjectivity of trust assessment, but also to improve the dynamics and accuracy of trust assessment under the zero trust theory and to conduct the assessment according to the characteristics of different network environments. Adaptive adjustment is also an issue that cannot be ignored.

5. Conclusion

In this article, we expound the concept of zero trust and introduce the background and development of this technology. The core technologies relied on in the ZTA are analyzed in detail: identity authentication, access control, and trust assessment. We analyze and summarize the advantages and disadvantages of existing research on identity authentication, access control, and trust assessment, summarize the urgent problems and challenges of each technology, and propose future efforts and development trends. The work of this paper has guiding significance for the future migration of perimeter-based network security structure to ZTA.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Polato, *Zero Trust Network Architecture with John Kindervag-Video*, 2021, https://www.Paloaltonetwork.com/resources/videos/zero-trust.

[2] R. Ward and B. Beyer, "Beyond Corp: a new approach to enterprise security," vol. 39, no. 6, pp. 6–11, 2014.

[3] A. Alagappan, S. K. Venkatachary, and L. J. B. Andrews, "Augmenting zero trust network architecture to enhance security in virtual power plants," *Energy Reports*, vol. 8, pp. 1309–1320, 2022.

[4] J. Kindervag, *Build Security into Your Network's DNA: The Zero Trust Network Architecture*, Forrester Research Inc, 2010.

[5] C. DeCusatis, P. Liengtiraphan, A. Sager, and M. Pinelli, "Implementing zero trust cloud networks with transport access control and first packet authentication," in *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, pp. 5–10, New York, NY, USA, 2016.

[6] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, *Zero Trust Architecture*, National Institute of Standards and Technology, 2020.

[7] M. Sultana, A. Hossain, F. Laila, K. A. Taher, and M. N. Islam, "Towards developing a secure medical image sharing system based on zero trust principles and blockchain technology," *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, pp. 1–10, 2020.

[8] C. de Weever and M. Andreou, *Zero Trust Network Security Model in Containerized Environments*, University of Amsterdam, Amsterdam, The Netherlands, 2020.

[9] K. Ramezanpour and J. Jagannath, "Intelligent zero trust architecture for 5G/6G tactical networks: Principles, challenges, and the role of machine learning," 2021, https://arxiv.org/abs/2105.01478.

[10] X. P. Tian and H. H. Song, "A zero trust method based on BLP and BIBA model," in *2021 14th International Symposium on Computational Intelligence and Design (ISCID)*, pp. 96–100, Hangzhou, China, 2021.

[11] N. Ghate, S. Mitani, T. Singh, and H. Ueda, "Advanced zero trust architecture for automating fine-grained access control with generalized attribute relation extraction," *IEICE Proceedings Series*, vol. 68, 2021.

[12] L. Henderson, *Multi-Factor Authentication Fingerprinting Device Using Biometrics*, Villanova University, 2019.

[13] Y. Matsuyama, M. Shozawa, and R. Yokote, "Brain signal's low-frequency fits the continuous authentication," *Neurocomputing*, vol. 164, pp. 137–143, 2015.

[14] U. Mahbub, V. M. Patel, D. Chandra, B. Barbello, and R. Chellappa, "Partial face detection for continuous authentication," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 2991–2995, Phoenix, AZ, USA, 2016.

[15] M. Ehatisham-ul-Haq, M. A. Azam, U. Naeem, Y. Amin, and J. Loo, "Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing," *Journal of Network and Computer Applications*, vol. 109, pp. 24–35, 2018.

[16] M. Abuhamad, T. Abuhmed, D. Mohaisen, and D. H. Nyang, "AUToSen: deep-learning-based implicit continuous authentication using smartphone sensors," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5008–5020, 2020.

[17] Y. H. Chuang, N. W. Lo, C. Y. Yang, and S. W. Tang, "A lightweight continuous authentication protocol for the Internet of Things," *Sensors*, vol. 18, no. 4, p. 1104, 2018.

[18] J. Wang, M. Ni, F. Wu, S. Liu, J. Qin, and R. Zhu, "Electromagnetic radiation based continuous authentication in edge computing enabled internet of things," *Journal of Systems Architecture*, vol. 96, pp. 53–61, 2019.

[19] L. Meng, D. C. Huang, J. H. An et al., "A continuous authentication protocol without trust authority for zero trust architecture," *China Communications*, 2022.

[20] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, "DTLS based security and two-way authentication for the Internet of Things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2710–2723, 2013.

[21] U. K. Verma, S. Kumar, and D. Sinha, "A secure and efficient certificate based authentication protocol for MANET," in *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pp. 1–7, Nagercoil, India, 2016.

[22] P. M. Kumar and U. D. Gandhi, "Enhanced DTLS with CoAP-based authentication scheme for the internet of things in healthcare application," *The Journal of Supercomputing*, vol. 76, no. 6, pp. 3963–3983, 2020.

[23] V. L. Shivraj, M. A. Rajan, M. Singh, and P. Balamuralidhar, "One time password authentication scheme based on elliptic curves for Internet of Things (IoT)," in *2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*, pp. 1–6, Riyadh, Saudi Arabia, 2015.

[24] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, "Lightweight and secure session-key establishment scheme in smart home environments," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 254–264, 2016.

[25] S. W. Shah, N. F. Syed, A. Shaghaghi, A. Anwar, Z. Baig, and R. Doss, "LCDA: lightweight continuous device-to-device authentication for a zero trust architecture (ZTA)," *Computers & Security*, vol. 108, article 102351, 2021.

[26] P. Gope and T. Hwang, "Untraceable sensor movement in distributed IoT infrastructure," *IEEE Sensors Journal*, vol. 15, no. 9, pp. 5340–5348, 2015.

[27] B. Ying and A. Nayak, "Anonymous and lightweight authentication for secure vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10626–10636, 2017.

[28] C. M. Chen, B. Xiang, Y. Liu, and K. H. Wang, "A secure authentication protocol for internet of vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

[29] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, "Attribute-based access control," *Computer*, vol. 48, no. 2, pp. 85–88, 2015.

[30] N. Kashmar, M. Adda, and M. Atieh, "From access control models to access control metamodels: a survey," in *Future of Information and Communication Conference*, pp. 892–911, Cham, 2020.

[31] S. Oh and S. Park, "Task-role-based access control model," *Information Systems*, vol. 28, no. 6, pp. 533–562, 2003.

[32] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE symposium on security and privacy (SP'07)*, pp. 321–334, Berkeley, France, 2007.

[33] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 195–203, 2007.

[34] N. Attrapadung and H. Imai, "Conjunctive broadcast and attribute-based encryption," in *International conference on pairing-based cryptography*, pp. 248–265, Berlin, Heidelberg, 2009.

[35] H. Nurmamat, R. Kaysar, and L. Huaizhi, "CP-ABE access control scheme for sensitive data set constraint with hidden access policy and constraint policy," *Security and Communication Networks*, vol. 2017, 13 pages, 2017.

[36] Y. Baseri, A. Hafid, and S. Cherkaoui, "Privacy preserving fine-grained location-based access control for mobile cloud," *Computers & Security*, vol. 73, pp. 249–265, 2018.

[37] R. Vanickis, P. Jacob, S. Dehghanzadeh, and B. Lee, "Access control policy enforcement for zero-trust-networking," in *2018 29th Irish Signals and Systems Conference (ISSC).*, pp. 1–6, Belfast, UK, 2018.

[38] Q. Yao, Q. Wang, X. Zhang, and J. Fei, "Dynamic access control and authorization system based on zero-trust architecture," in *2020 International Conference on Control, Robotics and Intelligent System*, pp. 123–127, Xiamen, China, 2020.

[39] G. R. da Silva, D. F. Macedo, and A. L. dos Santos, "Zero trust access control with context-aware and behavior-based continuous authentication for smart homes," *Anais do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pp. 43–56, 2021.

[40] K. Hatakeyama, D. Kotani, and Y. Okabe, "Zero trust federation: sharing context under user control towards zero trust in identity federation," in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 514–519, Kassel, Germany, 2021.

[41] S. Mandal, D. A. Khan, and S. Jain, "Cloud-based zero trust access control policy: an approach to support work-from-home driven by COVID-19 pandemic," *New Generation Computing*, vol. 39, no. 3-4, pp. 599–622, 2021.

[42] K. Yang, D. Li, L. Zhou, and K. Cheng, "Research on adaptive dynamic access control model based on blockchain and token," *Journal of Physics: Conference Series*, vol. 2166, no. 1, pp. 12–14, 2021.

[43] Y. Zhang and Y. Zhang, "A survey of zero trust research," *Journal of Information Security Research*, vol. 6, no. 7, pp. 608–614, 2020.

[44] L. Fang, L. H. Yin, Y. C. Guo, and B. X. Fang, "A survey of key technologies in attribute-based access control scheme," *Chinese Journal of Computers*, vol. 40, no. 7, pp. 1680–1698, 2017.

[45] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 62–91, Berlin, Heidelberg, 2010.

[46] U. Jayasinghe, G. M. Lee, T. W. Um, and Q. Shi, "Machine learning based trust computational model for IoT services," *IEEE Transactions on Sustainable Computing*, vol. 4, no. 1, pp. 39–52, 2019.

[47] Z. Gao, W. Zhao, C. Xia et al., "A credible and lightweight multidimensional trust evaluation mechanism for service-oriented IoT edge computing environment," in *2019 IEEE International Congress on Internet of Things (ICIOT)*, pp. 156–164, Milan, Italy, 2019.

[48] M. Boussard, S. Papillon, P. Peloso, M. Signorini, and E. Waisbard, "STewARD: SDN and blockchain-based trust evaluation for automated risk management on IoT devices," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 841–846, Paris, France, 2019.

[49] S. Guleng, C. Wu, X. Chen, X. Wang, T. Yoshinaga, and Y. Ji, "Decentralized trust evaluation in vehicular Internet of Things," *IEEE Access*, vol. 7, pp. 15980–15988, 2019.

[50] R. Rani, S. Kumar, and U. Dohare, "Trust evaluation for light weight security in sensor enabled Internet of Things: game theory oriented approach," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 8421–8432, 2019.

[51] T. Chuan, Y. Lv, Z. Qi, L. Xie, and W. Guo, "An implementation method of zero-trust architecture," *Journal of Physics: Conference Series*, vol. 1651, no. 1, pp. 1–7, 2020.

[52] N. Basta, M. Ikram, M. A. Kaafar, and A. Walker, "Towards a zero-trust micro-segmentation network security strategy: an evaluation framework," 2021, https://arxiv.org/abs/2111.10967.

[53] Y. Zhang, Y. Tian, Z. Wu, and W. Wu, "Trust evaluation optimization mechanism for cloud user behavior based on FANP," *Chinese Journal of Network and Information Security*, pp. 1–9, 2021.

[54] N. Papakonstantinou, D. L. van Bossuyt, J. Linnosmaa, B. Hale, and B. O'Halloran, "A zero trust hybrid security and safety risk analysis method," *Journal of Computing and Information Science in Engineering*, vol. 21, no. 5, pp. 1–10, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4087816, 17 pages
https://doi.org/10.1155/2022/4087816

Research Article

A Study on Scalar Multiplication Parallel Processing for X25519 Decryption of 5G Core Network SIDF Function for mMTC IoT Environment

Changuk Jang,[1] **Juhong Han**,[1] **Akshita Maradapu Vera Venkata Sai**,[2] **Yingshu Li**,[2] **and Okyeon Yi**[3]

[1]*Department of Financial Information Security at Kookmin University, Seoul 02707, Republic of Korea*
[2]*Department of Computer Science at Georgia State University, Atlanta, GA 30303, USA*
[3]*Department of Information Security Cryptology and Mathematics at Kookmin University, Seoul 02707, Republic of Korea*

Correspondence should be addressed to Okyeon Yi; oyyi@kookmin.ac.kr

Received 10 March 2022; Accepted 30 April 2022; Published 6 June 2022

Academic Editor: Qiang Ye

When 5G telecommunication becomes a standardized and widely used communication medium, it must be implemented in coherence with certain 5G network standards and requirements. One such requirement is a Subscription Concealed Identifier called SUCI. SUCI prevents the exposure of international mobile subscriber identity (IMSI), which was a vulnerability in previous generation mobile telecommunication networks. Unlike IMSI, SUCI is encrypted and transmitted using a symmetric key cryptographic algorithm, to prevent the aforementioned vulnerabilities. However, for the first terminal to be encrypted, it is necessary to exchange a key with the home network, and this key exchange for SUCI encryption is performed through the Elliptic Curve Integrated Encryption Scheme (ECIES) key exchange algorithm, which is a public-key encryption scheme. However, ECIES uses more computing resources compared to a symmetric key cryptographic algorithm. Additionally, for 5G Subscriber Identity Deconcealing Function (SIDF) to satisfy the massive machine-type communication (mMTC) requirements of 5G, it is necessary to decrypt at least a million SUCIs within a short time. This puts a great burden on the 5G home network to provide the mMTC service for IoT. Therefore, in this paper, we propose a method of constructing 5G SIDF in an mMTC IoT environment. A key method of the proposed 5G SIDF configuration is the use of GPUs. This proposal was aimed at reducing the load in the mMTC environment by performing parallel processing of all cryptographic operations performed in the SIDF using a GPU. In particular, we focused on parallelization of public-key encryption algorithms. In addition, we also compared the method proposed in this paper through a survey of various 5G security products.

1. Introduction

The fifth-generation (5G) telecommunication technology needs to follow the IMT-2020 Standard (International Mobile Telecommunications-2020) [1] requirements defined by the ITU-R (ITU-Radio communication sector) of the International Telecommunication Sector in 2015 for 5G networks. This standard defines the evolution of telecommunication with respect to various technical requirements. Based on this, the 3rd Generation Partnership Project (3GPP), an international standardization organization for mobile communications, standardized 5G NR (New Radio) in 2016 and has been contributing to the commercialization of 5G technology until Release 17 as of 2021. 5G telecommunication has caused various changes in the network and security fields. The three requirements called enhanced mobile broadband (eMBB), massive machine-type communication (mMTC), and ultrareliable low-latency communication (URLLC) govern the 5G telecommunication as it revolutionizes existing mobile communication. In other words, 5G telecommunication networks will provide services while satisfying one of these three requirements. As such, with the advent of the three 5G services, the communication environment of the Internet of Things

(IoT) has been further developed. Among them, this paper focused on the mMTC service. The mMTC service area is designed for massive IoT deployments using large numbers of low-power devices to regularly transmit small amounts of data. With the increasing popularity of intelligent transportation, smart city, etc., it is envisioned that the number of IoT devices will reach 75 billion by 2025, which is much larger than the number of the mobile phone users. To provide wireless connectivity to such a large number of devices by the time IoT comes to fruition, 3GPP has identified mMTC as one of the three main use cases of the 5G wireless systems [1]. As such, providing 5G mMTC service is very positive for the IoT environment.

However, 5G mMTC also has a big problem. From 5G mobile communication, the public-key encryption method is applied in the Authentication and Key Agreement (AKA) process of the 5G terminal (UE). When a public-key encryption scheme is applied, both the UE and the home network become a great burden. Particularly from the perspective of the network, it should be able to handle the simultaneous access and authentication process of up to 1,000,000 UEs to support the mMTC service. In this paper, we focused on the 5G Subscriber Identity Deconcealing Function (SIDF), which is in charge of public-key cryptography. This function is always used for initial access authentication to the terminal, and a load may occur if many terminals suddenly try to access it, such as in the mMTC environment. Unlike the core network up to 4G, the 5G's core network is designed through a software-based core structure; therefore, each network function (NF) can be configured as a module and the core network can be configured to meet various requirements. Therefore, in this paper, we propose a parallelization scheme using graphics processing unit (GPU) to improve the ECIES operation speed of the 5G telecommunication core network that satisfies mMTC.

The contributions to this paper are as follows:

2. Proposal of X25519 ECIES Decryption Parallel Processing Using GPU

This paper proposes to speed up the ECIES operation used in 5G AKA from the perspective of the home network. Various existing ECC speeding studies mentioned in this paper suggest a method for speeding up fixed scalar computation. That method is ultimately only available for the ECIES encryption method. However, in this study, to speed up ECIES decryption, we propose a method of predividing a fixed scalar and speeding up the random scalar computation using GPU.

3. Suggestion of 5G SIDF Function Construction Using GPU for mMTC IoT Environment

This paper presents a method to configure the SIDF function using GPU to speed up 5G AKA in the mMTC environment. The functions of the 5G core network are implemented the in software to freely configure the network. Therefore, in this paper, when the 5G home network SIDF performs the ECIES decryption operation, a new approach is introduced to quickly perform the 5G key agreement process.

4. Comparison with Commercial 5G Security Product

To show the superiority of the proposed SIDF function, we survey and compare cryptography and security function products used by various 5G carriers.

5. Background

5.1. 5G SUCI. Until the implementation of 4G, international mobile subscriber identity (IMSI) was used as a subscriber identifier. The corresponding identifier was transmitted to the network upon initial access to the terminal. At this time, a vulnerability was exposed in the wireless section [2, 3]. This vulnerability became a problem in the development of subsequent telecommunication standards. To solve it, various identifiers have been encrypted and transmitted starting from the implementation of 5G telecommunication. Consequently, SUCI has been developed [4–6]. SUCI is a concept first introduced in 5G telecommunication networks and is a value transmitted by concealing (encrypting) the identifier in the UE to prevent identifier's exposure when transmitted to the wireless section. Therefore, even if the corresponding value is captured in the wireless section, the terminal identifier value is not exposed because it is encrypted. Figure 1 demonstrates the structure of SUCI. Among the SUCI fields, the "Protection Scheme ID" field specifies how to conceal the identifiers with SUCI, in which 0×0 denotes "NULL = scheme", 0×1 denotes "ECIES Profile A (curve 25519)", and 0×2 denotes "ECIES Profile B (secp256r1)." Subsequently, to conceal an identifier, a public-key encryption scheme using an elliptic curve cryptography is used.

5.2. ECIES in 5G Security and X25519. ECIES is a hybrid encryption scheme that encrypts data using symmetric keys created through an elliptic curve cryptography-based key exchange method, a public-key cryptography method [7]. In 5G, ECIES is used to encrypt the MSIN in the IMSI and securely transmit it to the 5G home network (5G HN). Therefore, the UE side of 5G communication uses the ECIES encryption method as shown in Figure 2, and the SIDF of 5G HN uses the ECIES decryption method described in Figure 3. First, the UE must generate a temporary encryption key to encrypt the MSIN. To this end, a key exchange method using elliptic curves is used. The UE generates K different keys randomly for each access session. Then, the key material of the ephemeral shared key, namely, ephemeral public key R, is calculated. The shared value Z is generated using K and the home network public key Q. Then, encryption and MAC keys are constructed using the generated R and Z as inputs for the key-inducing function. Subsequently, the MSIN in IMSI is encrypted using the AES-128-CTR algorithm, and a MAC value (HMAC-SHA-256) is generated to verify that the correct value is encrypted. When the home network's UDM receives a SUCI signal from AUSF

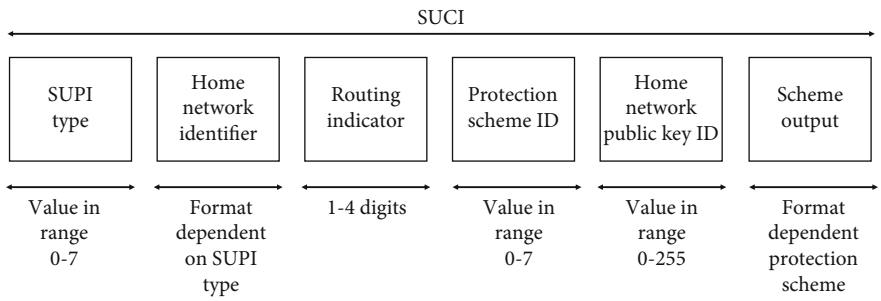


Figure 1: SUCI structure: SUCI is a concept first introduced in 5G telecommunication networks and is a value transmitted by concealing (encrypting) the identifier in the UE to prevent identifier's exposure when transmitted to the wireless section.

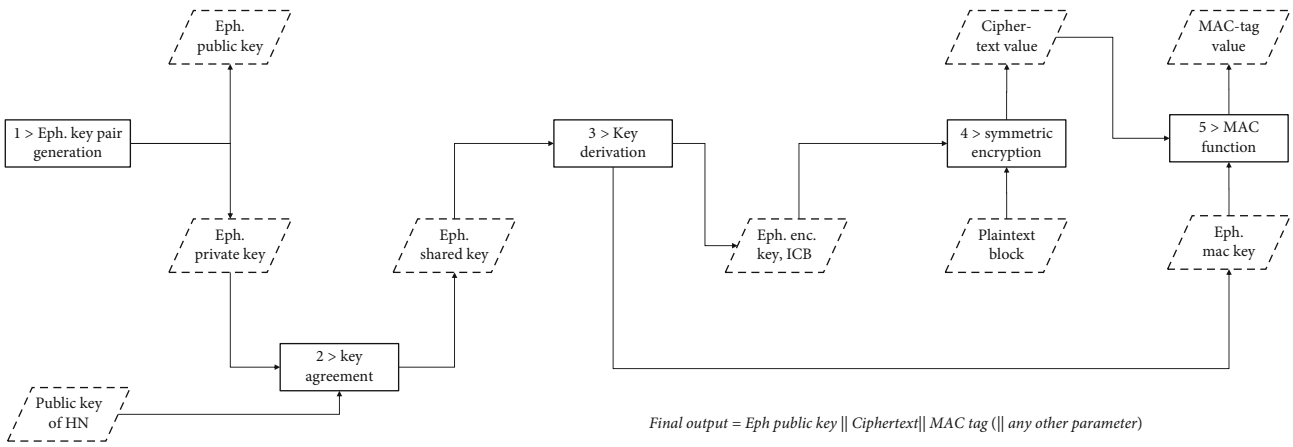


Figure 2: ECIES encryption process on UE side: in the UE (USIM), the ECIES encryption process is performed. In this step, a 128-bit encryption key and a 128-bit mac key used to generate SUCI are generated. Thereafter, the MSIN, which is used in 4G subscriber identifier, is encrypted to generate SUCI.

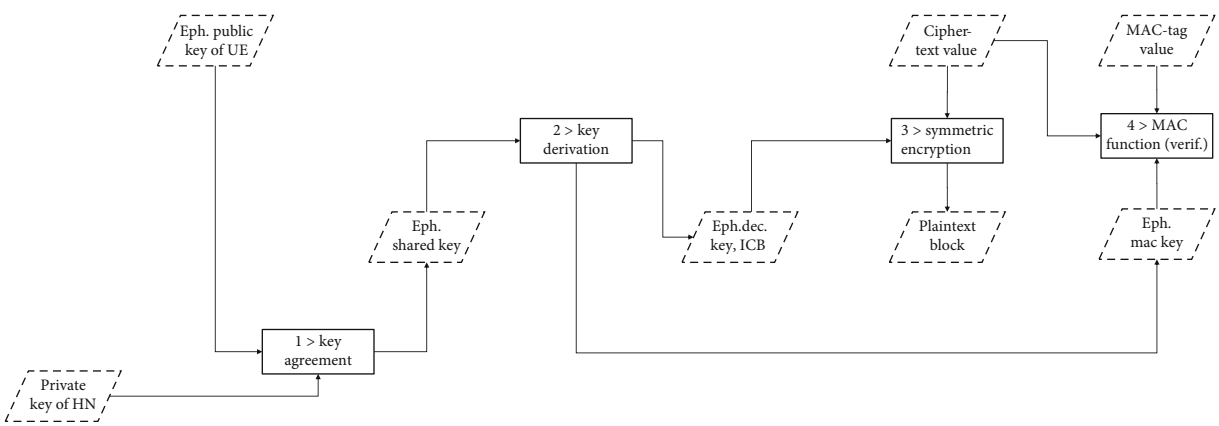


Figure 3: ECIES decryption process on home network side: in the SIDF in the home network, the ECIES decryption step is performed. In this step, a 128-bit encryption key and a 128-bit mac key are generated to decrypt SUCI. Thereafter, the MSIN is generated by decoding the SUCI transmitted from the UE.

on the serving network, the SIDF function is called. SIDF uses the received ECC ephemeral public key of the UE and private key of the home network to generate an ephemeral shared key. Next, the SIDF decrypts the MSIN that was encrypted and transmitted. After that, the KDF function generates the encryption and MAC keys and deconceals the SUCI of the UE. After this process, the HN determines the AKA method to be utilized with the UE based on deconcealed SUCI before its execution.

Elliptic curve cryptography (ECC), a public-key cryptography scheme used in the key agreement process, was independently introduced by Koblitz and Miller in 1985 [8, 9], addressing the existing discrete logarithm problem (DLP) with a much shorter key length and faster speed.

Therefore, it is used for key agreement or digital signature tasks in various small IoT environments or smart card environments. Because of its advantages, ECC was used for the first time in a communication network with the adoption of 5G technology. Because the ECC curve equation varies according to the definition of curves and parameters, it is important to determine which curve is used among the objects accurately. Therefore, in 5G telecommunication networks, profiles A and B are defined a priori. Profile A is the X25519 parameter on Curve25519 [10], and profile B is the secp256r1 curve [11].

5.3. Security Availability for 5G Services. Unlike 4G telecommunication security standards, the 5G security standards are diversified through authentication and key agreement methods, key hierarchy is more refined, and a public-key cryptographic scheme called Elliptic Curve Integrated Encryption Scheme (ECIES) is applied for the first time. ECIES has not been considered in the 4G security standard because of a required longer key length and slow operation compared to the symmetric key cryptography. Consequently, 5G telecommunication must satisfy both network and security requirements, and this is a challenging task.

In the eMBB service, improved transmission speed, maximum transmission speed, and terminal mobility are provided by improving the service quality of the existing 4G mobile broadband (MBB), and the URLLC service refers to a communication service that has a very high reliable user data transmission with a very short delay time. Therefore, in order for the eMBB service to transmit high-capacity data and provide maximum transmission speed, it is essential to speed up the data plane, signal data encryption/decryption, and integrity functions among 5G security technologies. URLLC service is particularly needed to speed up encryption/decryption functions, mainly the integrity functions in the data plane's wireless section. However, even in 3GPP TS 33.501, the 5G security standard issued by 3GPP, encryption/decryption and integrity functions of user data and signalling data are mostly classified as "optional" [4]. In addition, the network overhead caused by using the encryption/decryption and the integrity functions can be a great burden on the mobile operator. Therefore, many mobile carriers adopt the NULL-scheme, where they do not provide encryption/decryption and integrity functions. However, as mentioned above, security vulnerabilities are found because 5G mobile communication is used in various environments, so it is essential to provide a cryptographic algorithm availability.

In this paper, we want to focus on the mMTC environment in addition to the above two scenarios. Unlike eMBB, mMTC targets IoT devices that exchange relatively low-capacity data at low speed. The mMTC service defines the maximum number of terminals accessing the network as 1 million per unit area (1 square kilometer). For the 1 million devices to connect to the network and communicate with each other, it is absolutely necessary to speed up the initial authentication procedure on the core network of 5G encryption technology. However, the biggest problem is that, unlike eMBB and URLLC, the public-key encryption method using ECIES must perform up to 1,000,000 times in the initial authentication process. Therefore, to provide initial authentication and key agreement functions in the mMTC environment, the fastest and most accurate ECIES public-key operation is required for terminals and 5G core networks.

Secondly, the mobile communication technology defined in international standards is used for more than one generation; therefore, one has to take into consideration the restrictions associated with each generation. Finally, the upcoming 6G telecommunication technology is expected to be governed by faster dynamics than those of the current 5G telecommunication requirements, with the aim of expanding to more connected devices. In this case, faster and more accurate cryptographic operations are required. In 5G telecommunication environments, because the number of small terminals, using the universal subscriber identity module (USIM), in the existing IoT environment increases rapidly, various entities in the 5G core network must be able to simultaneously process transmissions from numerous terminals. In particular, the mMTC requirements must cover up to 1,000,000 devices per square kilometer.

5.4. GPGPU. Many personal computers use GPUs for computer graphics operations and screen resolution processing. A method for achieving parallel computing using the GPU hardware for various purposes is called General-Purpose Computing on Graphic Process Unit (GPGPU). In particular, NVIDIA has developed so-called "Compute Unified Device Architecture" (CUDA), a general-purpose parallel programming framework that uses a programming language, such as C language, for compilation and execution of GPU programming in 2007. CUDA framework can be used in both Windows and Linux environments. Hence, many application developers can implement various functions, such as AI, accelerated computing, and cryptocurrency, using NIVIDA GPUs [12]. The structure of NVIDIA GPUs using the CUDA framework has been gradually developed. The structure of a physical GPU consists of multiple streaming multiprocessors (SMs) and streaming processors (SPs) that execute the program codes in each SM. SPs have the same structure as a CUDA core in terms of hardware. The logical GPU structure is organized in the order of threads-blocks-grids, as shown in Figure 4 [13]. A thread is the minimum command processing unit, and usually, 32 threads are gathered to form one warp, which is the smallest unit of execution. The threads gathered in a warp form a block, and the blocks form a grid. Each logical structure depends on the resources of the GPU, while the warp size is usually composed of a multiple of 32 threads. Therefore, the warp unit is the most important when performing regular parallel programming.

To leverage the GPU effectively, one needs to understand how a GPU works. The GPU bundles several threads and executes the same commands in one warp unit. Therefore, it should be implemented to ensure control divergence does not occur, which is achieved by minimizing branch statements. Moreover, because the GPU resource that has the greatest influence on block configuration per SM is the registers, to increase occupancy, the

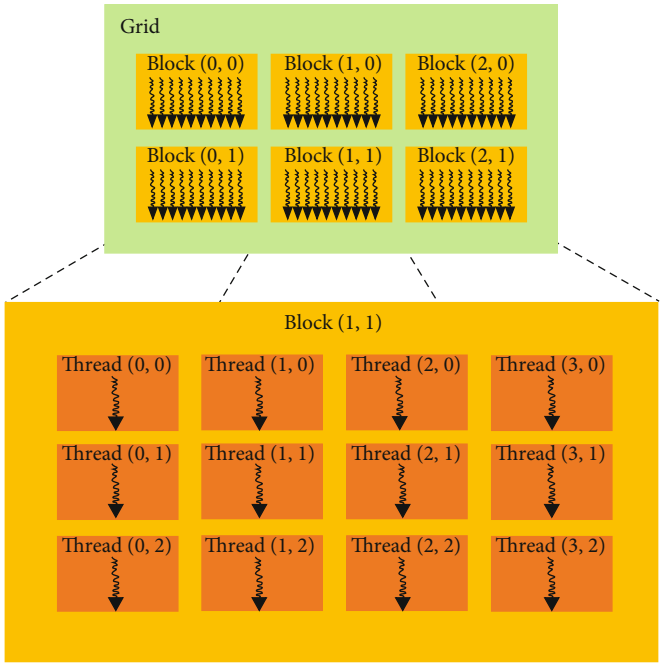


Figure 4: Thread hierarchy in CUDA programming: this figure is the logical structure of CUDA programming. As shown in the figure, it has a structure in which the kernel function operates simultaneously through the grid-block-thread stage. In this paper, the ECIES decryption kernel function operates through the structure as well.

number of active blocks should be increased by optimizing the registers used per thread.

6. Related Work

Cryptographic algorithms are used in various environments, such as small IoT environments, smart cards, and parallel computing environments. When providing security services by implementing cryptographic algorithms in various environments, because the physical capabilities of each environment are different, various methods are used to increase the operation speed or memory efficiency in environments with insufficient memory. Particularly, there are numerous restrictions when providing a security service using public-key cryptography algorithms. Many studies were aimed at improving speed and memory efficiency because basic operations, such as big-number operations of public-key cryptography algorithms consume more memory and are slower than symmetric key cryptography algorithms. In addition, public-key cryptosystems based on discrete logarithmic problems, such as RSA, DSA, and DH, are disadvantageous in terms of speed and memory efficiency compared to symmetric key cryptosystems. Therefore, ECC [7, 8], which provides security services such as digital signature and key exchange services, can outperform the existing public-key encryption system due to its short key length and has become popular in various environments, including small IoT applications. Therefore, most researches are focused on high-speed implementations in small chips with insufficient computing resources [14–17].

6.1. ECC Parallel Implementation Study Using GPU. As the GPGPU era started, cryptographers worldwide have conducted research to improve the speed of scalar multiplication operations in ECC algorithms using GPUs, defining a new computing environment. [18, 19] used the CUDA framework and obtained parallelization results for fixed scalar multiplication and fixed point using a single thread. However, in [18], the parallelization results based on discrete logarithm problems were mainly implemented. Thus, the ECC algorithm utilized only the NIST P-224 curve. On the other hand, [19] focused on the high-speed implementation of GMP-ECM using the Edward elliptic curve in various GPUs. Furthermore, in [20], high-speed implementation of ECC scalar multiplication through GPU programming was studied. This study demonstrated the parallelization results of various elliptic curves over gf (2^m). Unlike the previous studies, the performance was optimized by calculating the random and fixed scalar products differently. In [21], by optimizing the algorithm at the assembly level using CUDA Parallel Thread Execution (PTX) instruction set architecture (ISA) using GPUs in X25519 key agreement, an efficient method for modular addition and multiplication of Curve25519/448 and concise reduction arithmetic is proposed. In this paper, unlike [20], the ECC modular multiplication operation is configured as a PTX operation, and scalar multiplication of the base point and scalar multiplication of an unknown point multiplication are all implemented by using one thread for each scalar multiplication. Finally, in [22], the DPF-ECC framework was presented to accelerate the ECC system by maximizing the double precision floating point (DPF) computing power. Unlike the above papers, this

paper proposes a new framework that applies DPF operation to ECC. In particular, it showed up to 3 times the amount of throughput compared to the previous research results on the Edwards25519/448 curve as well as in a specific prime number computation using GPU.

6.2. 5G Core Network Security Solution Survey. Implementing security functions within 5G systems is standardized to a limited extent by 3GPP. Therefore, 5G equipment manufacturers need to implement 5G network functions such as AUSF, ARPF, SIDF, and UDM related to subscription authentication when connecting to a 5G UE network and implement various encryption keys and algorithms generated as a result of UE authentication. However, 5 g network solutions will be subjected to various stress tests due to the growth of 5G and industrial digitalization. Accordingly, various equipment manufacturers must provide an additional level of security to software-only solutions for security functions (subscription authentication, encryption, and key matching) standardized by 3GPP when building 5G network equipment and increasing the demand for network capacity. In particular, equipment manufacturers provide integrated management by building security solutions based on the cloud to satisfy the 5G characteristics of SDN, network slicing, and 5G NF design ideas. In this section, we would like to mention the solutions of two representative companies, Ericsson and Nokia. Ericsson, a leading 5G equipment company, configured Cloud Core Subscription Manager (CCSM) by combining UDM, HSS, AUSF, and EIR as a 5G authentication solution to respond to this problem. This CCSM is composed of a cloud-native 3GPP network function. The solution provides an authentication security module to provide 5G security functions standardized by 3GPP and provides various security and encryption functions by linking 3GPP ARPF and HSM provided by Thales [23]. In addition, Nokia, a leading 5G equipment company, has been engaged in 3GPP 5G standardization work and has been engaged in various activities such as 5G security function proposals and patent registration. In particular, in order to respond to the problems of 5G security threats, UDM, HSS, AUSF, HLR, EIR, etc. are bundled together as a 5G authentication solution, and Subscriber Data Management (SDM) and the integrated authentication solution Authentication, Authorization and Accounting (AAA) are integrated into 5G configured in the network [24].

6.3. 5G SIDF Open-Source Project Review. In Open5gs [25], an open-source 5G network project, SIDF is not implemented separately but is included in the UDM function. Moreover, there was no separate implementation of public-key cryptography, except for the implementation (embedded) of the null type. Looking at the SIDF implementation in Free5GC [26], another 5G network open-source project, the SIDF function was not implemented separately, similar to Open5gs. However, it differed from Open5gs regarding the ECC algorithm implementation for SUCI profiles A and B, which was implemented in UDM to deconceal SUCI. Because Free5GC is a 5G network-based open-source project, it has a format that is called every time a single HTTP message is transmitted. Particularly, the encryption algorithm was implemented in Go Language with no specific high-speed technique applied [27]. Looking at OpenAirInterface5g [28], another 5G network open-source project, the OpenAirInterface5g project is divided into 5G RAN and 5G CN. In particular, the OpenAirInterface5g CN project was aimed at making a CN stack fully compatible with 3gpp. The OpenAirInterface5g CN project is currently in progress up to version 1.3 and includes various 5G NFs such as AMF, AUSF, and NRF. The most prominent feature with other open sources is that C and C++ are used as the major languages, so there is an advantage for 5G network configurators to use these open sources in the actual 5G environment. However, as in the above two open-source projects, the SIDF function is not implemented separately; in particular, the method of handling SUCI within AUSF and UDM is not specified, and it is hard coded in the 5G RAN project. Like [25, 26, 28], the 5G core network has the characteristic of freely configuring the network by implementing each NF in the software. This is not only a 5G design idea but has several advantages. However, when composing SIDF with software alone, it was judged that the ECIES operation speed was slow like the two open sources, or it could not satisfy the mMTC requirements. Subsequently, in our study, when the 5G home network SIDF performs the ECIES decryption operation, instead of using the methods introduced in the abovementioned studies, we introduce a method that quickly performs the 5G key agreement process with a novel approach.

7. Suggested Method: Parallel ECIES Decryption Using GPU

7.1. Parallel 5G SIDF Construction Using a GPU. In the abovementioned studies, high-speed implementations of ECC scalar multiplication were studied in various ways. However, in the SIDF framework for 5G telecommunication networks, the ECIES decryption must be performed differently than the ECIES encryption step that performs the fixed point and random scalar multiplication operations similar to those performed in the abovementioned studies. The SIDF of the home network performing the ECIES decryption step must perform a fixed scalar product (home network private key, d) and assign it to a random point (ephemeral public key, R) for network access of one terminal. In addition, when many terminals are accessed simultaneously, it is necessary to multiply a fixed scalar by a number of random points. Therefore, in the ECIES decryption step, the random scalar multiplication operation on various fixed points cannot be performed similarly to those proposed above. Here, the 5G telecommunication home network uses a GPU to speed up the random point and fixed scalar multiplication operations, which requires the most computation time and resources during the ECIES decryption process. This method is expected to satisfy the mMTC requirement when numerous terminals access the 5G network. The basic configuration of the proposed 5G SIDF is depicted in Figure 5.

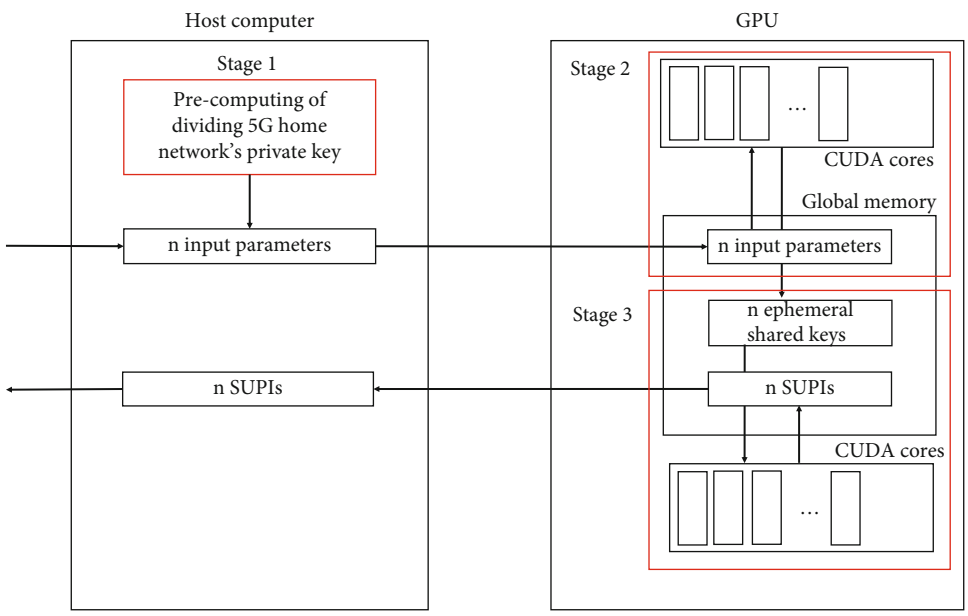


Figure 5: Suggested model of 5G SIDF using GPU: it is suggested to configure 5G SIDF in mMTC environment as shown in the diagram. The SIDF proposed in this paper consists of a host computer and GPU. In particular, in the GPU, the ECIES decryption step and the SUCI decryption step are performed using parallel computing technology.

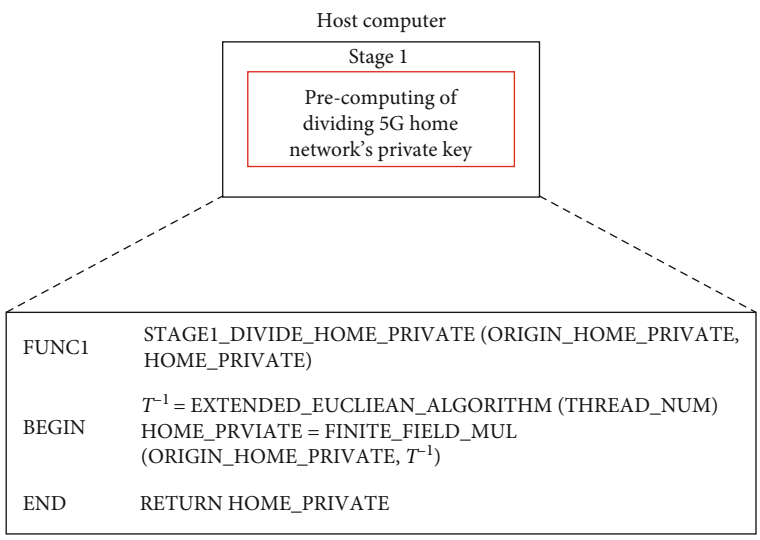


Figure 6: Stage 1 procedure: stage 1 performs precalculation. This step works on the host PC. In the precomputation stage, the private key of HN is divided through a finite field operation.

The SIDF parallelization structure using the GPU is divided into three stages.

7.1.1. Stage 1. Stage 1 is the precomputation part that divides the private key of HN according to the number of threads on the GPU. This method was not found in the previous studies. In the ECIES decryption stage, public key and private key pair of HN are a fixed value that continues to be used as long as the validity period has expired and is not changed according to Telco's policy. Therefore, the private key value of the HN is fixed and used whenever numerous UEs access the network. In addition, in SIDF, an operation of multiplying the private key of HN at different random points transmitted by numerous UEs is performed. Therefore, in this paper, the scalar, which is a fixed value to be used for parallel operation of the GPU, is divided and used through stage 1, the precomputation stage.

Stage 1 operates like FUNC1 in Figure 6. FUNC1 operates on the host PC. FUNC1 basically receives private key of HN. The private key of HN is a scalar value within the elliptic curve finite field. Therefore, in order to speed up the scalar multiplication operation performed during ECIES decryption, the private key of HN is divided by the GPU thread value (T). The calculation should be done on the finite filed using elliptic curve. However, in the finite field, the division operation is not defined separately, so the

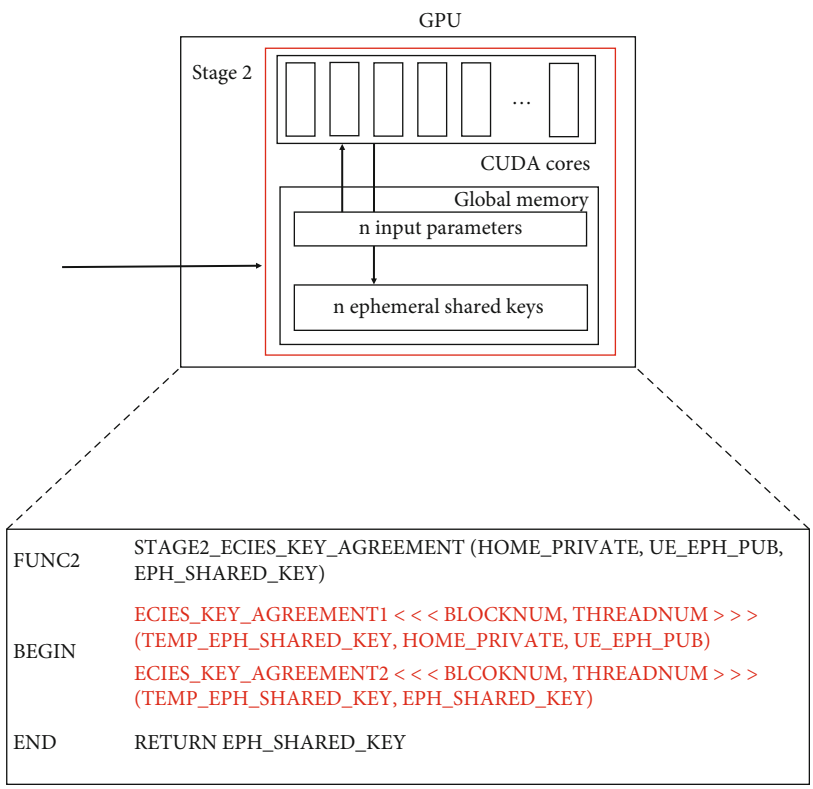


Figure 7: Stage 2 procedure: stage 2 calculates the ephemeral shared key. These steps run on the GPU. Stage 2 receives the value generated in stage 1 and the random point (ephemeral public key) received from the UE and performs ECC scalar multiplication operation.

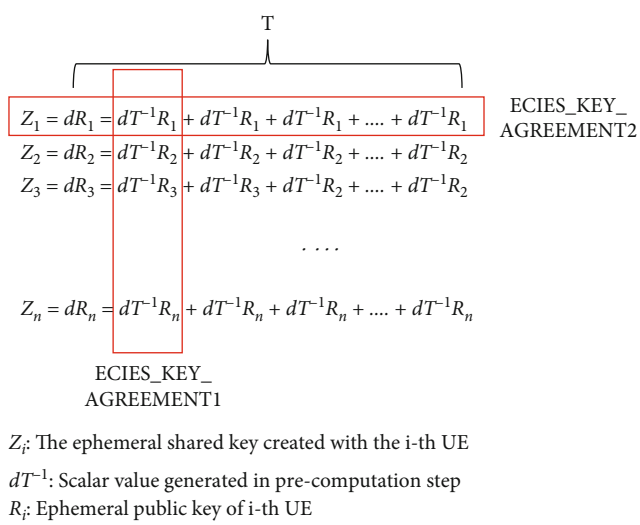


Figure 8: Stage 2 kernel function concept formula: in fact, the contents that operate in the kernel function of stage 2 are expressed as a formula.

method of multiplying the value to be divided by the inverse must be used. Therefore, it is necessary to find the inverse (T^{-1}) of the value T in the finite field. However, the GPU thread value is fixed depending on the GPU used. And in general, GPU thread size is a multiple of 32. Therefore, the T^{-1} is also a fixed value. Therefore, the precalculation stage, stage 1, is used only when the key pair of the HN is changed or the corresponding value is deleted because the power is turned off and generally does not operate when the UEs access.

The reason for performing stage 1, the precomputation stage, is that the scalar multiplication operation speed becomes slower as the length of the ECC private key increases, as demonstrated in various studies [29, 30]. The reason scalar multiplication takes longer as the ECC key length increases is because the key length is related to the size of the point. As a result, stage 1 makes it possible to reduce the amount of elliptic curve scalar product computation in stage 2 running on GPU.

7.1.2. Stage 2. Stage 2 calculates the ephemeral shared key Z. This step is the most essential part of the thesis. The GPU operates in a way that many cores simultaneously process the same operation with one source code. For example, if a GPU is configured to perform an addition operation, multiple cores derive a result by only performing an addition operation on a given input parameter. Therefore, an efficient configuration of GPU resources is required. This step consists of an ECC scalar multiplication operation part and ECC point addition operation part.

First, in the ECC scalar multiplication operation part, the UE's ephemeral public key transmitted from the UE is multiplied by the private key of HN divided in advance in stage 1. At this time, because private key of HN may not be a multiple of the operation unit, it is not easy to ensure a perfect resource. However, since the number of UEs connected to

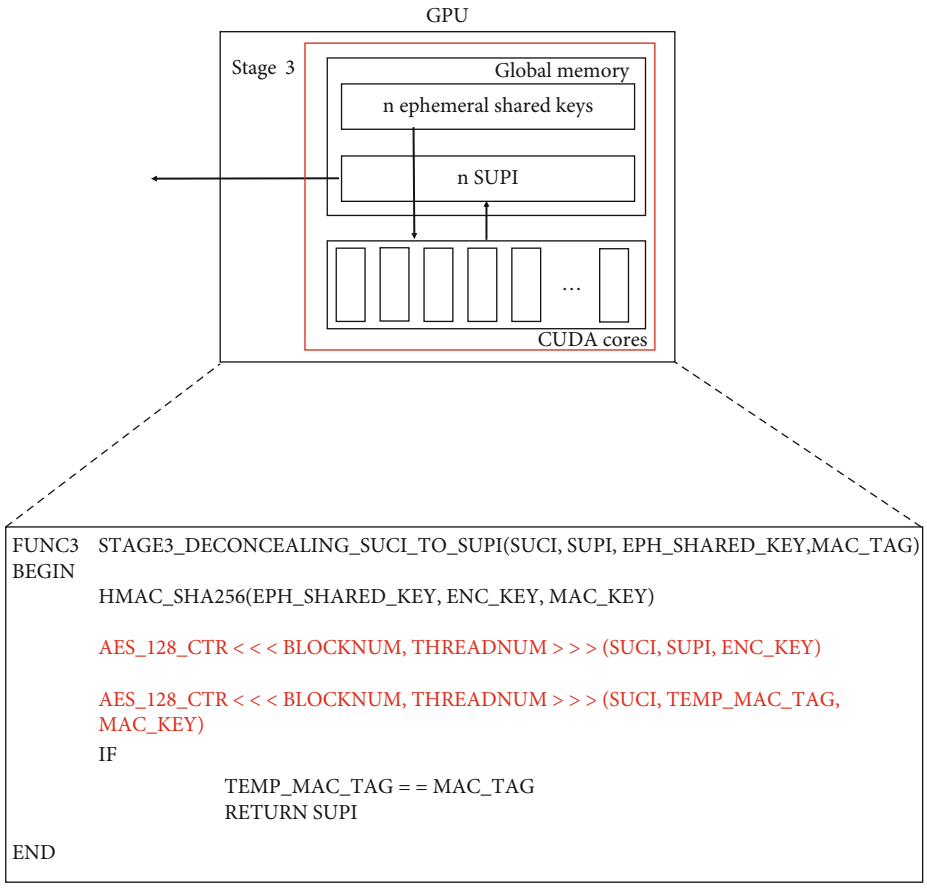


Figure 9: Stage 3 procedure: obtain encryption key and MAC key using the ephemeral shared key, the result of step 2 in step 3. The block cipher (AES) operation runs on the GPU.

the HN is exceedingly large, performing scalar multiplication by dividing in advance in stage 1 achieves better efficiency. In addition, because the HN has one public key pair that is used for a long time, the SIDF can be fixed and configured without changes for each UE connection. In this way, the scalar multiplication operation part multiplies R_i transmitted by each UE by dT^{-1}. The corresponding value is the same as the value obtain by dividing the ephemeral shared key by T. This part works on ECIES_KEY_AGREEMENT1 in Figure 7.

ECC points multiplied in this manner must be added together once again to match the UE index to form an accurate ephemeral shared key. Therefore, the ECC points calculated for each UE are stored after the scalar multiplication; then, the ECC point addition operation is performed. Because the ECC point addition operation is invariant and the number of points added for each UE is fixed, it can proceed without processing each UE approach separately. If the precomputation step is omitted, the Z value can be obtained directly using the scalar multiplication operation by inputting only the existing d without the divided private key. The corresponding contents works in ECIES_KEY_AGGREMENT2 in Figure 7. And all this process satisfies the equation in Figure 8. FUNC2 operates on the GPU as shown in Figures 5 and 7.

7.1.3. Stage 3. In stage 3, the ephemeral shared key Z value, which is the result of stage 2, is processed using the key derivation function HMAC-SHA256 to obtain the encryption and MAC keys. Then, the correct SUCI value is obtained using the AES-128-CTR decryption algorithm and verifying the MAC function. Unlike the other stages, the third stage is not a public-key cryptography operation but a symmetric key block cryptography operation. This stage has a simpler operation process than the above two processes; hence, its speed is rapid. Consequently, the operation can be performed on the CPU or GPU. If the third stage is configured using a GPU, in this case, the values must be derived for each n number of UEs, and GPU source code must be divided considering the entire key derivation and symmetric key block cipher. The last step can also be confirmed with FUNC3 in Figure 9.

When configuring the proposed 5G SIDF, the biggest difficulty in the GPU configuration is the GPU's operation method, as mentioned in Background. Kernel functions operate in warp units in all SMs within the GPU in the GPU. This is the biggest advantage and characteristic when implementing a cryptographic algorithm using GPU, but it is also the biggest disadvantage. It is important that as many warps as possible are simultaneously performing work on all SMs that are components of the GPU. When an ECIES

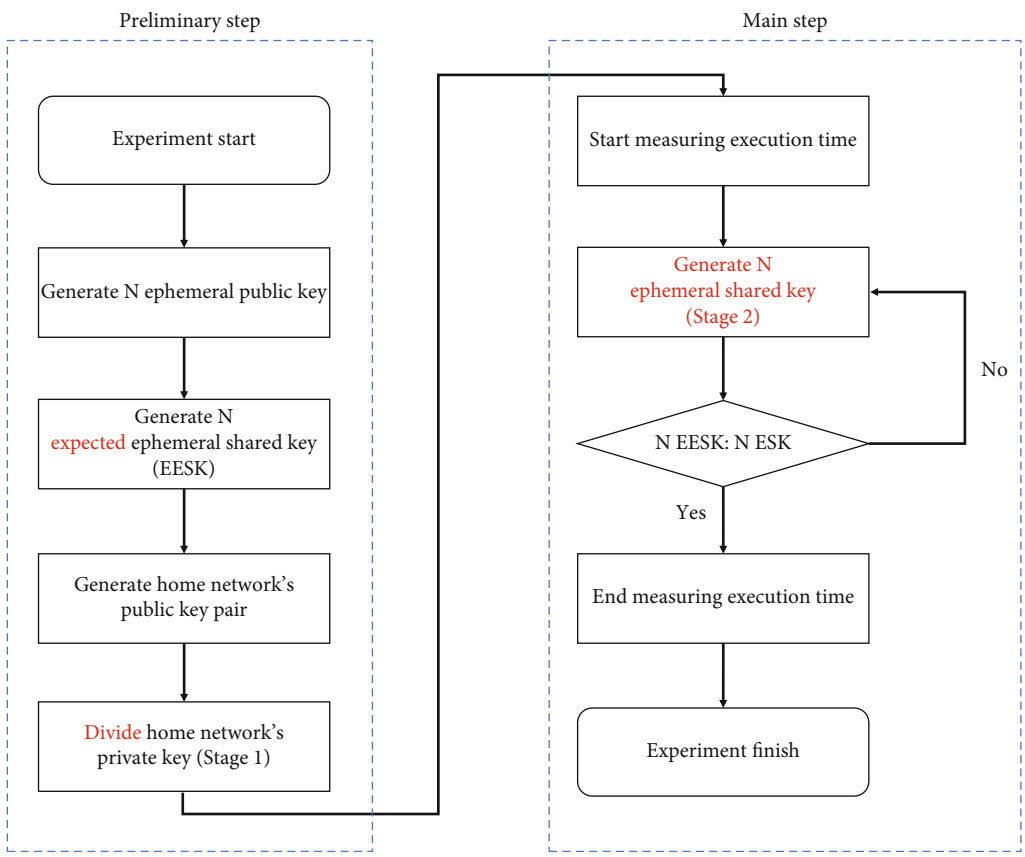


Figure 10: Experimental flowchart: the experiment was configured like the corresponding flowchart. The experiment is largely divided into a preliminary step and a main step. In the preliminary step, all the preparation values necessary for the experiment are generated. Therefore, EESK is generated, and the process of dividing the private key of the home network is performed. In the main step, N ESKs are generated, compared with the EESKs, and the total time is measured.

decryption request does not come with a warp multiple size, the kernel function is terminated first in a specific SM, and the final end time becomes the end time when the warp multiple size is the size. Therefore, it is necessary to consider how to configure the SIDF when a request is made rather than taking a multiple of the number of warps. The configuration method presented in this paper performs the ECIES decryption process without filling the number of warps even when a request is received, which is not a multiple of the number of warps. The second difficulty is selecting GPU equipment and implementing GPU kernel functions. GPUs are generally graphics devices and cannot be used on their own. Therefore, in the 5G network, when AUSF or SIDF receives an ECIES decryption command, there must be a host PC that can give a command to the GPU again. Also, as mentioned above, NVIDIA GPU was used in this paper. However, the CUDA library cannot be used when using an AMD GPU. Therefore, it has the disadvantage of not using all GPUs. However, the CUDA library was only available in C and C++ as the initial development languages, and of course, the C language is optimized for implementation. However, recently, by accommodating various languages such as go, python, and Fortran, network design and interoperability with various open-source projects are good. Therefore, if you use NVIDIA's GPU, there is no difficulty using the GPU in the 5G network.

Table 1: Device A target platform configuration.

Device A	Contents
CPU	Intel Core i7-9700
Clock	3.00G Hz (800~4700 MHz)
Core/thread	8 core/8 thread
RAM	DDR4 16 GB X 2 (32 GB)
OS	Ubuntu 18.04.3 LTS
Compiler	gcc 7.4.0

Table 2: Device B target platform configuration.

Device B	Contents
GPU	NVIDIA GeForce GTX 1060
Architecture	Pascal
CUDA driver version	11.3
CUDA capability	6.1
CUDA cores	1280
Warp size	32

8. Experiment

8.1. Experimental Setup. By implementing the SIDF structure proposed in Chapter 3, we present the performance values of the proposed method. The experiment in this study is performed only for the X25519 key agreement process using Curve25519, which denotes stage 2, 5G ECIES profile A, except for the key derivation function and decryption process of the symmetric key block cipher algorithm. The reason for experimenting only with stage 2 is that it is based on public-key cryptography operation, while stage 3 is based on symmetric key block cipher operation. Symmetric key block cipher has a simpler operation compared to public-key cryptography. Thus, the operation speed is exceedingly fast compared to public-key cryptography. However, many studies have been published on this subject. Therefore, separate experiments were not conducted in this paper [31–33]. In addition, the reason for testing only profile A and not B is that the key agreement process using profile A has a faster basic operation speed and uses less memory compared to the key agreement process using the Secp256r1 curve, which denotes profile B [34–36]. Therefore, profile A is likely used when many mobile carriers implement ECIES as an actual key exchange algorithm without using the NULL-scheme. The cryptography algorithm used in the experiment is a modified version of OpenSSL V1.1.1g [37]. There are various implementation methods for the basic operation of Curve25519, which may vary in speed, source code size, and memory usage depending on the implementation method [34, 38–40]. Therefore, to achieve an unbiased experiment, the Curve 25519 source code in OpenSSL, which is used worldwide, was used as the basic source code.

The experimental process is carried out according to the procedure described in Figure 10. In the preliminary experimental stage, basic information regarding all experiments is generated. Particularly, all values used as input parameters are generated in the ECIES decryption process, which is the main stage. When the UE accesses the 5G home network, the UE ephemeral public key is generated along with SUCI. Simultaneously, (d, Q), a public key pair used by the home network to perform public-key operations is generated. To generate the expected ephemeral shared key, which is the result of the main step, the ECIES encryption step is performed for each terminal. In addition, the precomputation, which divides the home private key according to the number of threads, is also carried out in the corresponding stage. After the preliminary phase, the main experimental phase proceeds.

During the main experiment, as the number of UEs attempting to access the home network increases, the home network load is examined and the effectiveness of the proposed SIDF configuration that uses ECIES decoding calculation for each experimental device is tested. As mentioned above, the number of UE terminals in this experiment, that is, the number of ECIES decryption calculations, is set to 2^{10}, 2^{15}, 2^{17}, 2^{19}, and 2^{20} for each iteration. Each device receives d and Q generated in stage 1, the number of UEs attempting to access the home network, and the ephemeral public key of each UE. Each device calculates the ephemeral shared key through the ECIES decryption using only the received value and compares it to the expected ephemeral shared key generated in stage 1. All calculation times are measured if all values match when UE is compared. In the process described above, Device A sequentially calculates up to ephemeral shared keys, and device B calculates up to ephemeral shared keys through GPU parallelization based on the calculations performed in the precomputation process. The experiment conducted using device A is called test A, and the experiments conducted using devices B and C are called tests B and C, respectively.

Table 3: Device C target platform configuration.

Device C	Contents
GPU	TITAN Xp
Architecture	Pascal
CUDA driver version	10.2
CUDA capability	6.1
CUDA cores	3840
Warp size	32

Experiments were performed using the three devices specified in Tables 1–3. The first device (device A) is configured using a general CPU. The second and third equipment (devices B and C) are GPU configured for testing the proposed implementation method and SIDF. The resources of devices B and C only differ in the global memory size and CUDA core, while the rest of the GPU resources remain mostly the same. The experimental method is stated as follows: first, perform the experimental preliminary steps on the host PC (device A).

8.2. Experiment Results. The superiority of the proposed SIDF configuration is demonstrated by comparing the results of the abovementioned experiments. In the test using the three devices, the generation times for 2^{10}, 2^{15}, 2^{17}, 2^{19} and 2^{20} ephemeral shared keys were measured, respectively. Moreover, because the experimental results measure the time needed to generate ephemeral shared keys, each generation time is different depending on the length or value of the input parameter. Therefore, to increase the measurement accuracy, each test was performed twenty times, where each generation time was measured in microseconds (ms). The values reported in Tables 4–6 are the time it took to generate each temporary shared key in each environment. Particularly, in the experiment where each ephemeral shared key is generated through a GPU parallelization, the time taken to move the memory from the host device to the GPU device was measured. In test A, generating 2^{10} ephemeral shared keys took approximately 79.00 ms, 2^{17} ephemeral shared keys took approximately 10108.57 ms, and 2^{20} ephemeral shared keys took approximately 80879.87 ms. In other words, it took approximately 81 sec for the SIDF to perform ECIES decoding when roughly 1,000,000 UEs attempted to access it simultaneously. However, when a GPU is used, the SIDF processing speed increases dramatically. In test B, it took approximately 8.43, 227.96, and 1788.13 ms to generate 2^{10}, $2^{17}2^{17}$ and 2^{20} ephemeral shared keys, respectively.

Table 4: Results of test A (ms).

Iterate count	Number of ephemeral shared keys				
	2^{10}	2^{15}	2^{17}	2^{19}	2^{20}
1	78.97	2527.04	10109.66	40437.55	80880.75
2	79.03	2527.30	10110.96	40438.98	80877.55
3	78.98	2526.82	10106.96	40438.34	80877.09
4	78.98	2527.23	10107.18	40438.65	80880.36
5	78.96	2527.45	10107.79	40438.48	80877.43
6	78.94	2527.65	10107.00	40438.71	80878.13
7	78.95	2528.36	10107.42	40438.26	80880.73
8	79.12	2527.93	10106.78	40438.94	80883.68
9	78.98	2526.46	10106.68	40438.82	80877.36
10	78.98	2527.58	10106.36	40438.34	80882.59
11	78.96	2526.97	10106.27	40438.11	80879.48
12	78.96	2527.40	10110.58	40438.41	80876.11
13	79.00	2527.92	10110.07	40438.67	80877.60
14	78.95	2527.51	10110.87	40438.89	80880.26
15	79.10	2528.27	10110.67	40438.42	80882.32
16	78.96	2527.94	10110.72	40438.65	80883.53
17	79.01	2527.61	10107.49	40438.74	80880.35
18	78.96	2526.44	10108.16	40438.35	80879.56
19	79.13	2527.76	10111.22	40438.42	80880.52
20	79.15	2526.37	10108.59	40438.61	80881.88
Average	79.00	2527.40	10108.57	40438.52	80879.86
1 device consumed	0.07715	0.07713	0.07712	0.07713	0.07713

In test C, it took approximately 10.84, 66.32, and 523.03 ms to generate 2^{10}, 2^{17}, and 2^{20} ephemeral shared keys, respectively. These results indicate that when SIDF is configured using a GPU, the time taken for SIDF to perform ECIES decryption operation is 1.8 and 0.5 sec, respectively, when roughly 1,000,000 UEs attempt simultaneous access. Figure 11 shows the average decryption speed and average operation time of the corresponding result as a graph.

8.3. Analysis of Results. Several analyses can be performed on the experimental results. First, observe that test A, an experiment that does not perform parallel processing using a CPU, requires 0.07 ms to perform one ECIES decryption operation. However, in tests B and C that perform parallel programming using a GPU, the execution time of one ECIES decryption operation was linearly reduced to that of ephemeral shared keys. After ephemeral shared keys, the operation time was constant at approximately 0.0017 ms and 0.0005 ms. Second, based on the minimum consumption time, notice that the parallelized tests B and C using a GPU are at least 43 and up to 150 times faster than test A, a nonparallelized test using a single CPU. Particularly, when device C is used with numerous CUDA cores, when configuring one SIDF per UE or more, the SIDF configuration efficiency using a GPU can improve by 150 times compared to that achieved when configuring using a single CPU. Third, if SIDF is configured through a GPU with device B specifications, up to 2^{19} UEs can be processed within 1 second. As mentioned earlier, this is crucial when configuring the mMTC 5G network.

The mMTC 5G network requirement states that the network should be configured to provide smooth access when 1,000,000 UEs per square kilometer access the network. Notice that our results satisfy the mMTC 5G network requirement. Therefore, even if it is not a URLLC 5G network requirement, our proposed approach can satisfy the mMTC requirement if a GPU is used. Test B and test C were conducted using the same CUDA source code. Analysing only the results of the two experiments, test C's operating time in ECIES decryption is 3 times faster than test B's operating time. At this time, the number of CUDA cores of device C used at this time is 3 times larger than the number of CUDA cores of device B used in test B. Therefore, it can be seen that the tests using GPU have a speed difference proportional to the number of CUDA cores. Device B and device C used in this experiment are not the latest graphics cards. GPUs released in 2021 have 2.6 times more CUDA cores than device C and are significantly ahead of memory clocks. Therefore, it is expected that better results can be obtained than the results of this experiment when the experiment is performed using the latest GPU. Therefore, if GPU having a specification of device B or higher is used alone or

Table 5: Results of test B (ms).

Iterate count	Number of ephemeral shared keys				
	2^{10}	2^{15}	2^{17}	2^{19}	2^{20}
1	8.46	62.27	229.19	891.86	1756.89
2	8.19	61.71	227.29	893.98	1779.35
3	8.58	61.87	228.89	884.21	1781.72
4	8.23	62.66	225.74	890.49	1778.26
5	9.34	62.17	226.25	886.25	1797.18
6	8.24	61.56	230.06	887.2	1777.28
7	8.17	61.53	227.7	898.44	1781.34
8	8.41	62.4	227.06	885.77	1783.97
9	8.32	61.92	231.07	889.06	1778.21
10	9.04	63.61	227.98	885.33	1783.23
11	8.42	62.14	225.56	898.17	1817.36
12	8.23	62.19	224.99	893.3	1790.2
13	8.4	61.37	226.87	882.89	1806.54
14	8.15	61.46	228.26	890.03	1797.49
15	8.54	61.7	228.1	888.63	1780.03
16	8.15	61.57	227.79	882.48	1797.53
17	8.18	61.92	230.64	897.27	1798.12
18	8.37	62.62	228.52	890.18	1799.07
19	8.23	62.16	228.67	890.06	1777.16
20	9.11	61.71	228.67	892.14	1801.67
Average	8.438	62.02	227.965	889.887	1788.13
1 device consumed	0.00824	0.001893	0.001739	0.001697	0.001705

multiple GPU is used simultaneously using a scalable link interface technique, 5G SIDF function that satisfies the mMTC requirement is configured.

Lastly, we would like to analyze the warp unit operation, which is a disadvantage of using GPU. In the case of a small request of 2^{10} or less, the difference between device A and device B is approximately 10 times. In the case of 1 warp unit, about 2.4 seconds is consumed by the CPU, and if this is calculated again, it can be calculated that device A consumes about 0.07 seconds when processing a single request. Similarly, in the case of device B, about 0.26 seconds is consumed by the use of GPUs in the 5G home network. Previously, the disadvantage of GPU when the calculation is not performed in warp units is that even if the calculation is terminated, the memory and computing device cannot be used until all warps are finished. However, looking at the above results, it can be said that it is faster to use the GPU unconditionally when the number of decryption requests is divided into warp units, and decryption is performed sequentially when the last remaining number is 4 or more. In other words, it is concluded that it is faster to use a general CPU only when there are a maximum of 3 decryption requests, and it is better to use a GPU in all other cases. In particular, this paper is characterized by the use of GPUs in the 5G home network. Previously, in [18–22], GPU was used to quickly operate only ECC cryptographic operations. However, in this paper, the 5G SIDF configuration is divided into three stages, and the GPU is used in all processes. In particular, in stage 1, HN's private key is divided into warp units so that the ECIES key matching process can be performed quickly in the subsequent stage. In addition, in stage 3 of decrypting the actual SUCI into SUPI, a general and effective method using a GPU device in cryptography was used. Therefore, the most important point in this paper is that the GPU is used for all operations performed by SIDF.

In addition, as suggested in this paper, when cryptographic algorithms and cryptographic systems are newly installed in the system, they should be evaluated through various evaluation methods by comparing operation time, power consumption, flexibility, financial cost, etc., with other commercial equipment. This is because these evaluation methods are items to consider when an encryption algorithm or system is operated in actual equipment. Firstly, I would like to compare it with other 5G commercial products in terms of power consumption. Power consumption of cryptographic algorithms becomes particularly important in IoT equipment, embedded equipment, and 5G UE that are currently used in various environments. IoT equipment, embedded equipment, 5G UE, etc., are equipment in which equipment is operated through batteries, not in an environment in which power is generally supplied at all times. Therefore, power consumption when a cryptographic algorithm or cryptographic system operation is added to the device is a very critical issue.

Table 6: Results of test C (ms).

Iterate count	Number of ephemeral shared keys				
	2^{10}	2^{15}	2^{17}	2^{19}	2^{20}
1	11.27	19.69	66.1	262.72	518.48
2	11.24	19.12	66.58	264.77	518.96
3	11.24	18.37	66.71	262.61	521.15
4	11.24	18.67	66.2	263.47	521.29
5	11.24	18.30	66.05	264.77	518.90
6	11.24	18.41	66.09	263.54	525.26
7	11.24	18.13	65.81	264.13	524.52
8	11.24	18.28	65.69	263.81	521.95
9	11.24	18.58	66.65	263.82	523.49
10	11.24	18.15	66.32	263.42	525.21
11	11.24	18.15	65.67	263.82	522.53
12	11.24	18.21	66.04	264.18	522.01
13	10.91	18.60	66.4	263.66	525.47
14	10.13	18.50	65.93	262.78	525.26
15	10.13	18.32	65.77	262.49	523.02
16	10.12	18.81	66.87	264.30	522.45
17	10.13	18.35	67.12	262.10	523.5
18	10.13	18.60	67.26	262.07	523.62
19	10.13	18.28	66.67	260.84	525.54
20	10.13	18.35	66.56	262.27	528.03
Average	10.84	18.49	66.32	263.28	523.03
1 device consumed	0.01058	0.00056	0.00050	0.00050	0.000499

However, the 5G core network environment that this paper focuses on does not require low-power computation and low-power cryptographic algorithms. 5G core network is an environment that requires more high security level (depending cryptographic algorithm key length) and compatibility of cryptographic algorithms than the advantages obtained by using low-power computation and low-power cryptographic algorithms [41]. In addition, the maximum power consumption of device B and device C used in the SIDF configuration method using GPU presented in this paper is about 260 watts [42, 43]. This value is not an absolute comparison because it is the maximum consumption of the GPU device, not the power consumption of the cryptographic algorithm, but it does not show a big difference when compared with the maximum power consumption of 5G equipment.

The second is a comparison of operation time compared to 5G commercial security product solutions. Most 5G network devices collaborate with the world's leading cryptographic equipment manufacturers such as Thales and IDQ to provide security and encryption functions. If we look at the cryptographic devices in these network devices, they focus on many security functions. The cryptographic function used in Ericssons's CCSM is provided through Thales's 5G Luna Hardware Security Module (HSM). All crypto operations and storing, generating, and managing of encryption keys are performed within the secure confines of the 5G Luna HSM FIPS 140-2 level 3 and Common Criteria EAL 4+, while ensuring the protection of subscriber identities, including the Subscription Concealed Identifier (SUPI), user equipment, radio area networks (RANs), and their core network infrastructure. 5G Luna HSM offers up to 1,660 transactions per second (tps) for profile A Decrypt 25519 with a single HSM. In case of High Availability Cluster 2 5G Luna HSMs rather than single HSM, it supports up to 3,440 tps. The 5G Luna HSM offers high assurance key protection and up to 6,070 tps for profile B Decrypt P-256 and 1,660 tps for profile A Decrypt 25519 to meet security and throughput and scalability requirements for 5G [44]. Tables 7 shows the comparison values between test C and Thales 5G Luna HSM tested in this paper.

Also, unlike Ericsson, Nokia's SDM solution does not mention the cryptographic algorithm. However, to configure a key distribution system within the 5G network through collaboration with SKT and IDQ, various security functions such as quantum key distribution (QKD) technology have been added to build a 5G network security system. However, we could not find specific information about 5G primary authentication like SUCI deconcealing mentioned in this paper. Even though the 5G core network is built through collaboration with each mobile operator and equipment manufacturer, the functions that can perform primary UE authentication when trying to access many devices at once in an mMTC environment are still lacking. According to the information disclosed by Ericsson, when ECIES profile A is selected, the performance is 1,660 tps. Of course, since the numerical values

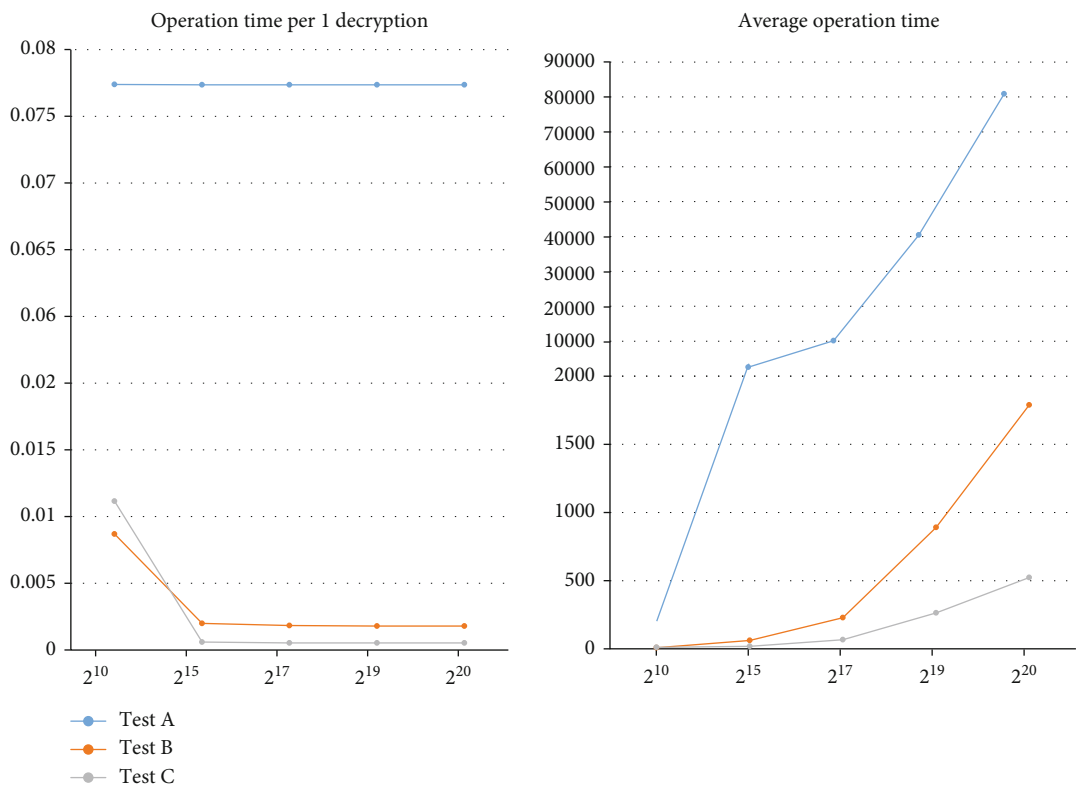


Figure 11: Operation time: each graph means the time required for each number of operations. The graph on the left means one decryption time consumed in each experiment. The graph on the right means the total decryption time consumed in each experiment.

Table 7: Comparison with 5G commercial product.

	Test C	Thales 5G Luna HSM
Power consumption	260 watts (GPU max power consumption)	84~100 watts
Operation time per 1,000,000 UE processing (profile A)	0.523 seconds	290 seconds (with clustering HSM)
Financial cost	Under $1,500	Expensive
Features	—	FIPS 140-2 and CC EAL4+

presented by the equipment are measured based on network transactions, it is difficult to directly compare them with this thesis, which only performs ECIES decryption calculations. However, considering the requirement to cover 1,000,000 devices per square kilometer, which is the standard of mMTC, it can be expected that it will consume about 631 seconds. However, when the method presented in this paper is performed on device C, the ECIES decryption processing time for the same 1,000,000 ECIES is about 0.523 seconds, which is approximately 1200 times faster. The third is financial cost comparison. In order to satisfy the mMTC standard through the aforementioned commercial products, AUSF/SIDF can be configured through multiple devices instead of using one device. However, as suggested in this paper, when a GPU is used, the figure is up to 1000 times faster, and the network configuration is much cheaper in terms of price, so I think the price competitiveness is better than using the existing solution.

9. Conclusion

In 5G telecommunication networks, SUCI is created and used with an ECIES scheme to prevent IMSI information for user identification from being exposed during the initial access of a mobile communication terminal. However, because the ECIES scheme includes a public-key operation for key sharing, significant overhead may occur in the 5G network when many terminals attempt simultaneous access, such as the mMTC service for IoT environment, owing to a large amount of required computation. We judged that it is insufficient to support the current mMTC service for IoT environment through surveys on 5G products and several 5G network open-source surveys. Therefore, in this study, to solve this problem, a parallelization technique that uses a GPU is proposed to configure the SIDF responsible for SUCI deconcealing. In our experiments, a minimal source modification was applied to operate the OpenSSL source code on the GPU, and no separate optimization was

performed. Nevertheless, as the number of device access requests increases, the key sharing time per device when using a GPU is up to 150 times faster than that achieved with a CPU. In addition, the method presented in this paper showed superiority through a comparison of power consumption and operation time with 5G products. Therefore, the experiment presented here proves that the parallel implementation method using GPUs for SIDF configuration can be a countermeasure against the overhead that occurs when numerous devices request access to multiple terminals.

In current experiments, a method of dividing the home network private key used for ECIES according to threads was used. Because GPU resource optimization was not applied separately here, we will experiment with an optimization implementation suitable for the GPU environment in future research. In addition, because this study only tested profile A, a study on the parallelization of profile B needs to be conducted. We plan to conduct research applying the experimental results obtained here to 5G open-source projects.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the Republic of Korea's MSIT (Ministry of Science and ICT), under the High-Potential Individuals Global Training Program (2021-0-01516) supervised by the IITP (Institute of Information and Communications Technology Planning & Evaluation).

References

[1] Series M, *IMT Vision–Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*, ITR-U, 2015.

[2] S. P. Rao, S. Holtmanns, I. Oliver, and T. Aura, "Unblocking stolen mobile devices using SS7-MAP vulnerabilities: exploiting the relationship between IMEI and IMSI for EIR access," in *2015 IEEE Trustcom/BigDataSE/ISPA*, pp. 1171–1176, Helsinki, Finland, 2015.

[3] M. Khan, A. Ahmed, and A. R. Cheema, "Vulnerabilities of UMTS access domain security architecture," in *2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pp. 350–355, Phuket, Thailand, 2008.

[4] 3rd Generation Partnership Project (3GPP), *Technical Specification Group Services and System Aspects; TS 33.501: Security Architecture and Procedures for 5G System (Release 16)*, 3GPP, 2020.

[5] 3rd Generation Partnership Project (3GPP), *Technical Specification Group Services and System Aspects; TS 23.503: Policy and Charging Control Framework for the 5G System (5GS); Stage 2 (Release 16)*, 3GPP, 2020.

[6] 3rd Generation Partnership Project (3GPP), *Technical Specification Group Core Network and Terminals; TS 23.003: Numbering, Addressing and Identification; (Release 16)*, 3GPP, 2020.

[7] SECG SEC 1, *Recommended Elliptic Curve Cryptography, Version 2.0*, 2009, http://www.secg.org/sec1-v2.pdf.

[8] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[9] Miller VS, "Uses of elliptic curves in cryptography," in *Advances in Cryptography-Crypto'85*, vol. 218, pp. 417–426, Springer, 1986.

[10] IETF RFC 7748, *Elliptic Curves for Security*, 2016, https://tools.ietf.org/html/rfc7748.

[11] SECG SEC 2, *Recommended Elliptic Curve Domain Parameters, Version 2.0*, 2010, https://www.secg.org/sec2-v2.pdf.

[12] NVIDIA, "Developer forum," https://forums.developer.nvidia.com/.

[13] NVIDIA, "CUDA C++ programming guide 11.3; see," 2007, https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html.

[14] Wouter de Groot, *A Performance Study of X25519 on Cortex-M3 and M4, [Ph.D. Thesis]*, Eindhoven University of Technology, 2015.

[15] M. Rivain, "Fast and regular algorithms for scalar multiplication over elliptic curves," 2011, IACR Cryptology ePrint Archive 2011/338, http://eprint.iacr.org/2011/338.pdf.

[16] T. Kudithi and R. Sakthivel, "High-performance ECC processor architecture design for IoT security applications," *The Journal of Supercomputing*, vol. 75, no. 1, pp. 447–474, 2019.

[17] M. Imran, I. Shafi, A. R. Jafri, and M. Rashid, "Hardware design and implementation of ECC based crypto processor for low-area-applications on FPGA," in *2017 International Conference on Open Source Systems & Technologies (ICOSST)*, pp. 54–59, Lahore, Pakistan, 2017.

[18] R. Szerwinski and T. Güneysu, "Exploiting the Power of GPUs for Asymmetric Cryptography," in *Cryptographic Hardware and Embedded Systems – CHES 2008*, Springer, 2008.

[19] D. J. Bernstein, T. R. Chen, C. M. Cheng, T. Lange, and B. Y. Yang, "ECM on graphics cards," in *Advances in Cryptology - EUROCRYPT 2009*, vol. 5479, pp. 483–501, 2000.

[20] S. C. Seo, T. H. Kim, and S. K. Hong, "Accelerating elliptic curve scalar multiplication over GF(2m) on graphic hardware," *Journal of Parallel Distributed Computing*, vol. 75, pp. 152–167, 2015.

[21] J. Dong, F. Zheng, J. Cheng, J. Lin, W. Pan, and Z. Wang, "Towards high-performance X25519/448 key agreement in general purpose GPUs," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, pp. 1–9, Beijing, China, May 2018.

[22] L. Gao, F. Zheng, N. Emmart, J. Dong, J. Lin, and C. C. Weems, "DPF-ECC: accelerating elliptic curve cryptography with floating-point computing power of gpus," in *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 494–504, New Orleans LA USA, May 2020.

[23] Ericsson, *Cloud Native Subscription and Data Management in 5G – A Guide to Mastering Data and Subscriber's Handling in Multi-Access Core Network*, Technical Paper, 2021.

[24] NOKIA, *Subscriber Data Management - Evolving SDM for 5G, Cloud and Future Networks*, Whitepaper, 2021.

[25] "The open source for 5G," https://open5gs.org.

[26] "The open source for 5G," https://www.free5gc.org/.

[27] "The open source for 5G," https://pkg.go.dev/golang.org/x/crypto/curve25519.

[28] "The open source for 5G," https://openairinterface.org/.

[29] M. Brown, D. Hankerson, J. López, and A. Menezes, "Software implementation of the NIST elliptic curves over prime fields," in *Topics in Cryptology — CT-RSA 2001*, Springer, Berlin, Heidelberg, 2001.

[30] H. Tschofenig and M. Pegourie-Gonnard, "Performance investigations," 2015, http://www.ietf.org/proceedings/92/slides/slides-92-lwig-3.pptx.

[31] A. D. Biagio, A. Barenghi, G. Agosta, and G. Pelosi, "Design of a parallel AES for graphics hardware using the CUDA framework," in *Proc. of 2009 IEEE International Parallel and Distributed Processing Symposium*, pp. 1–8, Rome, Italy, 2009.

[32] S. A. Manavski, "CUDA compatible GPU as an efficient hardware accelerator for AES cryptography," in *2007 IEEE International Conference on Signal Processing and Communications*, pp. 65–68, Dubai, United Arab Emirates, 2007.

[33] K. Iwai, N. Nishikawa, and T. Kurokawa, "Acceleration of AES encryption on CUDA GPU," *International Journal of Networking and Computing*, vol. 2, no. 1, pp. 131–145, 2012.

[34] M. Düll, B. Haase, G. Hinterwälder et al., "High-speed curve25519 on 8-bit 16-bit and 32-bit microcontrollers," *Designs, Codes and Cryptography*, vol. 77, no. 2-3, pp. 493–514, 2015.

[35] E. Mahe and J.-M. Chauvet, "Fast GPGPU-based elliptic curve scalar multiplication," *IACR Cryptology ePrint Archive*, vol. 2014, 2014.

[36] C. Huth, R. Guillaume, P. Duplys, K. Velmurugan, and T. Güneysu, "On the energy cost of channel based key agreement," in *Proceedings of the 6th International Workshop on Trustworthy Embedded Devices*, pp. 31–41, Vienna Austria, 2016.

[37] "The open source toolkit for SSL/TLS, OpenSSL-1.1.1.g," https://www.openssl.org/.

[38] T. Chou, "Sandy2x: "new Curve25519 speed records"," in *Selected Areas in Cryptography – SAC 2015*, O. Dunkelman and L. Keliher, Eds., vol. 9566, pp. 145–160, Springer, Cham, 2016.

[39] H. Fujii and D. F. Aranha, "Curve25519 for the Cortex-M4 and beyond," in *Progress in Cryptology – LATINCRYPT 2017*, pp. 36-37, Springer, 2017.

[40] E. Käsper, "Fast elliptic curve cryptography in OpenSSL," in *Financial Cryptography and Data Security*, vol. 7126, pp. 27–39, Springer, 2011.

[41] N. Mouha, "The design space of lightweight cryptography," 2015, https://hal.inria.fr/hal-01241013.

[42] NVIDIA, "Product specification," https://www.nvidia.com/en-in/geforce/products/10series/geforce-gtx-1060/.

[43] NVIDIA, "Product specification," https://www.nvidia.com/en-us/titan/titan-xp/.

[44] Thales, "Thales 5G Luna network HSM," 2021, https://cpl.thalesgroup.com/resources/encryption/5g-luna-network-hsm-data-sheet.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9058048, 7 pages
https://doi.org/10.1155/2022/9058048

Research Article

Learning the Correlations between IoT Systems Consisting of Massive Sensors

Shuze Jia, You Ma, Juan Xue, and Aijun Zhu

National Satellite Meteorological Center, Beijing 100081, China

Correspondence should be addressed to You Ma; mayou0531@126.com

Received 31 March 2022; Accepted 7 May 2022; Published 24 May 2022

Academic Editor: Yan Huo

An IoT system often consists of many sensors to collect data in different aspects. Meanwhile, all these sensors describe the IoT system's functional status, to which it belongs. The correlations between subsystems are always emphasized for a complex system that contains several IoT subsystems. At the same time, there are still no good ways to calculate these types of correlations since that (1) multiple sensors describe an IoT system as a matrix while the correlation between matrices cannot be calculated by the traditional methods (i.e., vector ways such as Pearson correlation coefficient) and (2) AI methods such as neural networks were introduced to resolve this problem; however, these black-box approaches cannot explain the mathematical mechanisms, and lots of memory or time are consumed. This paper proposed a novel approach named the matrix-oriented correlation computing method (MOCC) to learn the correlations between IoT systems. The critical problem of this proposed method is calculating the correlation between two curved surfaces, which are modeled as matrices, since an IoT system often contains many sensors which characterize different aspects of this system and continuously generate data in time series. By our MOCC method, the correlation or interaction between any two subsystems can be accurately measured, which means that we can predict the state of a system by its most important related system. Missing data value prediction based on our MOCC method is also presented in this paper. We verified the efficiency and effect of our proposed method via a satellite, a typical IoT system consisting of massive sensors, and the experimental result was proved to outperform existing methods.

1. Introduction

A complex system often contains several IoT subsystems. For example, a satellite platform consists of at least energy, attitude, propulsion, thermal control, data transmission, and payload six subsystems; and each subsystem is designed as the integration of smaller subsystems. The correlation between two subsystems is always an essential consideration in the maintenance or analysis of a complex IoT system [1]. By the correlation analysis, we can determine the most relevant factor of an IoT subsystem. For example, we can find that a system's data anomaly will affect other systems' functional status if there are high correlations between them [2]; or a system's data missing can be predicted based on its high relevant systems [3–7]. Generally, this type of correlation can only be calculated by the sensor data of IoT systems since the sensors are designed for monitoring data for all different properties. If a subsystem consists of M sensors that have collected data in a time series of length N, then the data would be modeled as an $M \times N$ matrix, and the correlation between two systems is actually the correlation between two matrices.

There are many existing correlation computing methods, such as Pearson correlation coefficient (PCC) [8] and cosine coefficient (COS) [9], and new correlation measuring methods have been presented in recent years [10–12]. However, most of the existing methods can only measure the correlation between one-dimensional vectors but not matrices. Figure 1 compares the vector and matrix oriented similarity measurement.

AI methods such as neural networks were introduced to resolve this problem in recent years [13]; however, these black-box approaches cannot explain the mathematical mechanisms, and lots of memory or time are consumed.

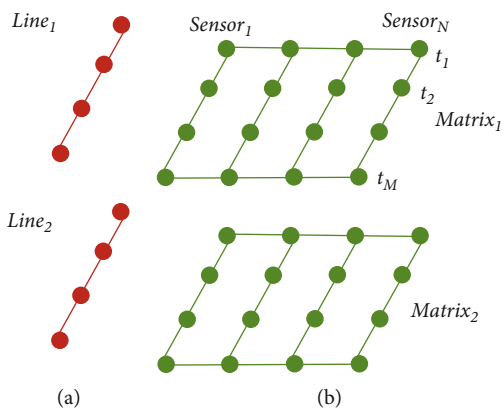


Figure 1: (a) shows that the traditional method can only measure the correlation between two vectors, meaning that each vector consists of only one sensor's data, while (b) shows two matrices, each of them consisting of several sensors.

This paper proposed a novel approach named the matrix oriented correlation computing method (MOCC) to learn the correlations between IoT systems, and this method can also be used for any multidimensional observation objects. After the M sensors of an IoT system have produced data in a time series of length N, we can get an $M \times N$ matrix. Since this matrix is constructed along with time, the MOCC method should measure the correlation between two matrices considering the time factor.

The rest of this paper is organized as follows. Section 2 presents our MOCC method. Section 3 describes a critical application—missing data prediction of our MOCC method. Section 4 describes our experiments. Section 5 concludes the paper, and Section 6 shows related work.

2. Proposed MOCC Method

For ease of explanation, we use the following matrix shown in Table 1 to describe the data produced by M sensors in a length N time series.

This section will illustrate the mathematical steps of our MOCC method to compute the correlation between any two matrices of this type. MOCC is inspired by a significant and novel math concept identified as distance correlation. The concept of distance correlation extends the correlation calculation from one-dimensional space to two-dimensional space, but it does not consider the time factor. This section presents the mathematical principles and advantages of distance correlation and then improves it by bringing the time factor into it, which eventually resulted in our MOCC method.

2.1. Distance Correlation

2.1.1. Mathematical Principles. We use the distance correlation concept to measure correlation considering multiple sensors of an IoT system integrally. Distance correlation—a statistics and probability theory-based concept—was proposed by Szekely et al. to measure the statistical dependence between two random vectors of arbitrary, not necessarily equal dimension [14, 15]. Therefore, distance correlation can measure the correlation between any two matrices more accurately and comprehensively. The distance correlation is defined as follows.

Table 1: Data of an IoT system.

	T_1	T_2	…	T_N
Sensor_1	v_{11}	v_{12}	…	v_{1N}
Sensor_2	v_{21}	v_{22}	…	v_{2N}
…	…	…	…	…
Sensor_M	v_{M1}	v_{M2}	…	v_{MN}

Let X and Y denote two different IoT systems consisting of p and q sensors, respectively. If these two systems continuously produce data in a time series of length N, then, we can get an observed random sample from the joint distribution of random vectors X in $\mathbb{R}^p$ and Y in $\mathbb{R}^q$ as follows:

$$(X, Y) = \{(X_n, Y_n): n = 1, 2, \cdots, N\}. \tag{1}$$

For example, if X is an IoT system shown in Table 1, then X_n is the n-th column of Table 1.

From the definition of X and Y, we can get that two systems that need to measure correlation need not have the same number of sensors but have to be observed in the same time series.

And define:

$$a_{kl} = \|X_k - X_l\|_p, \tag{2}$$

$$\|X\|_p = \left(\sum_{i=1}^{p} |x_i|^p\right)^{1/p}, \tag{3}$$

$$\bar{a}_{k\cdot} = \frac{1}{N}\sum_{l=1}^{N} a_{kl}, \tag{4}$$

$$\bar{a}_{\cdot l} = \frac{1}{N}\sum_{k=1}^{N} a_{kl}, \tag{5}$$

$$\bar{a}_{\cdot\cdot} = \frac{1}{N^2}\sum_{k,l=1}^{N} a_{kl}, \tag{6}$$

$$A_{kl} = a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot}. \tag{7}$$

Similarly,

$$b_{kl} = \|Y_k - Y_l\|_q, \tag{8}$$

$$\|Y\|_q = \left(\sum_{i=1}^{q} |y_i|^q\right)^{1/q}, \tag{9}$$

$$\bar{b}_{k\cdot} = \frac{1}{N}\sum_{l=1}^{N} b_{kl}, \tag{10}$$

$$\bar{b}_{.l} = \frac{1}{N}\sum_{k=1}^{N} b_{kl}, \tag{11}$$

$$\bar{b}_{..} = \frac{1}{N^2}\sum_{k,l=1}^{N} b_{kl}, \tag{12}$$

$$B_{kl} = b_{kl} - \bar{b}_{k.} - \bar{b}_{.l} + \bar{b}_{..}. \tag{13}$$

Before giving the distance correlation between X and Y, their distance variance is defined as follows.

$$v^2(X, Y) = \frac{1}{N^2}\sum_{k,l=1}^{N} A_{kl}B_{kl}. \tag{14}$$

Based on this, the distance correlation between X and Y is defined as follows.

$$d(X, Y) = \begin{cases} \dfrac{v(X, Y)}{\sqrt{v(X, X)v(Y, Y)}} & \text{if } v(X, X)v(Y, Y) \neq 0, \\ 0 & \text{else.} \end{cases} \tag{15}$$

Then, the distance correlation between X and Y is measured as $d(X, Y)$.

Some of the mathematical properties of distance correlation are

(1) $v(X, Y) \geq 0$;

(2) $0 \leq d(X, Y) \leq 1$;

(3) $d(X, Y) = 0$ if and only if X and Y are independent

(4) $d(X, Y) = 1$ implies that X and Y have the equal dimensionality and $Y = A + bCX$, wherein A is a vector, b is a real number, and C is an orthonormal matrix

2.1.2. Advantages of Distance Correlation. First, from the definition of distance correlation, we can get that its most important advantage is that it can measure correlation between multidimensional vectors.

Additionally, distance correlation can illustrate correlation between vectors more accurately than most of existing methods. We use an example shown in Figure 2 to compare the correlation between X and Y got by distance correlation method and the other two important methods—PCC and COS, respectively. Figure 2 presents two vectors, $X = \{x_n : n = 1, 2, \cdots, N\}$ wherein $x_i = N/2 - i$ if $i \leq N/2$ else $x_i = i - N/2$; and $Y = \{y_n : n = 1, 2, \cdots, N\}$ wherein $y_i = i - N$. We get from Figure 2 that the value of X allows for a nice estimation of the value of Y, vice versa. In another words, X and Y are very relevant to each other. However, the correlation between X and Y is 0 measured by PCC and 0.75 by COS. As the comparison, the correlation between X and Y is 1 measured by distance correlation.

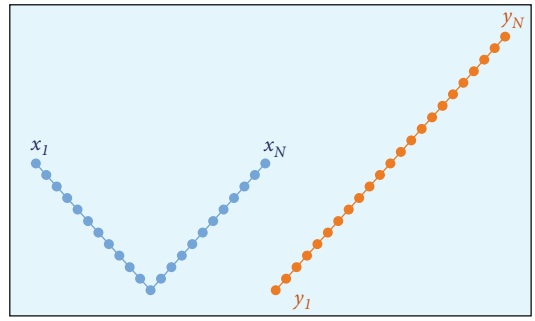


Figure 2: Two vectors: $X = \{x_n : n = 1, 2, \cdots, N\}$, $Y = \{y_n : n = 1, 2, \cdots, N\}$.

2.2. Enhance Distance Correlation by Time Factor. It is reasonable to consider that time can affect the correlation between two systems. It often means that the later the data is obtained, the more significant the impact on the correlation, vice versa. Therefore, this section combines time factor and distance correlation to measure correlations more accurately. The improved distance correlation is identified as MOCC correlation.

If two systems were observed in the same time series of length N, and let the 1st observation is the earliest, then, the N-th observation should have the largest weight on their correlation, and the $(N-1)$-th has the second largest. Therefore, if we bring time factor into distance correlation, equation (14) is revised as

$$v^2(X, Y) = \frac{1}{N^2}\sum_{k,l=1}^{N} (k + l)A_{kl}B_{kl}, \tag{16}$$

wherein $(k + l)$ is the time factor. The bigger k or l indicates the later time, therefore, has the bigger weight. Introduce $v^2(X, Y)$ of equation (16) into equation (15), the MOCC correlation will be got.

3. Applications of MOCC Correlation

A significant application of MOCC correlation is missing data prediction, which is also widely studied in the research field of IoT systems. MOCC correlation is a transform of the distance correlation. Although distance correlation is widely adopted by many researchers, the missing value prediction based on distance correlation has rarely been studied before.

An IoT system often consists of massive sensors, and data missing is a common phenomenon for system running. Data missing may be caused by network packet loss, or some sensors' transient exception. If it occurs, then some data in Table 1 will be missing, the prediction of missing data for an IoT system consists of the following two steps: (1) finding its high correlated systems based on history data and (2) predicting missing value based on its high correlated systems.

3.1. High Correlated System Finding. For ease of presentation, denote the set of all the IoT systems as $S = \{S_1, S_1, \cdots, S_N\}$, if one system has data missing, its high correlated systems have to be found for missing data prediction. Denote

S_a is the system that needs missing data prediction, the set of its all high correlated systems can be denoted as:

$$S' = \{S_k : d(S_a, S_k) \geq \text{THRESHOLD}, S_k \in S\}, \tag{17}$$

wherein $d(S_a, S_k)$ is the MOCC correlation between system S_a and S_k, THRESHOLD is a constant to indicate whether these two systems are high correlated or not. In practice, we let THRESHOLD = 0.8.

3.2. Missing Data Prediction. Before the missing data prediction, we should determine another essential issue: how a system is impacted by its high correlated systems. By experiments, we found two phenomena:

(1) If two systems have a high MOCC correlation, then, their MOCC correlation will hardly change with their data amount growing with time

(2) On the contrary, if two systems have only a low MOCC correlation, then, their MOCC correlation would change notably with their data amount growing

This experiment will be detailed and presented in Section 4, and this section only use the corollary of this experiment to predict missing data. From the above phenomena, we can get a corollary as follows.

Corollary 1. *If S_a is an IoT system with missing data and S_k is high correlated to S_a, then, the prediction of the missing data should keep $d(S_a, S_k)$ almost unchanged.*

Take the IoT system in Table 1 as an example, v_{mn} is the observed value of Sensor_m on Time_n, if this value is missing, denote the prediction of it as $\hat{v}_{mn}$.

Based on the observations of S_a and S_k from Time_1 to Time_{n-1}, we have got their MOCC correlation denoted as $d^{(n-1)}(S_a, S_k)$. If the observation on Time_n missed the value v_{mn}, then, the prediction $\hat{v}_{mn}$ should satisfy

$$d^{(n)}(S_a, S_k) \approx d^{(n-1)}(S_a, S_k), \tag{18}$$

wherein $d^{(n)}(S_a, S_k)$ is got by predicting the missing v_{mn} as $\hat{v}_{mn}$.

If we denote S_a as X, and its high correlate system S_k as Y, after the observations on Time_{n-1}, we rewrite equation (14) as

$$v^{(n-1)}(X, Y) = \sqrt{\frac{1}{(N-1)^2} \sum_{k,l=1}^{N-1} A_{kl} B_{kl}}. \tag{19}$$

By predicting the missing v_{mn} as $\hat{v}_{mn}$, namely, that we have completed the observations on Time_n, if the following satisfied

$$v^{(n-1)}(X, Y) \approx v^{(n)}(X, Y) = \sqrt{\frac{1}{N^2} \sum_{k,l=1}^{N} A_{kl} B_{kl}}, \tag{20}$$

then, equation (20) will be satisfied. Finally, we can get that the value that satisfied equation (22) is the needing prediction of v_{mn}. Although equation (22) seems complicated, it is just a multiple-order equation with only one variable $\hat{v}_{mn}$, we can solve it quickly with the facility of three-part tool, such as Apache Commons math library.

For the missing value of v_{mn}, we can get different predictions based on its different high correlated system. We can make the prediction more reasonably by combining all the results of all its correlated system in S_h, shown as follows:

$$\hat{v}_{mn} = \frac{1}{\sum_{S_k \in S'} d(S_a, S_k)} \sum_{S_k \in S'} d(S_a, S_k) \bullet \hat{v}^{(k)}{}_{mn}, \tag{21}$$

wherein, $\hat{v}^{(k)}{}_{mn}$ is the prediction value made by system S_k, and $\hat{v}_{mn}$ is the final prediction value of v_{mn}.

4. Experiments

In this section, we perform experiments to validate our MOCC method and compare the results with those from other correlation computing methods. Our experiments are intended to (1) verify the rationality of Corollary 1 that is presented in Section 3.2 and (2) compare efficiency of MOCC method with other correlation computing methods.

4.1. Experiment Setup. This experiment was constructed by employing the data of a typical IoT system—the FY-3D weather satellite, one of the most advanced weather satellites globally. This satellite consists of energy, attitude, propulsion, thermal control, data transmission, cabin, and payload seven subsystems. Each subsystem can also be divided into smaller systems; for example, the attitude control consists of stellar positioning, gyroscope, and flywheel systems. There are more than 10000 sensors deployed on this satellite. All the sensors' monitoring data are transferred to the ground station periodically fourteen times one day.

It is important to work to analyze the correlation between two subsystems of the satellite since one's status often impacts another one. Satellite communications are vulnerable to interference. Therefore, data missing or abnormality usually occurs. This motivated the work of this paper.

4.2. Experimental Proof of Corollary 1. In Section 3.2, Corollary 1 says: if S_a is an IoT system with missing data and S_k is high correlated to S_a, then, the prediction of the missing data should keep $d(S_a, S_k)$ almost unchanged.

To prove this corollary, we have made statistics of all subsystems of FY-3D weather satellite for their correlations. For any two subsystems, we called it a system pair. This experiment is constructed as the following six steps:

(1) Determine a period with no missing data; use this data to do steps 2 to 6

(2) Calculate the MOCC correlation of any two systems, and determine their correlation belongs to which range. There are 10 ranges in total, i.e., [0, 0.1)… [0.9, 1]

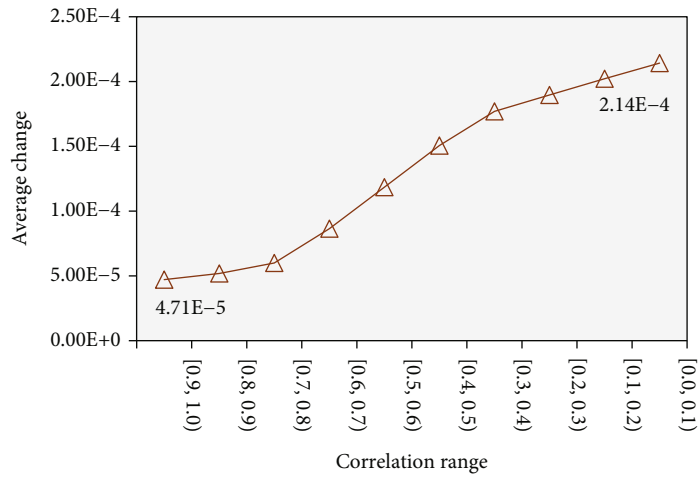


Figure 3: Average change in each correlation range, which proves Corollary 1.

Table 2: Accuracy comparison.

	Methods	Data = 5%		Data = 10%		Data = 20%		Data = 50%	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Voltage	TA	0.612	1.564	0.595	1.533	0.593	1.509	0.483	1.471
	MF	0.563	1.465	0.494	1.271	0.445	1.174	0.397	1.071
	MOOC	0.379	0.938	0.305	0.753	0.225	0.557	0.122	0.312
Current	TA	27.286	75.535	17.064	51.697	14.948	50.519	14.837	47.409
	MF	20.209	54.794	16.158	46.742	14.006	41.695	14.752	41.467
	MOOC	17.831	51.712	13.125	42.333	10.026	36.209	9.904	35.084
Temperature	LA	20.085	59.816	15.175	45.563	11.901	37.139	10.459	29.518
	MF	16.488	50.151	12.894	36.755	10.919	34.798	9.854	25.112
	MOOC	8.289	24.752	6.297	16.197	5.001	14.096	3.576	11.144

(3) For each system pair, randomly remove some data observed simultaneously. It can be seen to delete a column from Table 1 randomly, but the deletion of two tables should be at the same column position. The deletions were finished only 5% columns left. (Since we cannot know the actual values of the future, we deleted some values from the original dataset. Then, the dataset that deleted more values can be seen as "the past," and the dataset that deleted fewer values can be seen as "the future." By which, we can simulate the change of two systems' correlation along with time.)

(4) Once a deletion for a system pair finished, we calculated their new MOCC correlation and got the change value comparing their last correlation

(5) When all the deletions were finished, the average change of each system pair was got

(6) We find that if a system pair originally belonged to a relatively large range in step 2, then, their correlation would rarely change when deleting their data. The experimental result is shown as Figure 3

4.3. Comparisons. We compare the predictive accuracy for missing data of our MOCC method with other two well-known prediction methods. The two compared methods are the follows:

(1) *Time-Aware Method (TA).* This type of methods makes prediction based on the time factor, which was proposed in reference [16]

(2) *Matrix Factorization Based Method (MF).* This type of methods makes prediction by factorizing the dataset into matrices, and the reconstruct the dataset by multiplying these matrices. We selected the method proposed in reference [17] to compare with

We use the mean absolute error (MAE) and root mean squared error (RMSE) to measure the prediction accuracy. MAE and RMSE are defined as (22) and (23), respectively:

$$\mathrm{MAE} = \frac{\sum |v_{mn} - \hat{v}_{mn}|}{N}, \tag{22}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum (v_{mn} - \hat{v}_{mn})^2}{N}}, \tag{23}$$

where v_{mn} is a value in the dataset, and $\hat{v}_{mn}$ is its prediction value.

The predictions were made as follows:

(1) Randomly select some values which are not missing to predict, such we can compare the predictive value to the real value

(2) The dataset was made to different sparse ratio to test the prediction performance on spare data

There are many subsystems deployed on FY-3D satellite, we choose the battery system to make comparisons, and this system consists of the following sensors: (1) voltage, (2) current, and (3) temperature.

The prediction accuracies of MOCC and the comparisons with other methods are shown in Table 2. With reference to Table 2, we can see that MOCC is more accurate than all of the other methods for the two chosen datasets. As the data increases from 5% to 50%, the MAE and RMSE values become smaller.

5. Conclusion

We have enhanced the concept of distance correlation by bringing the time factor into it, which results in our MOCC method. This method considers all sensors of an IoT system as integration and can measure the correlation between subsystems accurately. We have also presented how to predict missing data values by our MOCC method. The prediction is based on an experimental proofed corollary, i.e., two highly correlated systems will rarely change their correlation with the data amount growing.

We also performed experiments to verify our corollary and our method's efficiency.

6. Related Work

Correlation between systems indicates their dependency, which is very important in system analysis. Based on correlation computing, we can determine the most relevant factor of the systems subsystem. For example, we can find that a system's data anomaly will affect other systems' functional status if there are high correlations between them; or a system's data missing can be predicted based on its high relevant systems. IoT systems consist of massive sensors producing data in matrix form. Therefore, the correlation must be calculated in multiple-dimensional ways.

There are many existing correlation computing methods, such as Pearson correlation coefficient (PCC) [8] and cosine coefficient (COS) [9], and new correlation measuring methods have been presented in recent years [4]. Many enhancements were brought into PCC and COS in recent years. By adding the weights to determine the different effects of different correlated objects, Zheng et al. [18] proposed an improved PCC correlation computation method and employed this method to predict missing data. Sun et al. [19] proposed a normalized correlation computing method to avoid the disadvantage that traditional PCC or COS neglects the mathematical features of observed vectors. However, all existing methods can only measure the correlation between one-dimensional vectors but not matrices. Figure 1 compares the vector and matrix-oriented similarity measurements.

Prediction of missing data has been widely studied in many fields, especially in the field of QoS prediction for service recommendation [20–22]. Correlation analysis is a crucial way to make a prediction. However, the missing value prediction based on distance correlation has rarely been studied before this paper.

AI methods such as neural networks were introduced to resolve multidimensional correlation analysis in recent years [13]. However, these black-box approaches cannot explain the mathematical mechanisms, and lots of memory or time are consumed.

Data Availability

Data are available on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Technology Research and Development Program of China (2021YFB3901000 and 2021YFB3901005) and NSFC (61602126).

References

[1] J. Singh, T. Pasquier, J. Bacon, H. Ko, and D. Eyers, "Twenty security considerations for cloud-supported internet of things," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 269–284, 2016.

[2] F. Gottwalt, E. Chang, and T. Dillon, "CorrCorr, a feature selection method for multivariate correlation network anomaly detection techniques," *Computers & Security*, vol. 83, pp. 234–245, 2019.

[3] A. Rawat, A. Gupta, A. Singh, and S. Bhushan, "Energy conservation and missing value prediction model in wireless sensor network," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pp. 1–5, Ghaziabad, India, April 2019.

[4] B. Liu, C. Gong, and Z. Xu, "Correlation analysis and link gain prediction for optical wireless scattering communication over broad spectra," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 1386–1396, 2020.

[5] K. Zhang, F. Zhao, S. Luo, Y. Xin, and H. Zhu, "An intrusion action-based IDS alert correlation analysis and prediction framework," *IEEE Access*, vol. 7, pp. 150540–150551, 2019.

[6] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel SVM with spatial–temporal correlation for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2001–2013, 2019.

[7] H. Yang, L. Gao, and G. Li, "Underwater acoustic signal prediction based on correlation variational mode decomposition and error compensation," *IEEE Access*, vol. 8, pp. 103941–103955, 2020.

[8] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating word of mouth," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 210–217, Denver USA, May 1995.

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, Hong Kong, April 2001.

[10] S. Beigi and A. Gohari, "Φ -entropic measures of correlation," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2193–2211, 2018.

[11] H. Aly and A. Winterhof, "A note on Hall's sextic residue sequence: correlation measure of order k and related measures of pseudorandomness," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1944–1947, 2020.

[12] J. Levin and G. Smith, "Optimized measures of bipartite quantum correlation," *IEEE Transactions on Information Theory*, vol. 66, no. 6, pp. 3520–3526, 2020.

[13] H. Chen, Z. Chen, Z. Chai, B. Jiang, and B. Huang, "A single-side neural network-aided canonical correlation analysis with applications to fault diagnosis," *IEEE Transactions on Cybernetics*, pp. 1–13, 2021, https://ieeexplore.ieee.org/document/9376288.

[14] G. J. Szekely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *The Annals Of Statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[15] G. J. Szekely and M. L. Rizzo, "Partial distance correlation with methods for dissimilarities," *The Annals of Statistics*, vol. 42, no. 6, pp. 2382–2412, 2014.

[16] Y. Jin, W. Guo, and Y. Zhang, "A time-aware dynamic service quality prediction approach for services," *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 227–238, 2020.

[17] J. Zhu, W. Ma, and Y. Song, "Attentive matrix factorization for recommender system," in *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 932–936, Chengdu, China, October 2020.

[18] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware web service recommendation by collaborative filtering," *IEEE Transactions on Services Computing*, vol. 4, no. 2, pp. 140–152, 2011.

[19] H. Sun, Z. Zheng, J. Chen, and M. R. Lyu, "Personalized web service recommendation via normal recovery collaborative filtering," *IEEE Transactions on Services Computing*, vol. 6, no. 4, pp. 573–579, 2013.

[20] W. Ma, Q. Zhang, M. Chunxiao, and M. Zhang, "QoS prediction for neighbor selection via deep transfer collaborative filtering in video streaming P2P networks," *International Journal of Digital Multimedia Broadcasting*, vol. 2019, Article ID 1326831, 10 pages, 2019.

[21] Y. Ma, S. Wang, P. C. K. Hung, C. Hsu, Q. Sun, and F. Yang, "A highly accurate prediction algorithm for unknown web service QoS values," *IEEE Transactions on Services Computing*, vol. 9, no. 4, pp. 511–523, 2016.

[22] S. Wang, Y. Ma, B. Cheng, F. Yang, and R. N. Chang, "Multidimensional QoS prediction for service recommendations," *IEEE Transactions on Services Computing*, vol. 12, no. 1, pp. 47–57, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8003887, 10 pages
https://doi.org/10.1155/2022/8003887

Research Article

A GNN-Based Variable Partition Framework for DCOPs

Chun Chen,[1,2] **Li Ning**,[1] **Rong Zhou**,[3,4] **Yong Zhang**,[1] **Chan Zhou**,[1] **and Shengzhong Feng**[4]

[1]*Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China*
[2]*University of Chinese Academy of Sciences, China*
[3]*Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, China*
[4]*National Supercomputing Center in Shenzhen, China*

Correspondence should be addressed to Li Ning; li.ning@siat.ac.cn

Received 14 January 2022; Accepted 10 March 2022; Published 20 May 2022

Academic Editor: Yan Huo

Many problems of the Internet of Things (IoT), such as radio frequency allocation and sensor network, can be regarded as constraint optimal problems (COPs), which can be formulated as graphical representations. The scale of graph is large, which is hard to implement, and the information shared by all the variables is unsafe for all the variables running in an agent. On the other hand, supercomputers are playing a significant and growing role in various fields of large-scale processing tasks. When countering this scene, the supercomputers can accelerate to complete the task according to the distributed solution, where they divide the task into sub-tasks and each sub-task is running on an agent, such as a process or a computation node. However, finding an optimal distributed solution is difficult to minimize the completion time with the optimal computing resources. Putting the task on too many agents not only wastes resources but also increases the risk of attacks. Conversely, fewer agents may take too much time, which is unacceptable for users. Determining the number of agents needs to strike a balance between communication and computation. In this paper, we propose a new framework GVPNN for predicting the optimal numbers of agents for COPs and further provide the allocation from variable to agent. Experimental result shows the framework can learn the structure of the corresponding graphical representation well, and the 1-distant accuracy rate and the top 3 accuracy rate of GVPNN reach 74% and 70%, respectively.

1. Introduction

Constraint optimal problems (COPs) is to maximize its aggregated utility or minimize its cost. It has a wide range of applications, such as meeting scheduling, resource allocation, smart homes, sensor networks [1], and many IoT problems [2, 3], which has attracted the attention of many researchers. The distributed constraint optimization problems (DCOPs) [4, 5] is a distributed implementation for COP, which is a general model to govern the agent's autonomous behavior in a cooperative multi-agent system (MAS). COPs and DCOPs can be often represented graphically using one of the following representations: constraint graphs, factor graphs, or pseudo-trees [6]. In this paper, we call them as graphical representations. In all of these graphical representations, nodes in these graphs (i.e., variables and/or factors) are held by the agents which are participating in the optimization process.

For many applications in COPs, large-scale graphical representations are hard to implement with all the variables running in an agent, or even cannot be handled. In addition, it is unsafe to run all the variables in an agent sharing the information. Supercomputers can provide a good platform. To speed up the calculation on the supercomputer, an appropriate suggestion is important to provide for users with the number of computing resources (agents) and the allocation from variables to agents after learning the graphical representation. Therefore, users can adopt suggestions before submitting the task and purchase computing resources on supercomputers at reasonable prices. For the supercomputer, the resource can be efficiently utilized, and the tasks can be completed in the shortest possible time. How to find

the optimal variable partition is very difficult because excessive agents are not only a waste resource but also increase the risk of being attacked. Otherwise, fewer agents may take too much time which is unacceptable for users.

In other words, the division of DCOPs is a plan that figures out the number of agents and the allocation from the variables to agents and have few work on the filed [7–10]. In the existing articles, many works focus on the DCOP algorithm and simply assigned one variable to an agent, which is negative for the large-scale DCOP. In ref [11], assume that the mapping of variables to agents is part of the model description, which means that variables that belong to each agent are given as an input. This assumption is reasonable in many applications where there are obvious and intuitive mappings. Take the smart home scheduling problem as an example; agents correspond to the different smart homes, and variables correspond to the different smart devices within each home. Under this scenario, it is reasonable for the agent to control all the variables, which are the devices in its home.

However, there are few scenarios like smart homes, and there may be more flexibility in other applications, which makes it hard to find the mapping from variables to agents. For example, imagine an application in which a group of unmanned robots needs to coordinate with each other to effectively survey an area. In this application, the agents correspond to robots, and the variables correspond to the different regions of the region to be solved. The domain for each variable may correspond to the different types of sensors to be used at different times to survey the region. Since a robot can survey any region, there are multiple possible assignments of regions to robots. That is, there are multiple possible mappings of variables to agents.

However, a good mapping is important as it has a significant impact on the completion time of an algorithm for the flexible scenarios. Choosing an optimal mapping may be prohibitively time-consuming as it is a NP-hard problem, as shown by Rust, Picard, and Ramparany [12]. To solve this problem, literature [13] proposed a time-efficient heuristic mapping algorithm (named MNA) based on the node's degree of the graphical representations. But this algorithm gives the agent number in advance and is only effective for messages passing algorithms such as Action-GDL, Max-Sum, or Bounded Max-Sum.

All the work discussed above focuses on the variable partition when the agent number is given. However, in many practical scenarios of large-scale distributed computing operations, different numbers of agents may lead to great differences in completion time, and the performance is not directly proportional to the number of agents. Therefore, it is very important to find the optimal number of agents for large-scale distributed computing operations, which can allocate computing resources of the supercomputers to users reasonably. In addition, compared with other DCOPs algorithms such as distributed random algorithm (DSA) [14] and distributed upper confidence tree algorithm (DUCT) [15], MNA has no advantage, and it is only effective when the distribution of the node degree in the graphical representations is casual. When the divergence of the node degree is unclear, the advantages of the algorithm cannot be highlighted.

In this paper, we commit to find the agent number when the DCOP algorithm is given and propose a new end-to-end variable partition framework. This framework employs graph neural networks (GNNs) to learn the structure of the corresponding graphical representation and then predicts the optimal number of agents and further provides the mapping from variables to agents.

The structure of the rest of this paper is as follows: Section 2 introduces the background and the definitions of the optimal number of agents, Section 3 describes the proposed framework and details the framework, Section 4 explains and analyzes the experimental results, and we conclude in Section 5.

2. Preliminaries

In this section, we introduce the background information of DCOPs and the objective of the paper.

2.1. Constraint Optimization Problem. A COP is a tuple $< X, D, F >$. X is a discrete and finite set of variables $\{x_1, x_2, \cdots, x_n\}$. D is a set of domains $\{D_1, D_2, \cdots, D_n\}$. Each domain D_i contains a discrete and finite set of values that can be assigned to variable x_i. We denote an assignment of value $d_{ij} \in D_i$ to x_i by an ordered pair $<x_i, d_{ij}>$. F is a set of relations (constraints). Each constraint $f_j \in \mathrm{F}$ defines a non-negative cost (or aggregated utility) for every possible value combination of a set of variables and is of the form $f_j : d_{i1} \times d_{i2} \times \cdots \times d_{ik} \longrightarrow F + \cup\{0\}$.

2.2. Distributed Constraint Optimization Problem. A DCOP is a distributed implementation for COPs, which is a tuple $<X, D, F, A, \mu>$. X, D, F is the same as COPs. A is a finite set of agents $\{a_1, a_2, \cdots, a_k\}$. $\mu : \mathrm{X} \longrightarrow \mathrm{A}$ is mapping from a set of nodes in the corresponding graphical representation to agents.

An optimal solution for COPs and DCOPs is an assignment with minimized cost or maximal utility. To be consistent with the literature, the aim for COPs and DCOPs in this paper default to minimize the cost, which is

$$X^* = \mathrm{argmin}_X \sum_{i=1}^{n} f_i(x_i), \tag{1}$$

2.3. Variable Partition. Given an agent number k, where $k > 1$, an arbitrary mapping manages a subgraph G_j of the graphical representations G, and k agents hold all the nodes which no nodes hold for different agents. The partition can be described as follows:

$$\begin{cases} U_{j=1}^{k} G_j = G \\ G_j \cap G_j' = \varnothing \quad \forall j' \neq j \end{cases}. \tag{2}$$

Given an arbitrary partition algorithm, the divergence of completion time may be huge for different number of agents.

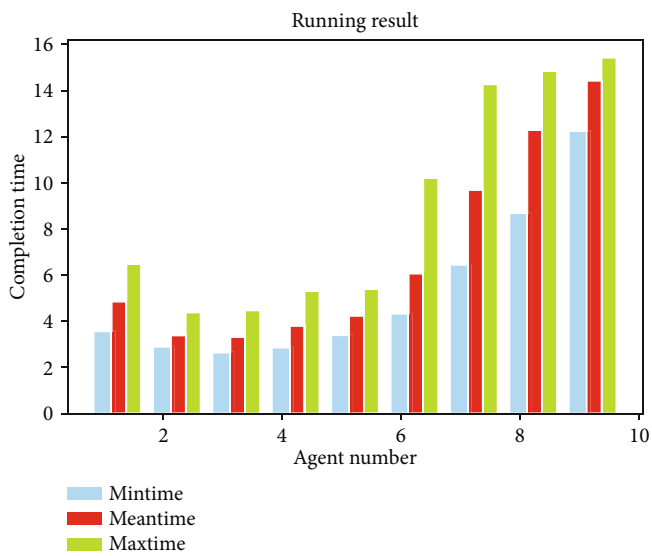


Figure 1: The completion time with different numbers of agents for a graph coloring problem, in which the graphical representation has 2 communities. For each partition solution, asynchronous execution DSA algorithm 10 rounds compute the mean time by formula (3), which is shown in the red bar. The min time (blue bar) and the max time (green bar) are minimal and maximal completion time in the 10 rounds.

Here, we give an example of a graph coloring problem (a special COP) on different agent numbers with partition by Girvan-Newman algorithm [16], in which an agent is asynchronous execution DSA algorithm on a process. The result can be shown in Figure 1. From the figure, we find that the optimal number is 3 (is computed by Formula (4)) and the mean completion time (is computed by Formula (3)) with 9 agents is more than 4 times slower than the mean completion time on the optimal number—3 agents.

2.4. The Optimal Agent Number. In this paper, the goal is to find the optimal number of agents and the mapping for an arbitrary COP. In this section, we first define the optimal number of agents and then verify the necessity of finding the optimal number of agents with an example.

To solve a COP with a large-scale graphical representation, an approximation algorithm can be the best choice. The completion time of the two experiments is different, and the variance is tiny from the expected value shown in the experimental result. Thus, we aim to find out the expected number of agents which are mostly performed the best. In this paper, the definition for the number of exceptional agents is given.

Given a COP problem p and the number of agent number k, it is defined that the valid completion time of each round i is the time t_{k_i} when the cost of the DCOP algorithm reaches 0 for the first time. In this paper, we supposed that each instance for DCOPs can be solved so that the cost can arrive at 0 and the total number for the round is n.

Since the Law of Large Numbers (LLN) describes the result of performing the same experiment many times, the average value of the results obtained from many experiments should be close to the expected value, and the average value will become closer as more experiments are performed in probability theory. In this paper, the expected completion time of agent number k is t_k, which is approximately defined as the average running time:

$$t_k = \frac{\sum_{i=1}^{n} t_{k_i}}{n}, \tag{3}$$

where t_{k_i} is the completion time time of the i round with k agent number and n the total number of rounds.

Then, we define the optimal number of agents with the minimum expected completion time as follows:

$$p_{ban} = \operatorname{argmin}(t_1, t_2, \cdots, t_k, \cdots, t_N), \tag{4}$$

where N is the total number of agents that can run on the supercomputer.

Finding the optimal agent number, the most immediate idea is searching for the number with a maximum module as most of the graphical representations for COPs are sparse and have the community structure. Girvan-Newman [16] proposes a module—Q value—which is an indicator to measure the quantity of clustering:

$$Q = \sum_i \left(e_{ii} + a_i^2\right), \tag{5}$$

where i is the i_th community, e_{ii} is the ratio of the edges of community i to all the edges of the original network, and a_i represents the ratio of all the edges connected with the vertices in the community i to the total number of edges.

From Formula (5), we find that the higher Q value, the better the corresponding community division results, and the best community division is the one with the largest Q value.

However, we find that the number of the optimal module may take more time than the agents of the other number. From Figure 2, we can find that there is no direct relationship between the minimum completion time and the

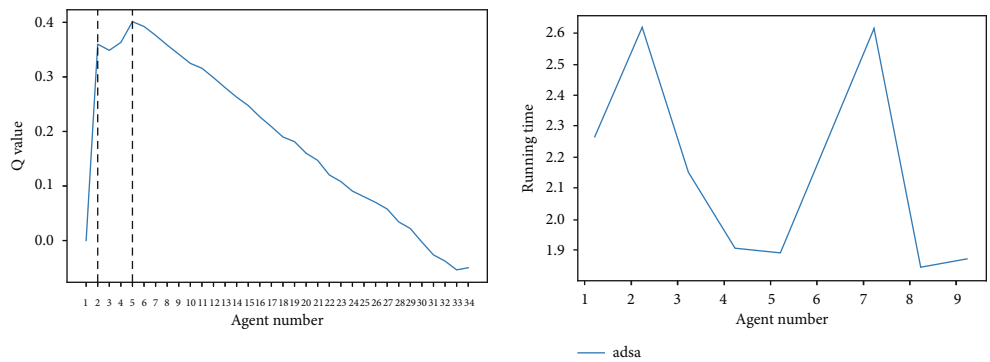


Figure 2: Q value and the mean completion time for a graph coloring problem with karate data.

module—Q value—in which the maximum q value is located on the agent number equal to 5, but the corresponding agent number of the minimum completion time is 8.

Therefore, this paper proposes a GVPNN framework to learn the characteristics of the graphical representations and find the best mapping for variables and agents, including the number of agents and specific mappings for each agent. In the next section, we will give more detail about the framework.

3. Graph Variable Partition Neural Network Framework

In this section, we first roughly introduce our Graph Variable Partition Neural Network Framework—GVPNN framework—which is based on graph neural networks [17–20]. When any COP arrives, this framework can abstract the COP to a graphical representation and then import the graphical representation into the graph neural networks to find the optimal number of agents, as shown in Figure 3. Finding the optimal number of agents is a graph classification problem in the domain of graph neural networks, which means that it predicts the label of an entire graph by learning a graph representation vector when given a group of graphs and their corrpending labels.

For an arbitrary node, it updates the representation vector by recursively aggregating and transforming feature vectors of its neighboring nodes in aggregate and combine stages. In GVPNN framework, we employ the GIN [21] and GraphSNN [22], which aggregate the feature of neighbor by multiset and aggregate the neighbor's feature and the overlap structure by the structural coefficients defined in Formula (8). After all the nodes are updated, the GVPNN obtained the feature of the whole graph. In the readout stage, all node features of the graph are converted into graph features, and then the optimal number of agents is predicted. Further, catch the optimal variable partition by Girvan-Newman algorithm.

3.1. Node Representation. The vector of a node representation is updated by recursively aggregating and transforming feature vectors of its neighboring nodes in the aggregate stage and combine stage. After $t+1$ aggregation iterations, we can capture the structure information of the neighborhood of the node's $(t+1)$-hop network. Formally, the node representation in the $(t+1)$_th layer can be represented as

$$\begin{aligned} a^{(t+1)} &= AGGREGATE^{(t+1)}\left(h^{(t)} : u \in N(v)\right), \\ h^{(t+1)} &= COMBINE^{(t+1)}\left(h^{(t)}, a^{(t+1)}\right). \end{aligned} \tag{6}$$

where $N(v)$ is a set of nodes adjacent to v.

In our paper, we employ two GNNs—GIN [21] and GraphSNN [22]—to update the node representation in our framework, as shown in Figure 4, which are all based on the Weisfeiler-Lehman test. The Weisfeiler-Lehman test is a test of graph isomorphism, which is an effective and computationally efficient test to verify whether two graphs are topologically identical in most cases. Node representation of u for the Weisfeiler-Lehman [23] test after $t+1$ iterate is the sub-tree structure of height $t+1$ rooted at the node u , which is updated by a hash function, which models injection multi-set functions of the neighbor aggregation.

Node representation for GIN is similar to 1- WL, in which the neighbor aggregation is also injected. Only when two nodes have the same sub-tree structure and have the same characteristics on the corresponding nodes, GIN will map these two nodes to the same location, where the sub-tree structures are recursively defined by the neighborhoods of the node. Representing a neighborhood as a multiset of feature vectors and treating the neighborhood aggregation as an aggregation function over multisets. To ensure injectivity, GIN sets the aggregate function to sum and the combination function as $(1+\epsilon^{(t+1)})$. The node representation is updated as

$$h_v^{(t+1)} = MLP^{(k)}\left(\left(1+\epsilon^{(k)}\right)h_v^{(t)} + \sum_{u \in \mathcal{N}(v)} h_u^{(t)}\right). \tag{7}$$

GraphSNN injects local structure into an aggregation scheme, considering not only the neighbor's feature but also the overlap subgraphs, which is more expressive than 1-WL. This GraphSNN define the structural *coefficients* $\omega(S_v, S_{uv})$ for each vertex v and its local neighborhood, i.e., $\omega : S \times S^* \longrightarrow R$ such that $A_{vu} = \omega(S_v, S_{uv})$, which satisfies the properties of local closeness, local denseness,

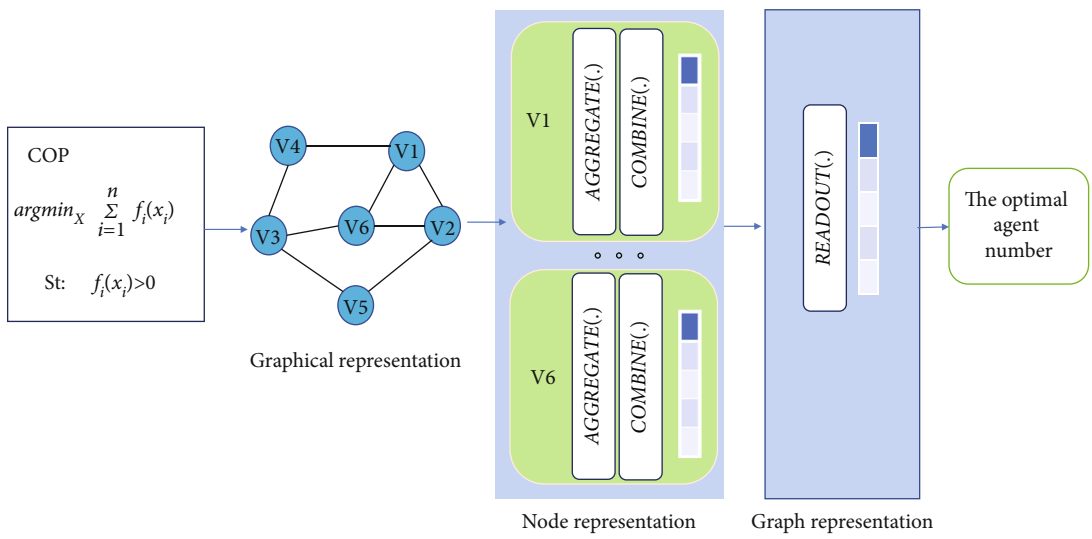


Figure 3: The framework for GVPNN.

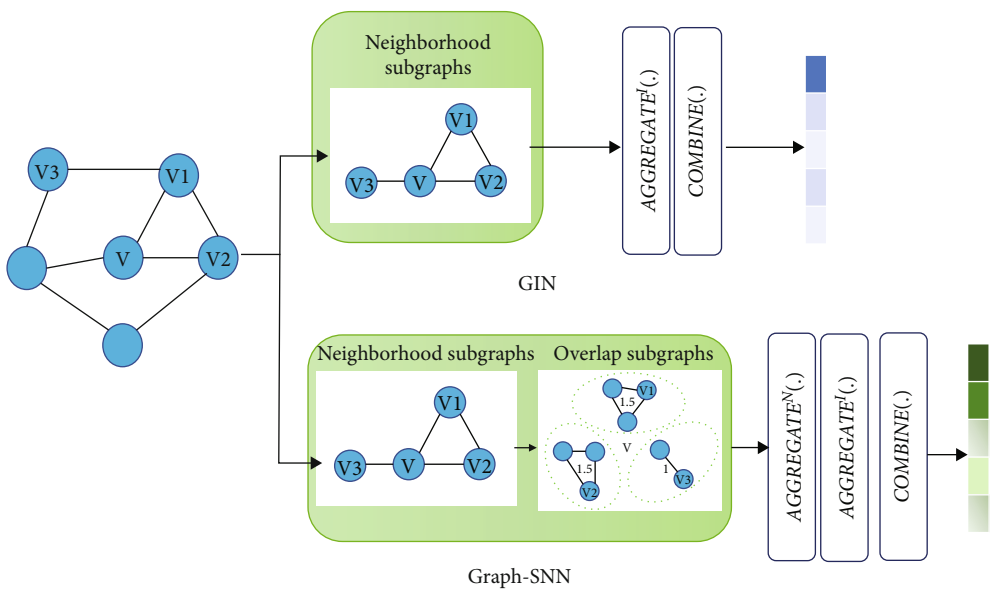


Figure 4: Node representation by GVPNN. For any node, GIN updates the representation vector by aggregating the neighbor's feature by multi-set, while GraphSNN updates the representation vector by aggregating the neighbor's feature and the overlap structure.

and isomorphic invariant:

$$\omega(S_v, S_{uv}) = \frac{|E_{vu}|}{|V_{vu}| \bullet |V_{vu} - 1|} |V_{vu}|^{\lambda}, \tag{8}$$

where $\omega(S_v, S_{uv})$ is a structural coefficient for vertex v and its neighbors. S_v is the neighborhood subgraph for vertex v, and S_{uv} is the set of overlap subgraphs for vertex v and $\lambda > 0$.

GraphSNN also define a weighted adjacency matrix $\tilde{A} = (\tilde{A}_{vu})_{v,u \in V}$, where $\tilde{A}_{vu}$ is a normalized value of A_{vu} and $\tilde{A}_{vu} = A_{vu} / \sum_{u \in \mathcal{N}(v)} A_{vu}$. The node feature vector of v is updated by

$$\begin{aligned} m_a^{(t)} &= AGGREGATE^N\left(\left\{\left\{\left(\tilde{A}_{vu}, h_u^{(t)}\right) | u \in \mathcal{N}(v)\right\}\right\}\right) \\ m_v^{(t)} &= AGGREGATE^I\left(\left\{\left\{\tilde{A}_{vu} | u \in \mathcal{N}(v)\right\}\right\}\right) \\ h_v^{(t)} h_v^{(t+1)} &= COMBINE\left(m_v^{(t)}, m_a^{(t)}\right). \end{aligned} \tag{9}$$

$AGGREGATE^N(\cdot)$ and $AGGREGATE^I(\cdot)$ are two possibly different parameterized functions. Here, $m_a^{(t)}$ is a message aggregated from the neighbors of v and their structural coefficients, and $m_v^{(t)}$ is an "adjusted" message

from v after performing an element-wise multiplication between $AGGREGATE^I(\cdot)$.

Specifically, GraphSNN details the $AGGREGATE^N(\cdot)$, $AGGREGATE^I(\cdot)$, and $COMBINE(\cdot)$. Thus, for each vertex v∈V, the feature vector at the $t+1\_th$ layer is generated by

$$h_v^{(t+1)} = MLP_\theta\left(\gamma^{(t)}\left(\sum_{u\in\mathcal{N}(v)}\tilde{A}_{vu}+1\right)h_v^{(t)} + \sum_{u\in\mathcal{N}(v)}\tilde{A}_{vu}+1\right)h_u^{(t)}\Bigg). \tag{10}$$

where $\gamma(t)$ is a learnable scalar parameter. Since $\mathcal{N}(v)$ refers to one-hop neighbors of v, one can stack multiple layers to handle more than the one-hop neighborhood. To ensure the injectivity in the feature aggregation,the graphSNN adds 1 into the first and aecond terms in the formula (10).

3.2. Graph Representation. For the graph classification problem, it is necessary to transform all the node features in the graph into graph features, and the representation of the entire graph is h_G:

$$h_G = READOUT\left(h^{(t+1)}\middle|v\in G\right), \tag{11}$$

where h_G is graph G representation vector and *READOUT* represents a permutation invariant function and can also be a graph-level pooling function. This *READOUT* function of the two GNNs is injective.

To consider all the structure information, the GVPNN framework utilizes the information from all iterations of the models and adopts an architecture like Jumping Knowledge Networks. The graph representations are concatenated across all the iterations/layers, and the *READOUT* function is summing all node features from the same iterations:

$$h_G = CONCAT\left(SUM\left(\left\{h_v^{(t+1)}|v\in G\right\}\right)\right)|k=0,1,\cdots,K). \tag{12}$$

4. Experiment

All the experiments were executed on a server equipped by an Intel Xeon CPU 4110 with 20 cores of 2.20 GHz. The system is Linux 3.10.0, and all DCOP is implemented in the PyDCOP library.

4.1. Graph Coloring Benchmark. In the experiment, the graph coloring problem is used to benchmark coordination algorithms for our COP problem. In this work, in distributed graph coloring problems, variables in the graph must select their color (i.e., the state) from a set of possible colors (i.e., $x_i \in 1,\cdots,c$) and avoid conflicts (i.e., choosing the same color) with other variables. Thus, the cost is expressed as

$$U_m(x_m) = \gamma_m(x_m) - \sum_{i\in\mathcal{N}(m)\backslash m} x_i\otimes x_j \tag{13}$$

where,

$$x_i\otimes x_j = \begin{cases} 10, & \text{if} x_i = x_j \\ 0, & otherwise \end{cases}, \tag{14}$$

and $\gamma_m(x_m) << 1$ reflects the preference for any color in the absence of conflicts. As before, the goal is to find the state of each variable such that the total number of conflicts is minimized. In this work, we set $\gamma_m(x_m) = 0$ for the variable and set the conflict cost $x_i\otimes x_j$ =10 if two variable select the same coloring.

4.2. Dataset. In this paper, we choose three kinds of random graph datasets generated by networks to accomplish the graph coloring problem, and these graphs are undirected, unweighted, and connected:

(1) 991 instances of the 11-color random graph in the first dataset are generated by Stochastic Block model in which the number of communities is ranged from 2 to 6. The intra-block and cross-block probabilities are set to 0.001-0.002 and 0.1-0.3, respectively, and the size |v| of each graph was 200. Figure 5(a) give an example for a graph with 4 communities that the intra-block is 0.2 and cross-block probabilities is set to 0.001

(2) 454 instances of the 11-color random graph in the second dataset are generated by Erdős–Rényi model, which 307 with 200 nodes and 200-400 edges generated by gnm function and 147 instances of the 11-color random graph with excepted nodes of 200, and the probability between two nodes is 0.001 to 0.0025 generated by the gnp function. Figure 5(b) gives an example for a graph with 200 nodes and 350 edges

(3) 489 instances of the 11-color random graph with 200 nodes in the third dataset generated by the Barabasi Albert model. Figure 5(c) gives an example for a graph with 200 nodes

4.3. Labeling. For training in the framework GVPNN, we should first give a label for all the graphical representations. The labeling process includes the variable division, DCOP algorithm selection, and the optimal agent number calculation.

As the object is to find the optimal agent number with the minimum completion time, the initial allocation should be robust, and each part after being divided should not overlap. Many graphical representations are sparse and have a structure of communities. So in his work, the Girvan-Newman algorithm is adopt to divide each instance graph into k classes, k in [1,n], in which k= 1 means all the nodes in the same partition and each node is a apartition when k=N.

The Girvan-Newman algorithm [16] is proposed for graph partitioning in parallel computing, which minimizes the number of edges that run between processors. Giran-

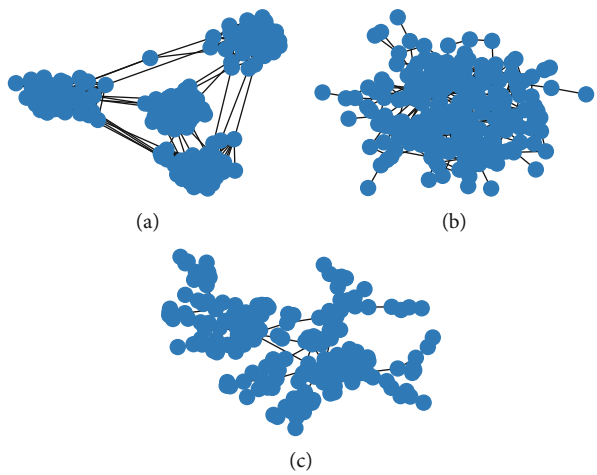


Figure 5: Random graphs generated by various models.

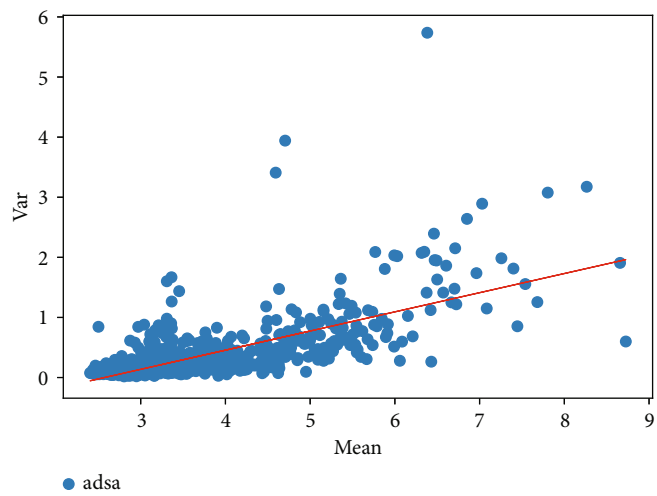


Figure 6: The relationship between the expected completion time and the variance.

Newman method is a hierarchical method, which detects communities by gradually removing edges from the original network, and the remaining connected networks are the communities, which ensure the non-overlap among communities. The algorithm deletes the edges that are most likely "between" communities, an edge as the number of shortest paths between pairs of nodes that run along with it, which is named "edge betweenness." After removing the edge with the highest score betweenness, the Girvan-Newman algorithm recalculates betweenness for all remaining edges, which are robust for the whole graph partition.

For algorithms, the field of classical DCOPs is mature, and lots of different solution algorithms have been proposed. According to whether the DCOP algorithm can guarantee the optimal solution, or whether it can trade optimality for shorter execution times, to generate the near-optimal solutions, the algorithms can be divided into complete algorithms and incomplete algorithms. For incomplete algorithms, there are three categories, such as search-based algorithms such as distributed random algorithm (DSA), maximum gain message (MGM), a reasoning-based algorithm such as max-sum algorithm, and sampling-based algorithm such as distributed upper confidence tree (DUCT) algorithm and distributed Gibbs (D-Gibbs) algorithm [24].

Because DSA is a good robust benchmark, and it tends to find high-quality solutions in practice, we choose to implement the DSA algorithm. Asynchronous actions may improve the performance of a DSA algorithm. So, this paper uses asynchronous DSA to get the completion time.

DSA requires an activation probability p before choosing new assignments, so we adopted $p = 0.7$ as reported in [14]. In addition, we used DSA version B because such a decision process is known to be more aggressive than other versions [14].

After picking the variable partition method and DCOP algorithm, we start to label the graphical representation of each instance of COPs. Specifically, we run the DSA algorithm for 10 rounds, in which each agent manages a partition on a process in the PyDCOP Library, and the optimal number of agents is calculated by the Formulas (3) and (4).

Because it is an approximate algorithm to run the DCOPs algorithm to find the best number for agents, we need to ensure the stability and effectiveness of the labeled result at first. To prove that the average time is stable and

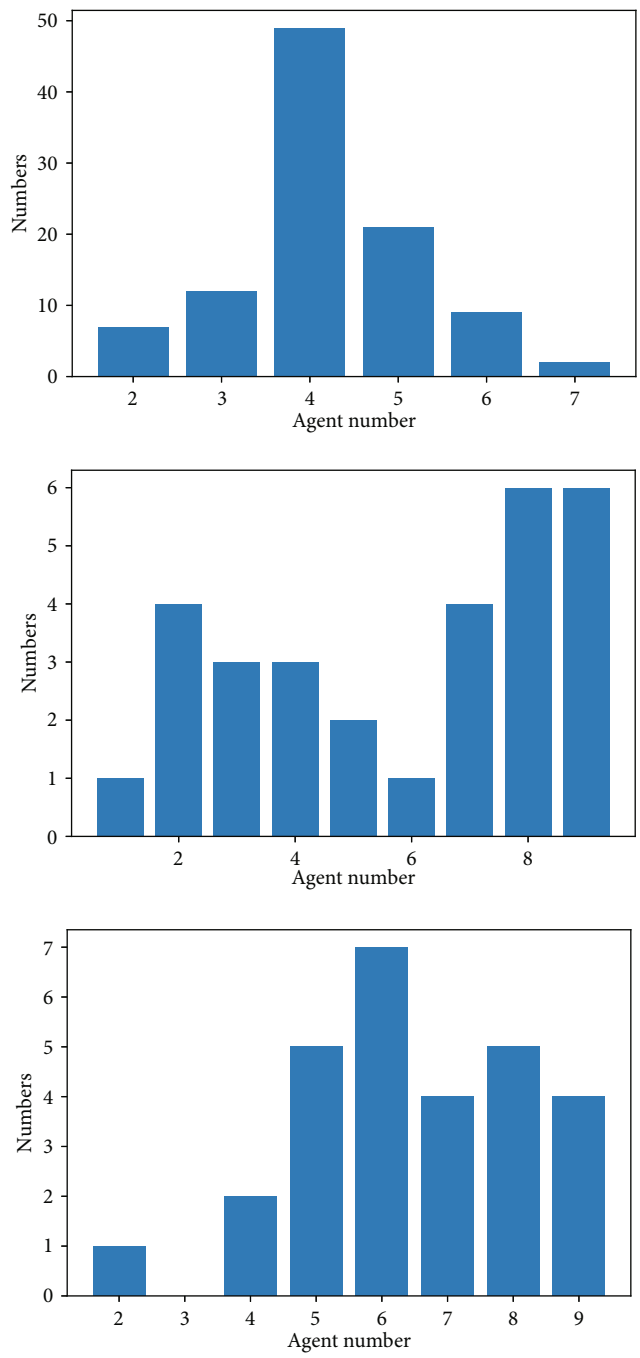


Figure 7: The best process distribution for random graphs.

effective, we analyze the relationship between the expected completion time of the distribution solution and the result variance, as shown in Figure 6.

From Figure 6, we found that the variance of the fitting curve coefficient was low, about 1/5 times the average value. The expected time can be considered the label of the graphical representations.

For the label distribution, we expect that it can be any value in the scope of supporting resources, not just the first three. Therefore, before marking the graph, we test various random graphs and make statistics on their distribution. Here, we give an example of optimal agent number distribution under different graph densities, as shown in Figure 7.

As shown in Figure 7, we find that the distribution of the optimal number of agents is diverse when analyzing graphs, which can meet the application of supercomputers. With the increase of edge, the number of agents may be limited to 3 agents.

5. Results

We train the GVPNN framework with GIN and GraphSNN on the graph coloring problem and compute the accuracies of the GNNs. As an agent is implemented in a process, the accuracy comes from existing documents:

$$Accuracy = \frac{NUM_{pred}}{NUM_{totle}}, \tag{15}$$

where NUM_{pred} is the $number\ pred_s = v_{opt}$.

What's more, we also use 1-distant accuracy and top 3 accuracy as the measurement standard. For the supercomputer, providing more or less than 1 process than the optimal process to COPs is also reasonable. So, 1-distant accuracy is set as follows:

$$Accuracy_{1dist} = \frac{NUM_{pred\_s}}{NUM_{totle}}, \tag{16}$$

where NUM_{pred_s} is the $number\ pred_s \in \{v_{opt} - 1, v_{opt}, v_{opt} + 1\}$.

Meanwhile, the completing time of the top 3 is always proximity. For the user, the prediction of the optimal agent number to a set with the top 3 for COPs works well. Thus, this paper shows the top 3 accuracy setting as follows:

$$Accuracy_{top3} = \frac{NUM_{pred\_top3}}{NUM_{totle}} \tag{17}$$

where NUM_{pred_top3} is the $number\ pred_s \in \{v_{top1}, v_{top2}, v_{top3}\}$.

Table 1 lists the results for GIN and GraphSNN. From the results, we found that the GVPNN can learn the structure of the graphical representations well. In terms of accuracy, GraphSNN network can reach $54.69\% \pm 6.59\%$ for the random graph with communities and improved 36% by GIN. For the random graph and the total graph dataset, the accuracy of GraphSNN and GIN is relatively close, which means that the overlap structure features in random graphs are not obvious.

For the supercomputer, the 1-distant precision GraphSNN can be improved more than GIN on three datasets, which is 48.5% for the random graph with communities, 65.1% for the random graph, and 31.2% for general graphs. For the user, the improvement of the accuracy of the top three GraphSNN is 23.4% for the random graph with communities, 12.5% for the random graph, and 4.9% for the total graph.

Table 1: Classification accuracies (%).

The random graph with communities			
Graphs	991		
Node degree avg/std/min/max	9.97/2.83/1/27		
Edge avg/std/min/max	997/29/896/1108		
	Accuracy	$Accuracy_{1dist}$	$Accuracy_{top3}$
GIN	40.2 ± 2.31	55.3 ± 5.72	69.3 ± 7.2
GraphSNN	54.69 ± 6.59	82.14 ± 6.84	85.57 ± 12.09
The random graph			
Graphs	943		
Node degree avg std/min/max	3.03/2.75/1/79		
Edge avg/std/min/max	303/109/199/728		
	Accuracy	$Accuracy_{1dist}$	$Accuracy_{top3}$
GIN	40.44 ± 1.83	45.2 ± 2.15	62.1 ± 3.60
GraphSNN	41.20 ± 1.66	74.65 ± 2.18	69.89 ± 4.74
The total graphs combine the random graph with communities and the random graph			
Graphs	1934		
Node degree avg/std/min/max	6.59/4.45/1/79		
Edge avg/std/min/max	658.86/355.7/199/1108		
	Accuracy	$Accuracy_{1dist}$	$Accuracy_{top3}$
GIN	39.6 ± 4.98	56.8 ± 3.96	67.3 ± 2.15
GraphSNN	41.11 ± 5.07	74.52 ± 3.75	70.63 ± 4.63

The result shows that GraphSNN can learn the structure of the graphical representations better than GIN and the predicted distribution of DCOPs' new graph is reasonable.

6. Conclusion

This paper presents an efficient framework for variable partition for DCOPs which is a pre-processing for DCOPs. COPs generates the labeled dataset by running the DCOPs algorithm based on the distribution during the Girvan-Newman algorithm. When a new COP arrives, GVPNN learns the graphical representation structure and further predicts the distribution. Experiments show that the framework can learn the architecture of graphical representations well, and the 1-distant accuracy of GraphSNN can reach 74.5%, and the top 3 accuracy can reach 70.6%. However, this framework only worked for Girvan-Neman community detection and DSA algorithm. In the following research, we will keep working to find the optimal variable partition with hybrid graph partition and DCOP algorithm.

Data Availability

The COP data (graphs) used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

This research was supported by the National Key Research and Development Project of China (Grant No. 2019YFB2102500) and the National Natural Science Foundation of China (Grant No. NSFC 12071460).

References

[1] W.-T. Chan, F. Y. L. Chin, D. Ye, Y. Zhang, and H. Zhu, "Greedy online frequency allocation in cellular networks," *Information Processing Letters*, vol. 102, no. 2-3, pp. 55–61, 2007.

[2] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 38, no. 5, pp. 968–979, 2020.

[3] J. Li, A. M. V. V. Sai, X. Cheng, W. Cheng, Z. Tian, and Y. Li, "Sampling-based approximate skyline query in sensor equipped IoT networks," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 219–229, 2021.

[4] F. Fioretto, E. Pontelli, and W. Yeoh, "Distributed constraint optimization problems and applications: a survey," *Journal of Artificial Intelligence Research*, vol. 61, pp. 623–698, 2018.

[5] Y. Zhang and Y. L. Francis, "A 1-local asymptotic 13/9-competitive algorithm for multi-coloring hexagonal graphs," *Algorithmica*, vol. 54, no. 4, pp. 557–567, 2009.

[6] A. Petcu and B. Faltings, "A distributed, complete method for multi-agent constraint optimization," in *CP 2004 - fifth international workshop on distributed constraint reasoning*, vol. 15, pp. 1–15, 2004.

[7] F. Fioretto, W. Yeoh, and E. Pontelli, "Multi-variable agent decomposition for DCOPs," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 30, pp. 2480–2486, Phoenix, Arizona, USA, 2016.

[8] P. Rust, G. Picard, and F. Ramparany, "On the deployment of factor graph elements to operate max-sum in dynamic ambient environments," in *International Conference on Autonomous Agents and Multiagent Systems*, pp. 116–137, Springer, Cham, 2017.

[9] P. Rust, G. Picard, and F. Ramparany, "Self-organized and resilient distribution of decisions over dynamic multi-agent systems," *International Workshop on Optimization in Multiagent Systems*, vol. 2018, 2018.

[10] A. Farinelli, A. Rogers, A. Petcu, and N. Jennings, "Decentralised coordination of low- power embedded devices using the max-sum algorithm," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, vol. 2, pp. 639–646, Estoril, Portugal, 2008.

[11] F. Fioretto, W. Yeoh, and E. Pontelli, "A multiagent system approach to scheduling devices in smart homes," in *Proceedings of the 16th Conference on Autonomous Agents and Multi-Agent Systems*, pp. 981–989, Sao Paulo, Brazil, 2017.

[12] P. Rust, G. Picard, and F. Ramparany, "Using message-passing DCOP algorithms to solve energy-efficient smart environment configuration problems," in *In proceedings of the twenty-fifth inter- national joint conference on artificial intelligence, IJCAI16, 468474*, pp. 468–474, New York, USA, 2016.

[13] P. Agrawal, A. Kumar, and P. Varakantham, "Near-optimal decentralized power supply restoration in smart grids," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, vol. 69, pp. 1275–1283, Istanbul, Turkey, 2015.

[14] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg, "Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks," *Artificial Intelligence*, vol. 161, no. 1-2, pp. 55–87, 2005.

[15] B. Ottens, C. Dimitrakakis, and B. Faltings, "DUCT: an upper confidence bound approach to distributed constraint optimization problems," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 5, pp. 1–27, 2017.

[16] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, pp. 1–15, 2004.

[17] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *AAAI Conference on Artificial Intelligence*, pp. 4438–4445, New Orleans, LA, USA, 2018.

[18] R. Sato, "A survey on the expressive power of graph neural networks," 2020, http://arxiv.org/abs/2003.04078.

[19] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[20] B. M. Oloulade, J. Gao, J. Chen, T. Lyu, and R. Al-Sabri, "Graph neural architecture search: a survey," *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 692–708, 2021.

[21] X. Keyulu, H. Weihua, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks," in *In International Conference on Learning Representations (ICLR)*, New Orleans, LA,USA, 2019.

[22] A. Wijesinghe and Q. Wang, "A new perspective on "how graph neural networks go beyond Weisfeiler-Lehman?"," in *In International Conference on Learning Representations (ICLR)*, 2021.

[23] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels," *Journal of Machine Learning Research*, vol. 12, no. 9, pp. 2539–2561, 2011.

[24] D. T. Nguyen, W. Yeoh, and H. C. Lau, "Distributed Gibbs: a memory-bounded sampling- based DCOP algorithm," in *In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, vol. 69, pp. 167–174, Saint, MN, USA, 2013.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2432351, 15 pages
https://doi.org/10.1155/2022/2432351

Research Article

Graph Embedding-Based Sensitive Link Protection in IoT Systems

Yanfei Lu, Zhilin Deng, Qinghe Gao, and Tao Jing

School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing, China

Correspondence should be addressed to Qinghe Gao; qhgao@bjtu.edu.cn

Received 8 December 2021; Revised 21 February 2022; Accepted 22 March 2022; Published 30 April 2022

Academic Editor: Chunqiang Hu

In the Internet of Things (IoT), massive interconnected intelligent terminal devices constitute diverse networks. Link prediction can serve as a powerful inference attack to speculate the sensitive links in the networks, posing a security threat to entity privacy in IoT. Most antilink prediction methods reduce the prediction ability of link prediction models through link disturbance to hide sensitive links but fail to consider the impact of node attributes on link prediction. This paper proposes a sensitive link protection method based on graph embedding (SLPGE) to combat link prediction attacks. This method is aimed at compressing network topology data into an embedding matrix and lessening private information by combining Variational Graph Autoencoder (VGAE) and Adversarially Regularized Variational Graph Autoencoder (ARVGA). Based on our experiment on two datasets, SLPGE reduces the prediction accuracy of two attack models for sensitive links by up to 30.05% and 15.03% compared to the original data, and the corresponding utility sees a drop of 7.54% and 7.79% at most, which verifies the feasibility of SLPGE—achieving the tradeoff between privacy protection and data utility effectively.

1. Introduction

To build a highly automated, informative, and intelligent system, the Internet of Things (IoT) integrates numerous communication, computing, and sensing devices, ranging from smartphones to vehicles [1], which is an organic collection of intelligent terminal devices and users. In IoT, widely distributed terminal devices establish reliable wireless links through advanced wireless communication and network technology, forming distributed multidomain networks [2]. Networks are ubiquitous in the real world, such as communication networks, social networks, biological networks, and transportation networks, represented by graphs containing nodes and edges. Similarly, the networks in IoT can also be regarded as graphs with terminal devices as nodes and communication links as edges. Although attractive and convenient, IoT also brings a significant challenge, i.e., the concerns on privacy disclosure [3]. As a new paradigm of big data platform, IoT deploys smart city applications to timely monitor, analyze, and respond to volumes of physical data. The data in IoT collected in a distributed manner are strongly correlated with users' sensitive status. However, some information platforms disclose private information inadvertently while trading the data, most likely the graphs in IoT. Furthermore, it does not rule out the possibility that malicious attackers may spy on entity privacy, analyze network traffic, and track users' behavior by stealing the complete network graphs, which invade the entity privacy and threaten the security of the IoT system. At present, the study on privacy for IoT mainly focuses on the privacy of data, identity, and location [4], while rarely mentioning graph privacy, especially the privacy of the communication links between terminal nodes in graphs, i.e., sensitive links. Actually, the disclosure of sensitive links will bring many security threats to the IoT system. For example, some sensitive links usually involve personal privacy, such as the doctor-patient relationship in smart healthcare, one of the typical application scenarios of IoT, and the user trajectories that data requesters may expose when accessing IoT.

In addition, in the man-in-the-middle (MITM) attack, hackers will try to intercept private data; control devices in smart homes, smart industries, and smart healthcare; or destroy the communication links in the IoT system, resulting in privacy disclosure, device failure, and even system collapse, which seriously threaten personal privacy, business activities, and industrial operations. Hence, it is imperative to detach private information from the graphs in advance. The most straightforward operation to hide the sensitive links is to delete the sensitive links in the graphs directly. Unfortunately, sensitive links may be predicted out of released data through data mining techniques, even if they have been deleted [5]. As an essential task in data mining, link prediction has been heating up in recent years. More and more link prediction methods and their application technologies have been proposed. Link prediction can predict the relationship between nodes by mapping the graph information to a continuous vector space. While being widely applied in network analysis, link prediction can also be used as an inference attack to speculate the sensitive links in graphs. Therefore, the data publisher shall carry out privacy processing for the published data to defend link prediction attacks while retaining necessary data utility. In recent years, the privacy disclosure caused by link prediction attacks has attracted researchers' attention, and many researches on antilink prediction have emerged. To defend link prediction based on similarity and deep learning methods, most antilink prediction methods adopt various link disturbances, e.g., random link disturbance, heuristic link disturbance, and evolutionary link disturbance, at the expense of part of data utility [6–13]. Besides, these methods only focus on the graph structure information and fail to consider the unstructured information in graphs, such as node attributes. The node attributes may include the performance, identity, and type of devices, deepening the association strength between nodes and making the attacker's prediction more accurate. As mentioned above, protecting sensitive links against link prediction attacks is an urgent problem to be solved. Significantly, Li et al. [14] proposed an adversarial privacy graph embedding (APGE) method to conceal users' sensitive attributes from inference attacks, which opens up a novel idea for our work. In this paper, we intend to fill this blank by developing a graph embedding-based sensitive link protection method named SLPGE. Our basic idea is to use the graph embedding model combined with Variational Graph Autoencoder (VGAE) and Adversarially Regularized Variational Graph Autoencoder (ARVGA) to encode graph data into an embedding matrix before publishing the data. To be concrete, we utilize adversarial training assisted by two schemes to eliminate private information in the embedding matrix. Then, to balance the tradeoff between privacy and utility, we design the loss functions in SLPGE to retain the utility of graph structure and node labels. The main contributions of this paper are summarized below:

(i) This article focuses on the privacy protection of sensitive links in IoT and proposes a sensitive link protection method (SLPGE) to conceal sensitive links from link prediction attacks

(ii) The results of experiments on two public datasets with node attributes validate that our SLPGE can reduce the prediction accuracy of attack models for sensitive links by 30.05% and 15.03% at most on the basis of the original data

(iii) Our method achieves a tradeoff between privacy and utility. Different from the previous method, our method abandons the idea of directly applying link disturbance on the original graph to remove private information, for which we reduce the loss of utility

The rest of the paper is organized as follows. The related work and preliminaries are reviewed in Sections 2 and 3, respectively. The system models and problem formulation are presented in Section 4. The details of our SLPGE are described in Section 5. The simulation and results are shown in Section 6. Moreover, we give the conclusions and future work in Section 7.

2. Related Work

The emergence of various IoT platforms not only facilitates people's lives but also generates a huge volume of data-carrying personal information. These data can be modeled into graph structure data, and attackers can then easily expose the privacy information hidden in graphs via link prediction. In this section, we briefly introduce the relevant work of graph privacy protection, link prediction, and antilink prediction.

2.1. Graph Privacy Protection. The main methods of graph privacy protection include anonymization, random disturbance, and clustering. Since Sweeney [15] introduced anonymization into graph structure data, different anonymization variants for graphs have also been derived. Ying and Wu [16] disrupted the graph structure by deleting and adding k edges randomly. Li et al. [17] performed spectral clustering according to the distance between nodes firstly and then anonymized subgraphs. For the graphs with node labels, Yuan et al. [18] proposed the protection method of node attribute label K-anonymity to ensure that the labels of at least k nodes are the same. Chester and Srivastava [19] proposed an attribute probability distribution anonymity method to make the probability distribution of the label carried by each node in the attribute sets of its neighbors as close as possible to the global label probability distribution. The random graph modification technology proposed by Hay et al. [20] is the simplest technology to prevent node reidentification and edge exposure. Mittal et al. [9] proposed a link perturbation based on the random walk (LPRW), which improved the privacy and utility of data compared with Hay's method. In edge clustering methods, Liu et al. [21] proposed privacy protection methods for sensitive edge weights in weighted graphs, adopting Gaussian noise disturbance and greedy disturbance. Zheleva and Getoor [22]

mainly considered the privacy of graphs with multiple types of edges and one type of node. Its main idea is to divide the original graph into subgraphs via spectral clustering and then modify the links in the subgraphs and add new links between the subgraphs randomly.

Low data availability and high computational complexity are the common problems of these methods, and their privacy will continue to decrease as inference attacks intensify.

2.2. Link Prediction. Link prediction is aimed at predicting missing facts according to existing entities and has found wide application in social, biological, and communication networks. Known for its powerful inference attack, link prediction has been maliciously used to spy on the privacy of entities in the networks. Among plenty of link prediction methods, classification models such as support vector machine (SVM) [23], multilayer perceptron (MLP) [24], and k nearest neighbor (KNN) [25] regard link prediction as a binary classification problem, in which the connected node pairs and unconnected node pairs are regarded as positive samples and negative samples, respectively.

2.3. Antilink Prediction. At present, most antilink prediction methods for graph structure data disturb the graph structure by adding some new links and deleting part of nonsensitive links strategically to reduce the prediction ability of various link prediction methods and achieve the privacy protection of sensitive links. Liu and Terzi [6] proposed to achieve k-degree anonymization through edge addition or deletion strategies. Rousseau et al. [7] proposed two approaches that preserve the coreness of a graph while anonymizing it through various edge modification operations. Fard and Wang [8] and Mittal et al. [9] proposed two structure-aware randomization perturbation methods based on local perturbation and random walk considering the structural proximity of nodes. Zhou et al. [10] regarded the links between the end nodes of a sensitive link and their common nodes as the candidate links to be deleted and expressed the attack on local similarity as an optimization problem to determine which links to delete. Chen et al. [11] proposed an iterative gradient attack (IGA) method based on integral gradient information in Graph Autoencoder (GAE). The gradients obtained by maximizing the loss of sensitive links represent the influence of other links on sensitive links. During k iterations, n links with the largest gradients are modified. Yu et al. [12] combated resource allocation (RA) indicator link prediction via random, heuristic, and evolutionary link disturbance. Among these three methods, random link disturbance increases and changes links without any strategy, heuristic link disturbance reduces the link prediction ranking of node pairs in the test set, and evolutionary link disturbance selects the links to be added and deleted according to the fitness function. Waniek et al. [13] selected to delete or add the most influential links to hide sensitive links by reducing or creating the closed triangles containing sensitive links.

The methods mentioned above can be used in IoT systems to avoid the leakage of sensitive links in data transactions. However, two shortcomings are present in the above methods: the first is that the utility of the graph will be lost due to link disturbance, and the second is that they lack the consideration of the impact of node attributes on link prediction.

3. Preliminaries

As a kind of non-Euclidean data, a graph is difficult to be directly processed by traditional data analysis methods or deep learning models such as Convolutional Neural Network (CNN) [26] and Recurrent Neural Network (RNN) [27] due to the high computational and space overhead. Graph embedding, also called network representation learning, is aimed at mapping graph data, usually a high-dimensional dense matrix to low-dimensional dense vectors. Graph embedding has more flexible and rich calculation methods to apply deep learning models directly for graph analysis tasks. Graph Neural Network (GNN) represents the deep learning method of graph embedding. By modeling the nodes and communication links in the networks, GNN can be applied to solve the privacy disclosure problem in IoT. For the advantages of feature extraction from non-Euclidean data, our SLPGE is based on some GNN models. In this section, the GNN models involved in SLPGE, e.g., Graph Convolutional Network (GCN), VGAE, and ARVGA, are briefly introduced. For the sake of clarity, the frequently used notations and their meanings are listed in Table 1.

3.1. Graph Convolutional Network. In 2013, Bruna et al. [28] first proposed the neural network on the graph and gave two structures based upon a hierarchical clustering of the domain and the spectrum of the graph Laplacian. As a typical GNN model, GCN [29] is a scalable approach for semisupervised learning on graph data, which uses the spectrum of the graph Laplacian to achieve convolution on graphs. After each convolution of GCN, the node features are the weighted sum of the previous features of the nodes and their neighbor nodes, for which the nodes can aggregate further features with the deepening of layers. Hence, the superiority of GCN is to incorporate local graph structure and node features naturally. Suppose the adjacency matrix $A \in \mathbb{R}^{N\times N}$ represents the connection relationship between n nodes, then the layer-wise propagation rule of GCN is as follows:

$$H_{l+1} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H_l W_l\right), \tag{1}$$

where $\mathbf{H}_l$ is the feature matrix of the lth layer, $\mathbf{W}_l$ is the trainable weight matrix, and $\sigma(\cdot)$ is an activation function. $\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ is the normalization of $\tilde{A}$ where $\tilde{A} = \mathbf{A} + \mathbf{I}_N$, $\mathbf{I}_N \in \mathbb{R}^{N\times N}$ is the identity matrix, $\tilde{D}$ is the degree matrix of $\tilde{A}$, and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. The degree of a node is the number of first-order neighbors connected to the node. Equation (1) can be abbreviated as $\mathbf{H}_{l+1} = f(\mathbf{H}_l, \mathbf{A})$, for $\mathbf{A}$ is the input of each layer.

Table 1: Summary of notations.

Notations	Meanings
$\mathbf{G}$	The undirected original graph
$\mathbf{V}$	The set of nodes in $\mathbf{G}$
$\lvert\mathbf{V}\rvert$	The number of nodes
$\mathbf{E}$	The set of edges in $\mathbf{G}$
$\lvert\mathbf{E}\rvert$	The number of edges
v_i	The ith node
e_{ij}	The edge between v_i and v_j
$\mathbf{X}$	The node feature matrix of $\mathbf{V}$
F	The number of node attributes
$\mathbf{A}$	The adjacency matrix of $\mathbf{G}$
$\mathbf{A}_p$	The adjacency matrix of privacy graph
$\mathbf{A}_t$	The adjacency matrix of training graph
$\widehat{\mathbf{A}}$	The reconstructed adjacency matrix of $\mathbf{G}$
$\widehat{\mathbf{A}}_p$	The reconstructed adjacency matrix of privacy graph
A_{ij}	The link state between v_i and v_j in $\mathbf{A}$
$\widehat{A}_{ij}$	The link state between v_i and v_j in $\widehat{\mathbf{A}}$
L	The number of categories for node labels
$\widehat{\mathbf{y}}$	The node label matrix predicted by softmax classifier with each row includes the predicted values of L categories
$\mathbf{Z}_p$	The privacy embedding of privacy graph
$\mathbf{Z}_f$	The link protection graph embedding
$\mathbf{Z}$	The higher dimensional graph embedding concatenated by $\mathbf{Z}_f$ and $\mathbf{Z}_p$
m	The maximum number of edges added for each sensitive link
$\mathbf{E}_{\text{sl}}$	The sensitive links in $\mathbf{G}$
$\mathbf{E}_{\text{nsl}}$	Part of nonsensitive links in $\mathbf{G}$
$\mathbf{E}_{\text{know}}$	The links which are known to the attack models
L_{link}	The reconstruction loss
L_{lable}	The node classification loss
L_{g}	The distribution loss of the generator
L_{G}	The total loss of the generator
L_{D}	The distribution loss of the discriminator
Acc_{sl}	The classification accuracy of the attack models for sensitive links
Acc_{nsl}	The classification accuracy of the attack models for nonsensitive links
$\text{Acc}_{\text{recon}}$	The link reconstruction accuracy of $\mathbf{Z}_f$
$\text{Re c}_{\text{recon}}$	The link reconstruction recall of $\mathbf{Z}_f$
Acc_{node}	The node classification accuracy of $\mathbf{Z}_f$

3.2. Variational Graph Autoencoders. Soon after the proposal of GCN, to expand the capability of GCN, VGAE proposed by Kipf and Welling [30] adopts GCN as an encoder to generate specific graph embedding for different tasks of the graph, not limited to node classification. VGAE is an unsupervised learning framework derived from Variational Autoencoders (VAE) [31], which obtains graph embedding through the encoder-decoder structure. VGAE consists of a two-layer GCN encoder and a simple inner-product decoder. The two-layer GCN can be defined as follows:

$$\begin{aligned}\text{GCN}(\mathbf{X},\mathbf{A}) &= f(\mathbf{H}_1,\mathbf{A}) = \sigma\left(\bar{\mathbf{A}}f(\mathbf{H}_0,\mathbf{A})\mathbf{W}_1\right)\\ &= \bar{\mathbf{A}}\text{ReLU}\left(\bar{\mathbf{A}}\mathbf{H}_0\mathbf{W}_0\right)\mathbf{W}_1,\end{aligned} \tag{2}$$

where $\bar{\mathbf{A}} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ is the symmetrically normalized adjacency matrix and $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$ is the activation function of the first layer. $\sigma(\cdot)$ of the second layer is determined according to the specific task. The encoder of VGAE is aimed at learning the mean μ and the standard deviation σ of a multi-dimensional Gaussian distribution from which the graph embedding $\mathbf{Z}$ is sampled. The process is briefly described below:

$$\begin{aligned} \mu &= \mathrm{GCN}_{\mu}(\mathbf{X}, \mathbf{A}), \\ \log \sigma &= \mathrm{GCN}_{\sigma}(\mathbf{X}, \mathbf{A}), \\ \mathbf{Z} &= \mu + \varepsilon \times \sigma, \end{aligned} \tag{3}$$

where $\mathbf{X} \in \mathbb{R}^{N\times F}$ replaces $\mathbf{H}_0$ in Equation (2) as the node feature matrix of the first layer and $\mathrm{GCN}_{\mu}(\mathbf{X}, \mathbf{A})$ and $\mathrm{GCN}_{\sigma}(\mathbf{X}, \mathbf{A})$ share first-layer parameters $\mathbf{W}_0$. $\mathbf{Z} \sim \mathcal{N}(\mu, \sigma^2)$ is the graph embedding matrix and $\varepsilon \sim \mathcal{N}(0, 1)$ is the noise sampled from the standard Gaussian distribution. The inner product is used as a decoder in VGAE, and the formula is as follows:

$$\hat{\mathbf{A}} = \sigma\left(\mathbf{Z} \cdot \mathbf{Z}^T\right), \tag{4}$$

where $\sigma(\cdot) = 1/1 + \exp(-\cdot)$ is the sigmoid function. $\hat{\mathbf{A}}$ is the reconstructed adjacency matrix, and $\hat{A}_{ij}$ can be regarded as the product of independent event probabilities of the ith node and the jth node. When $\hat{A}_{ij}$ is greater than the threshold 0.5, it means that there is a link between the ith node and the jth node.

VGAE has two optimization objectives: one is to make $\hat{A}$ and A as similar as possible; the other is to make the distribution of Z as close to the standard Gaussian distribution as possible. Since binary cross-entropy (BCE) can determine the proximity between the actual output and the expected output and Kullback-Leibler (KL) divergence can measure the difference between two distributions, the loss function of VGAE composed of BCE and KL divergence can be expressed as

$$\mathrm{loss} = \mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{A})}[\log p(\mathbf{A} \mid \mathbf{Z})] - \mathrm{KL}[q(\mathbf{Z} \mid \mathbf{X}, \mathbf{A}) \| p(\mathbf{Z})]. \tag{5}$$

Here, the former minimizes the reconstruction loss through the cross-entropy function, and the latter minimizes the KL divergence. $p(\mathbf{A} \mid \mathbf{Z}) = \sigma(\mathbf{Z} \cdot \mathbf{Z}^T)$, $q(\mathbf{Z} \mid \mathbf{X}, \mathbf{A}) = \prod_{i=1}^{N} q(\mathbf{z_i} \mid \mathbf{X}, \mathbf{A}) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{z_i} \mid \mu_{\mathbf{i}}, \mathrm{diag}(\sigma_{\mathbf{i}}^{\mathbf{2}}))$ is the real distribution function we get, and $p(\mathbf{Z}) = \prod_i p(\mathbf{z_i}) = \prod_i \mathcal{N}(\mathbf{z_i} \mid 0, \mathbf{I})$ is a Gaussian prior. $\mathrm{KL}[q(\cdot) \| p(\cdot)]$ is the KL divergence between $q(\cdot)$ and $p(\cdot)$. We expect $q(\mathbf{Z} \mid \mathbf{X}, \mathbf{A})$ to be as close to $p(\mathbf{Z})$ as possible.

More specifically, $\mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{A})}[\log p(\mathbf{A} \mid \mathbf{Z})]$ in Equation (5) can be abbreviated as $\mathrm{loss}_{\mathrm{link}}$ below:

$$\mathrm{loss}_{\mathrm{link}} = -\frac{1}{|\mathbf{V}|^2} \sum_{i\in\mathbf{V}} \sum_{j\in\mathbf{V}} \left(p_1 A_{ij} \log \hat{A}_{ij} + \left(1 - A_{ij}\right) \log\left(1 - \hat{A}_{ij}\right)\right), \tag{6}$$

where A_{ij} represents the value which is 0 or 1 of an element in A, $\hat{A}_{ij}$ represents the probability value of the corresponding element in $\hat{A}$, and p_1 is the ratio of the number of 0 to 1 in A, which can be used to solve the problem of imbalance between positive and negative samples. $\mathrm{KL}[q(\mathbf{Z} \mid \mathbf{X}, \mathbf{A}) \| p(\mathbf{Z})]$ in Equation (5) can be abbreviated as $\mathrm{loss}_{\mathrm{dist}}$ below:

$$\mathrm{loss}_{\mathrm{dist}} = -\frac{1}{2}\left(1 + \log \sigma^2 - \mu^2 - \sigma^2\right). \tag{7}$$

3.3. Adversarially Regularized Variational Graph Autoencoder. To force the graph embedding learned by VGAE to fit the prior distribution better, Pan et al. [32] proposed ARVGA by combining VGAE and Generative Adversarial Network (GAN). GAN was first proposed by Goodfellow et al. [33] to serve as a generative model bridging supervised learning and unsupervised learning in 2014. Most recently, exploiting GAN to work out elegant solutions to severe privacy and security problems has become increasingly popular in both academia and industry due to its game theoretic optimization strategy [34]. Typically, GAN consists of a generator G and a discriminator D, the purpose of which is to mix the spurious with the genuine in a nutshell. During the iterative training, G is trained to generate the fake samples to convince D that the fake samples come from a prior data distribution, while D discriminates whether an input sample comes from the prior data distribution or G we built. In ARVGA, we take VGAE as G, a two-layer fully connected network as D where the output layer only has one dimension with a sigmoid function. The equation for training the encoder model with the discriminator can be written as follows:

$$\min_{G} \max_{D} \left(\mathbb{E}_{x\sim p_{\mathrm{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]\right). \tag{8}$$

Here, $x \sim P_{\mathrm{data}}(x)$ is the real sample, $z \sim P_z(z)$ is the original data, $G(z)$ is the fake sample, and $D(\cdot)$ is the probability that the sample is true. G is aimed at minimizing the equation while D is aimed at the opposite of G. Through the game between G and D, ARVGA can enforce the graph embedding to match the prior distribution and produce a robust representation.

4. Model and Problem Formulation

In this article, our work is based on the following assumptions in the graph of IoT: The connections between devices are bidirectional. There are L types of devices in the graph, and each device has its own attribute information such as internal storage, bandwidth, and hard disk. Sensitive links are the links that need to be hidden, while nonsensitive links are those which can be made public. The links whose end nodes have a larger total degree are defined as sensitive links. The nodes with larger degrees usually have more influence in the graph, so the links between these nodes are also more meaningful. Moreover, we take SVM and MLP as attack models to test the performance of our method, and part of nonsensitive links and nonexistent links in the graph are known to the attack models.

4.1. Network Model. We express one of the graphs of IoT as an undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{X})$. $\mathbf{V} = \{v_1, v_2, \cdots, v_N\}$ is the set of n terminal nodes and $N = |\mathbf{V}|$. $\mathbf{E}$ contains the edges e_{ij} with the communication link between v_i and $v_j (1 \le i, j \le N)$, including sensitive links and nonsensitive links. $\bar{\mathbf{E}}$ is the set of nonexistent links and $\mathbf{E} \cup \bar{\mathbf{E}} = \mathbf{E}_{N^2}$, where $\mathbf{E}_{N^2}$ contains $n \times n$ edges that can be connected by n nodes. Node attributes are summarized in a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$ with the ith row representing the attributes of v_i and F is the number of attributes. $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix, where $A_{ij} = 1$ if $e_{ij} \in \mathbf{E}$; otherwise, $A_{ij} = 0$. $\mathbf{E}_{\text{sl}} \subset \mathbf{E}$ is the set of sensitive links, $\mathbf{E}_{\text{nsl}} \subset \mathbf{E}$ is the set of nonsensitive links, and $E_{\text{sl}} \cap E_{\text{nsl}} = \varnothing$.

4.2. Attack Model. Both SVM and MLP have strong classification abilities for nonlinear problems with different structures.

SVM is a classification model based on the structural risk minimization criterion in machine learning. For the nonlinear classification problems, SVM adopts a nonlinear function $\phi(x)$ to map the samples from the input space to a high-dimensional feature space where the samples are linearly separable and construct an optimal classification hyperplane to categorize new samples utilizing labeled training data. Given the training set $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_k, y_k)\} (x_i \in \mathbb{R}^N)$, SVM can transform the classification problem into a convex quadratic optimization problem as follows:

$$\begin{cases} \min\limits_{\alpha} \dfrac{1}{2} \displaystyle\sum_{i=1}^{k} \sum_{j=1}^{k} \alpha_i \alpha_j y_i y_j \phi(x_i) \phi(x_j) - \sum_{i=1}^{k} \alpha_i \\ \text{s.t.} \displaystyle\sum_{i=1}^{k} \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C, i = 1, 2, \cdots, k, \end{cases} \tag{9}$$

where α_i is a Lagrange multiplier and C is the penalty factor. Since the computation of $\phi(x_i) \times \phi(x_j)$ increases sharply in the high-dimensional space, SVM introduces kernel function $K(x_i, x) = \phi(x_i) \cdot \phi(x)$ to avoid the problem. The kernel function we choose is Gaussian kernel:

$$K(x_i, x) = \exp\left(-\frac{\|x_i - x\|^2}{2\sigma^2}\right), \tag{10}$$

where σ^2 is the variance. In this case, the classification decision function is as follows:

$$f(x) = \text{sign}\left(\sum_{i=1}^{k} \alpha_i y_i \exp\left(-\frac{\|x_i - x\|^2}{2\sigma^2}\right) + b\right), \tag{11}$$

where b is the bias constant.

MLP is a fully connected artificial neural network, consisting of an input layer, hidden layer, and output layer. MLP adjusts the parameters in the hidden layer units through the supervised back propagation (BP) algorithm and gradient descent algorithm to reduce the error between the actual output and the expected output. The forward propagation mechanism of MLP is expressed as below:

$$\mathbf{H}^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \mathbf{H}^{(l)} + \mathbf{b}^{(l)}\right), \tag{12}$$

where $\mathbf{H}^{(l)}$ is the input matrix, $\mathbf{W}^{(l)}$ is the weight matrix, $\mathbf{b}^{(l)}$ is the bias, and $\mathbf{H}^{(l+1)}$ is the output of the hidden layer. Thus, the decision function of MLP with only one hidden layer can be expressed as follows:

$$f(x) = \sigma\left(\mathbf{W}^{(1)}\left(\sigma\left(\mathbf{W}^{(0)}\mathbf{x} + \mathbf{b}^{(0)}\right)\right) + \mathbf{b}^{(1)}\right), \tag{13}$$

where $\mathbf{x}$ is the input and $\sigma(\cdot)$ is an activation function.

4.3. Problem Formulation. Given a graph $\mathbf{G}$, our model will compress it into a graph embedding $\mathbf{Emb}$ where the ith row represents the vector emb_i of v_i. The vector of e_{ij} can be expressed as $\text{emb}_{ij} = (\text{emb}_i, \text{emb}_j)$. Suppose a link set $\mathbf{E}_{\text{know}}$ containing k nonsensitive links (class 1) and k nonexistent links (class 0) in $\mathbf{G}$ have been exposed to attackers. Then, the embedding matrix of $\mathbf{E}_{\text{sl}}$ and $\mathbf{E}_{\text{know}}$ are $\mathbf{Emb}_{\mathbf{E}_{\text{sl}}}$ and $\mathbf{Emb}_{\mathbf{E}_{\text{know}}}$ where each row represents an edge embedding vector.

During data transactions, attackers will collect or steal $\mathbf{Emb}$ by any means to infer sensitive links through link prediction. Our goal is to achieve the balance between privacy protection and data utility. To this end, we use "minmax" strategy to maximize the distance between the predicted label $\text{label}_{\text{pred}}$ of sensitive links and its real label $\text{label}_{\text{real}}$ and then minimize the distance between $\text{label}_{\text{pred}}$ of nonsensitive links and its $\text{label}_{\text{real}}$. The mathematical description is as follows:

$$\begin{aligned} &\text{Training}: \text{Clf\_fit}\left(\mathbf{Emb}_{\mathbf{E}_{\text{know}}}, \text{label}_{\mathbf{E}_{\text{know}}}\right), \\ &\text{Prediction}: \text{label}_{\text{pred}} = \text{Clf}_{\text{predict}}\left(\mathbf{Emb}_{\mathbf{E}_{\text{sl}}}\right), \\ &\text{Objective}: \min_{E_{\text{nsl}}} \max_{E_{\text{sl}}} \left|\left|\text{label}_{\text{pred}} - \text{label}_{\text{real}}\right|\right|^2, \end{aligned} \tag{14}$$

where $\text{Clf\_fit}(x_{\text{train}}, y_{\text{train}})$ means to fit the classifier model with the training data and $\text{Clf\_predict}(x_{\text{test}})$ means to predict the labels of x_{test}. $\text{label}_{\mathbf{E}_{\text{know}}}$ is the label set of $\mathbf{E}_{\text{know}}$ where k ones represent nonsensitive links and k zeros represent nonexistent links. We expect to get a graph embedding which can work for our purpose.

5. Algorithm

The SLPGE framework consists of two parts. In this section, we will introduce the framework of SLPGE in Subsections 5.1 and 5.2, and the evaluation indicators are described in Subsection 5.3.

5.1. Generate the Privacy Embedding $\mathbf{Z}_p$. Part 1 is to generate a privacy embedding $\mathbf{Z}_p$. In order to put more privacy

Input: $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{X})$: the original graph
$\mathbf{A}$: the adjacency matrix of $\mathbf{G}$
$\mathbf{E}_{\text{sl}}$: the set of sensitive links
$\mathbf{N}_i$: the neighbor nodes set of v_i
m: the maximum number of edges added for each sensitive link
Output: $\mathbf{A}_p$: the adjacency matrix of privacy graph
for $\text{sl}_{ij} \in \mathbf{E}_{\text{sl}}$ **do**
  find the neighbor nodes sets $\mathbf{N}_i$ and $\mathbf{N}_j$
  for $n_i \in \mathbf{N}_i, n_j \in \mathbf{N}_j$ **do**
    if $A_{n_i n_j} = 1$ and $A_{n_i v_j} = 0 (A_{n_j v_i} = 0)$ **then**
      if the number of edges added for $sl_{ij} \leq m$ **then**
        $A_{n_i v_j} = 1 (A_{n_j v_i} = 1)$
      end if
    end if
  end for
end for
return take the modified $\mathbf{A}$ as $\mathbf{A}_p$

Algorithm 1: Generate privacy graph by adding edges.

information into $\mathbf{Z}_p$, we first change the structure of the original graph $\mathbf{G}$ to enhance the connection strength of end nodes of sensitive links. Before inputting the graph data into the model, we preprocess $\mathbf{G}$ through Algorithms 1 and 2 corresponding to Figures 1 and 2. For Algorithm 1, we believe that two nodes with more common neighbors have a closer relationship. As shown in Figure 1, e_{01} is a sensitive link, $\{v_2, v_4\}$ and $\{v_3, v_5\}$ are the neighbor sets of v_0 and v_1, respectively, and e_{23} exists; in this case, we link e_{03} and e_{12} to make the relationship between v_0 and v_1 closer. For Algorithm 2, we believe that the main information in the graph will focus on sensitive links when other irrelevant nodes and links are removed. As shown in Figure 2, we only keep the sensitive links and their adjacent links to retain the information about the sensitive links to the greatest extent. The two privacy graph adjacency matrices $\mathbf{A}_p$'s obtained by Algorithms 1 and 2 are, respectively, used as the input of the encoder to output two $\mathbf{Z}_p$'s.

For VGAE is more robust and suitable for small graphs, we adopt VGAE to obtain $\mathbf{Z}_p$ in this part as shown in Figure 3. As discussed in Section 3, the mechanism for the encoder to generate $\mathbf{Z}_p$ can refer to Equations (2) and (3). Then, we get the reconstructed adjacency matrix $\widehat{\mathbf{A}}_p = \text{sigmoid}(\mathbf{Z}_p \cdot \mathbf{Z}_p{}^T)$. The reconstruction loss L_{link} is the same as Equation (5), except that $\mathbf{A}$ and $\widehat{\mathbf{A}}$ are replaced by $\mathbf{A}_p$ and $\widehat{\mathbf{A}}_p$.

For node classification, a softmax classifier is followed by the encoder to predict the labels of the nodes. The node classification loss function L_{label} is as follows:

$$L_{\text{label}} = -\sum_{i=1}^{|\mathbf{V}|} \sum_{l=1}^{L} y_{il} \ln \widehat{y}_{il}, \tag{15}$$

where y_{il} represents the real label of v_i in category l with a value of 0 or 1, while $\widehat{y}_{il}$ is the value we predict in $\widehat{\mathbf{y}}$ and $\widehat{y}_{il}$

Input: $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{X})$: the original graph
$\mathbf{A}$: the adjacency matrix of $\mathbf{G}$
$\mathbf{E}_{\text{sl}}$: the set of sensitive links
$\mathbf{V}_{\text{sl}}$: the end nodes set of $\mathbf{E}_{\text{sl}}$
$\mathbf{A}_p$: the empty matrix with the same shape as $\mathbf{A}$
$\mathbf{N}$: the neighbor nodes set
Output: $\mathbf{A}_p$: the adjacency matrix of privacy graph
for $v \in \mathbf{V}_{\text{sl}}$ **do**
  find the neighbor nodes set $\mathbf{N}$ of v
  for $n \in \mathbf{N}$ **do**
    $A_p vn = 1 (A_p nv = 1)$
  end for
end for
return $\mathbf{A}_p$

Algorithm 2: Generate privacy graph by deleting edges.

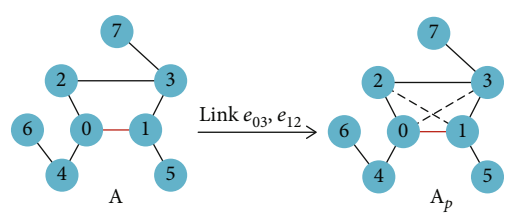


Figure 1: Generate the privacy graph with edge added.

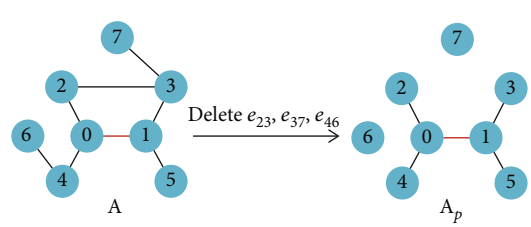


Figure 2: Generate the privacy graph with edge deleted.

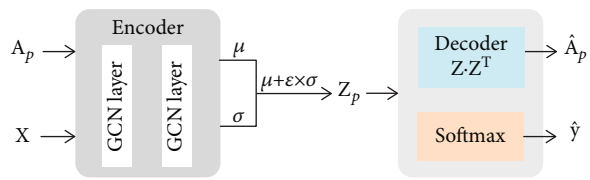


FIGURE 3: Generate the privacy graph embedding $\mathbf{Z}_p$.

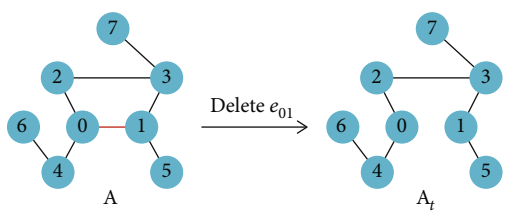


FIGURE 4: Generate the training graph.

$= \text{softmax}((\mathbf{Z}_p)_{il}) = 1/\mathscr{Z} \exp((\mathbf{Z}_p)_{il})$ with $\mathscr{Z} = \sum_{l=1}^{L} e^{(\mathbf{Z}_p)_{il}}$. Therefore, the total loss of Part 1 is as follows:

$$L_1 = L_{\text{link}} + L_{\text{label}}. \tag{16}$$

Through the BP mechanism of L_1, we train the encoder to generate $\mathbf{Z}_p$ that contains privacy information and conforms to a prior distribution.

5.2. Generate the Link Protection Graph Embedding $\mathbf{Z}_f$. Part 2 generates a graph embedding $\mathbf{Z}_f$ that can protect sensitive links. In order to reduce the most intuitive privacy information, we remove the sensitive links in $\mathbf{A}$ to obtain $\mathbf{A}_t$, the adjacency matrix of the training graph, as shown in Figure 4. The model in this part is designed based on ARVGA, as shown in Figure 5. The inputs of the encoder are $\mathbf{A}_t$ and $\mathbf{X}$. $\mathbf{Z}_f$ output from the encoder is the input of the discriminator and the softmax classifier. Unlike Part 1, $\mathbf{Z}_f$ and $\mathbf{Z}_p$ are combined by adding or concatenating to form a higher dimensional matrix $\mathbf{Z}$ as the input of the inner-product decoder. The node classification loss function L_{label} is the same as Equation (15). In order to distinguish the two $\mathbf{Z}_p$'s obtained in Part 1, in Section 6.2, we will use **SLP GE**$^+$ to explain that Algorithm 1 is used for the generation of $\mathbf{Z}_p$ and **SLPGE**$^-$ to explain that Algorithm 2 is used.

Since the adversarial training between the encoder and the discriminator can force $\mathbf{Z}_f$ to match a prior distribution, the KL divergence in Equation (5) is omitted. Here, we try to use Mean Squared Error (MSE) as the reconstruction loss L_{link}, and the reconstruction target is changed to $\mathbf{A}$:

$$L_{\text{link}} = -\frac{1}{|\mathbf{V}|}\sum_{i\in\mathbf{V}}\sum_{j\in\mathbf{V}}\left\|A_{ij} - \hat{A}_{ij}\right\|^2. \tag{17}$$

In the discriminator, we take $\mathbf{Z}_f$ as the fake samples and Gaussian distribution samples as the real samples, then input them into the discriminator, i.e., a two-layer full connection layer network to get two estimated value d_{fake} and d_{real}, respectively. L_{g} and L_{D} are the distribution loss of the generator and the discriminator, which are both calculated by BCE:

$$L_{\text{g}} = -\log(d_{\text{fake}}), \tag{18}$$

$$L_{\text{D}} = -\log(d_{\text{real}}) - \log(1 - d_{\text{fake}}). \tag{19}$$

Therefore, the total loss of the generator can be written as follows:

$$L_{\text{G}} = L_{\text{link}} + L_{\text{label}} + L_{\text{g}}. \tag{20}$$

Through the adversarial training, the discriminator learns how to distinguish between the real samples and the fake samples, while the generator learns to generate a better $\mathbf{Z}_f$ to confuse the discriminator. In general, the training process of obtaining $\mathbf{Z}_f$ can be summarized as Algorithm 3.

5.3. Evaluation Indicators. This subsection will introduce the quantitative indicators of privacy and utility.

5.3.1. Privacy. Our chief target is to reduce the prediction accuracy of the attack models for sensitive links. Input an embedding vector of a sensitive or nonsensitive link to the attack models; if the predicted value is 1, it means the link exists and vice versa. Privacy is measured by the prediction accuracy Acc_{sl} of the attack models for sensitive links:

$$\text{Acc}_{\text{sl}} = \frac{N_{\text{sl}}}{|\mathbf{E}_{\text{sl}}|} \times 100\%, \tag{21}$$

where N_{sl} is the number of sensitive links predicted to exist and $|\mathbf{E}_{\text{sl}}|$ is the total number of sensitive links. When the security of $\mathbf{Z}_f$ is stronger, Acc_{sl} is lower.

5.3.2. Utility. Utility includes three parts: the prediction accuracy of the attack models for nonsensitive links, the accuracy and recall of the reconstructed graph, and the accuracy of node classification. Taking the existing links as positive samples and the nonexistent links as negative samples, the quantitative expression of utility is as follows:

$$\text{Acc}_{\text{nsl}} = \frac{N_{\text{nsl}}}{|\mathbf{E}_{\text{nsl}}|} \times 100\%, \tag{22}$$

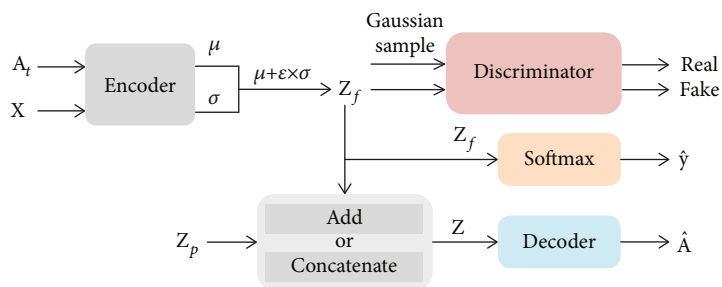


Figure 5: Generate the link protection graph embedding $\mathbf{Z}_f$.

Input: $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{X})$: the original graph
$\mathbf{A}_t$: the adjacency matrix of training graph
$\mathbf{Z}_p$: the privacy embedding
Output: $\mathbf{Z}_f$: the link protection graph embedding
for each epoch **do**
Generate the adjacency matrix $\mathbf{A}_t$ of training graph
Input $\mathbf{A}_t$ and $\mathbf{X}$ to the encoder to generate $\mathbf{Z}_f$
Input $\mathbf{Z}_f$ to the softmax classifier
Adding or concatenating $\mathbf{Z}_f$ and $\mathbf{Z}_p$ to form $\mathbf{Z}$
Input $\mathbf{Z}$ to the inner-product decoder
Input $\mathbf{Z}_f$ and the Gaussian samples to the discriminator
Update the generator by minimizing L_G
Update the discriminator by minimizing L_D
end for
return$\mathbf{Z}_f$

Algorithm 3: Generate link protection graph embedding.

where $\mathrm{Acc}_{\mathrm{nsl}}$ is the prediction accuracy of nonsensitive links, N_{nsl} is the number of nonsensitive links predicted to exist, and $|\mathbf{E}_{\mathrm{nsl}}|$ is the total number of nonsensitive links:

$$\mathrm{Acc}_{\mathrm{recon}} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \times 100\%, \tag{23}$$

$$\mathrm{Re\ c}_{\mathrm{recon}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \times 100\%, \tag{24}$$

where $\mathrm{Acc}_{\mathrm{recon}}$ is the ratio of existing links and nonexistent links that are reconstructed correctly and $\mathrm{Re\ c}_{\mathrm{recon}}$ represents how many existing links have been reconstructed. TP and FP are the numbers of reconstructed positive and negative samples, and FN and TN are the numbers of nonreconstructed positive and negative samples:

$$\mathrm{Acc}_{\mathrm{node}} = \frac{N_{\mathrm{node}}}{|\mathbf{V}|} \times 100\%, \tag{25}$$

where $\mathrm{Acc}_{\mathrm{node}}$ is the ratio of the nodes classified correctly to the total number of nodes. N_{node} is the number of nodes classified correctly, and $|\mathbf{V}|$ is the number of nodes. Our tradeoff is protecting privacy while preserving utility, that is, reducing $\mathrm{Acc}_{\mathrm{sl}}$ and keeping $\mathrm{Acc}_{\mathrm{nsl}}$, $\mathrm{Acc}_{\mathrm{recon}}$, $\mathrm{Re\ c}_{\mathrm{recon}}$, and $\mathrm{Acc}_{\mathrm{node}}$ high.

Table 2: Details of experiment.

Parameters	Cora	Yale
$\lvert\mathbf{V}\rvert$	2708	5278
$\lvert\mathbf{E}\rvert$	5278	405450
F	1433	188
L	7	7
$\lvert\mathbf{E}_{\mathrm{sl}}\rvert$	100	200
$\lvert\mathbf{E}_{\mathrm{nsl}}\rvert$	100	200
$\lvert\mathbf{E}_{\mathrm{know}}\rvert$	400	400
m	10	15
$\mathbf{Z}_p$	8-dim	7-dim
$\mathbf{Z}_f$	8-dim	7-dim
$\mathbf{Z}$(add)	8-dim	7-dim
$\mathbf{Z}$(cat)	16-dim	14-dim

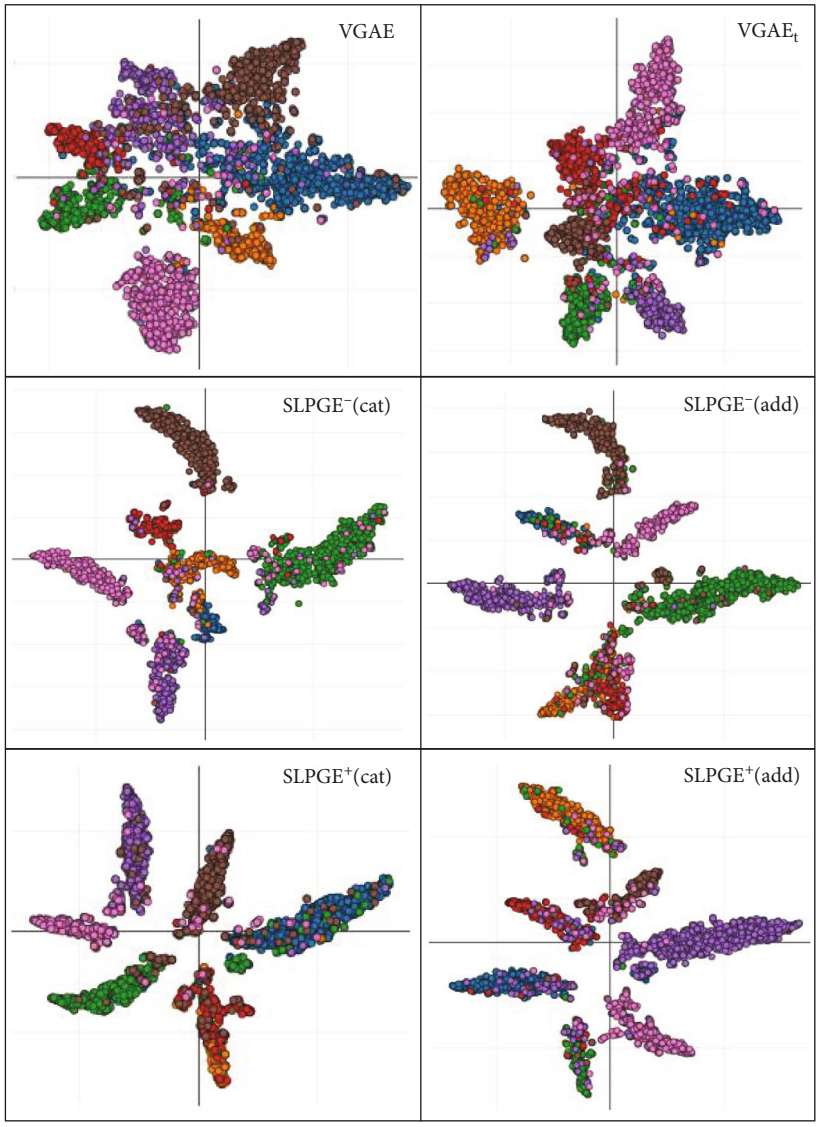


Figure 6: Visualization of node classification on Cora.

6. Simulation

In this section, we will evaluate the performance of SLPGE on two public datasets, Cora [35] and Yale [14].

6.1. Experiment Setting

(1) *Datasets*. Cora is a citation network composed of 7 categories of machine learning papers. Cora includes 2708 papers as **V** and 5278 citation relationships between papers as **E**. 1433 unique words appear in all papers as the attributes of **V**. Yale is a social network including 8578 people and 188 attributes. The class year attribute divides the nodes into 7 categories. Part of links and labels of the datasets are used as training sets.

(2) *Training*. The experimental parameters are shown in Table 2. The initial features of nodes are 1433 and 188 dimensions. $\mathbf{Z}_f$ and $\mathbf{Z}_p$ are both 8-dim in Cora and 7-dim in Yale. As shown in Figure 5, we have two splicing modes of $\mathbf{Z}_f$ and $\mathbf{Z}_p$ in SLPGE: "concatenate (cat)" and "add," where "cat" means stacking $\mathbf{Z}_f$ and $\mathbf{Z}_p$ in the horizontal direction (i.e., column order) and "add" means that the elements in $\mathbf{Z}_f$ and $\mathbf{Z}_p$ are added correspondingly. $\mathbf{Z}$ is 16-dim and 14-dim when using "cat" and 8-dim and 7-dim when using "add." The embeddings of two nodes in $\mathbf{Z}_f$ are concatenated together as an edge embedding, so the dimension of edge embedding is twice as large as node embedding.

Besides, we take the original graph **G** and the training graph $\mathbf{G}_t$ with sensitive links deleted as the input of VGAE to compare with SLPGE. At the same time, we use TSNE to visualize $\mathbf{Z}_f$ in 2-dim to observe the node classification result, and the nodes belonging to the same label are

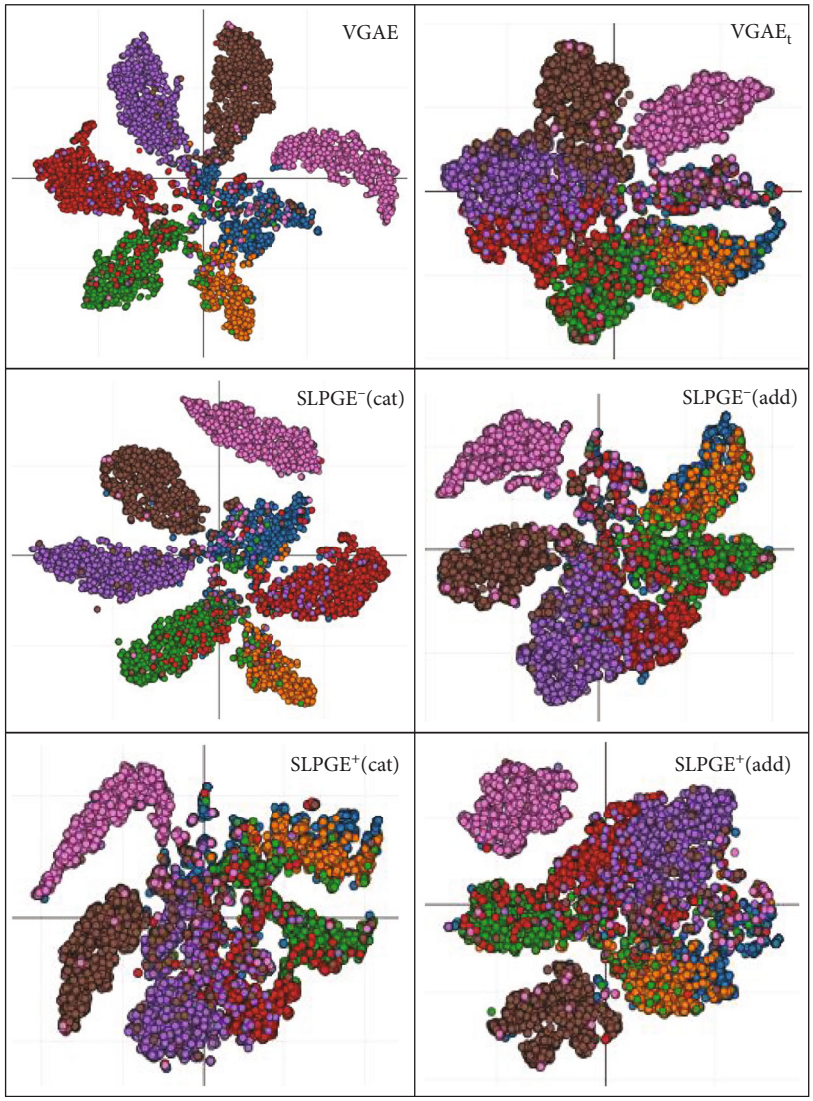


Figure 7: Visualization of node classification on Yale.

represented by the same color. In essence, TSNE uses PCA to reduce the dimension of the feature and then maps it to a 2-dimensional or 3-dimensional space for visualization to observe each layer's feature distribution.

(3) *Attack*. 100 and 200 edges with larger node degrees in the training sets are selected as the sensitive links of Cora and Yale, respectively, and m is 10 and 15 in Algorithm 1. We randomly select 200 nonsensitive links and 200 nonexistent links to form $\mathbf{E}_{\text{know}}$ which has been exposed to the attackers. Moreover, the edge embeddings of $\mathbf{E}_{\text{know}}$ will be used as the training set to train the attack models. The edge embeddings of the same number of sensitive and nonsensitive links are the input of the attack models. We train each model four times, and the attack models make 10 predictions after each training. Finally, the averages of the 40 prediction results are taken as the prediction accuracy of sensitive and nonsensitive links.

6.2. Result Analysis. We carried out our experiments under four models: **VGAE**, **VGAE$_t$**, **SLPGE$^+$**, and **SLPGE$^-$**. **VGAE** means the input is the original graph without any modification. **VGAE$_t$** means the input is the training graph in which the sensitive links are deleted. Our SLPGE is divided into two types: **SLPGE$^+$** and **SLPGE$^-$** where $\mathbf{Z}_p$ comes from Algorithms 1 and 2, respectively.

Figures 6 and 7 show node classification of SVM under different models for Cora and Yale in a visualization method, respectively. In each subgraph, the points in the same color constitute a cluster, representing different classes. A larger distance between different clusters means higher accuracy. Corresponding numerical results are listed in Tables 3 and 4. The decline degree of five indicators of **VGAE$_t$**, **SLPGE$^+$**,

Table 3: The results on Cora.

Model	Splicing mode	SVM		MLP		Reconstruction		
		Acc_{sl}	Acc_{nsl}	Acc_{sl}	Acc_{nsl}	Acc_{recon}	Re c_{recon}	Acc_{node}
VGAE	—	87.5	85.0	84.1	85.0	88.6	90.7	83.5
$VGAE_t$	—	75.2	83.0	80.4	83.5	84.7	89.7	79.9
$SLPGE^-$	cat	66.5	80.1	65.8	79.1	85.6	86.8	76.4
	add	61.6	84.6	61.3	80.5	83.5	85.4	74.1
$SLPGE^+$	cat	68.0	82.0	61.8	79.8	86.6	87.3	74.5
	add	61.2	82.3	60.6	84.4	84.4	87.2	77.2

Table 4: The results on Yale.

Model	Splicing mode	SVM		MLP		Reconstruction		
		Acc_{sl}	Acc_{nsl}	Acc_{sl}	Acc_{nsl}	Acc_{recon}	Re c_{recon}	Acc_{node}
VGAE	—	84.5	84.5	76.5	77.0	72.1	84.5	81.0
$VGAE_t$	—	81.0	83.0	74.0	73.5	73.1	85.7	77.6
$SLPGE^-$	cat	75.0	81.2	65.0	71.0	67.3	79.7	80.1
	add	74.5	84.5	65.2	71.7	65.6	71.4	76.5
$SLPGE^+$	cat	76.5	83.6	65.8	68.1	68.9	75.8	75.6
	add	73.7	82.0	67.5	71.5	68.8	75.1	75.5

Table 5: The decline degree of five indicators compared with VGAE on Cora.

Model	Splicing mode	SVM		MLP		Reconstruction		
		Acc_{sl}	Acc_{nsl}	Acc_{sl}	Acc_{nsl}	Acc_{recon}	Re c_{recon}	Acc_{node}
$VGAE_t$	—	14.05	2.35	4.40	1.76	4.40	1.10	4.31
$SLPGE^-$	cat	24.00	5.76	21.76	6.94	3.39	4.30	8.50
	add	29.60	0.47	27.11	5.29	5.76	5.84	11.26
$SLPGE^+$	cat	22.28	3.52	26.52	6.12	2.26	3.75	10.78
	add	30.05	3.17	27.94	0.71	4.74	3.86	7.54

Table 6: The decline degree of five indicators compared with VGAE on Yale.

Model	Splicing mode	SVM		MLP		Reconstruction		
		Acc_{sl}	Acc_{nsl}	Acc_{sl}	Acc_{nsl}	Acc_{recon}	Re c_{recon}	Acc_{node}
$VGAE_t$	—	4.14	1.78	3.27	4.55	-1.39	-1.42	4.20
$SLPGE^-$	cat	11.24	3.91	15.03	7.79	6.66	5.68	1.11
	add	11.83	0.00	14.77	6.88	9.02	15.50	5.56
$SLPGE^+$	cat	9.47	1.07	13.99	11.56	4.44	10.30	6.67
	add	12.78	2.96	11.76	7.14	4.58	11.12	6.79

and **SLPGE**$^-$ compared with **VGAE** is shown in Tables 5 and 6, and the decline degree are calculated by $|A - B|/B\%$, where B represents **VGAE** and A represents the others.

6.2.1. Privacy. There is a comparison of Acc_{sl} of the four models in Tables 3 and 4 that Acc_{sl} of $\mathbf{VGAE_t}$, **SLPGE**$^+$, and **SLPGE**$^-$ decrease in varying degrees, but Acc_{sl} of **SLPGE**$^+$ and **SLPGE**$^-$ decrease more. Especially for Cora, **SLPGE**$^+$ reduces Acc_{sl} by 30.05% at most and 22.28% at least on the basis of **VGAE** while $\mathbf{VGAE_t}$ reduces Acc_{sl} by 14.05% at most. For Yale, **SLPGE**$^+$ reduces Acc_{sl} by 15.03% at most and 9.46% at least on the basis of **VGAE** while $\mathbf{VGAE_t}$ reduces Acc_{sl} by 4.14% at most. The privacy of SLPGE has significant improvement compared with $\mathbf{VGAE_t}$.

Although the privacy of SLPGE is 1.3 ~ 3.6 times higher than that of $\mathbf{VGAE_t}$ on Yale, the protection effect of sensitive links on Yale is not as good as that on Cora, which results from the fact that the node attributes of Yale are more closely related

to the links. This also signifies that similar attributes will make the privacy information between nodes more difficult to remove. In general, these comparisons can confirm that our SLPGE has better performance on sensitive link protection.

6.2.2. Utility. The loss of partial utility is the necessary cost of privacy protection. Taking **VGAE** as a comparison, we can see that the classification accuracy of four variant models has decreased, reflecting a partial sacrifice of data utility. $\mathrm{Acc}_{\mathrm{nsl}}$, $\mathrm{Acc}_{\mathrm{link}}$, Re c_{recon}, and $\mathrm{Acc}_{\mathrm{node}}$ of SLPGE and $\mathbf{VGAE_t}$ all decrease simultaneously, but the decline ranges are generally lower than that of $\mathrm{Acc}_{\mathrm{sl}}$. From Tables 5 and 6, it can be seen that $\mathrm{Acc}_{\mathrm{nsl}}$ of SLPGE decrease by 6.94% and 11.56% at most on the basis of **VGAE** for Cora and Yale, but the two $\mathrm{Acc}_{\mathrm{sl}}$ decrease more, reaching 21.76% and 13.99%. $\mathrm{Acc}_{\mathrm{link}}$ of SLPGE decrease by 5.75% and 9.07% at most for Cora and Yale. The maximum decline ranges of $\mathrm{Acc}_{\mathrm{link}}$, Re c_{recon}, and $\mathrm{Acc}_{\mathrm{node}}$ of SLPGE on two datasets are 5.76% and 9.02%, 5.84% and 15.50%, and 11.26% and 6.79%, which are basically lower than the decline ranges of $\mathrm{Acc}_{\mathrm{sl}}$. Tables 5 and 6 reflect the tradeoff between privacy and utility.

6.2.3. Models. The data in four tables show that $\mathbf{SLPGE}^{+}$ and $\mathbf{SLPGE}^{-}$ are very close in performance on privacy and utility, which also proves that both Algorithms 1 and 2 are feasible. For the two splicing modes, the privacy and utility of mode "add" are better than those of mode "cat." The analysis of this result is as follows.

The distributions of $\mathbf{Z}_f$ and $\mathbf{Z}_p$ both approach $\mathcal{N}(0,1)$ (standard normal distribution), and the weight of privacy information in $\mathbf{Z}_p$ is large and fixed. When $\mathbf{Z}$ obtained by adding $\mathbf{Z}_f$ and $\mathbf{Z}_p$ is to fit the link labels of the original graph after decoding, the MSE loss function will force $\mathbf{Z}_f$ to reduce the weight of privacy information, so that we can squeeze more privacy information. Therefore, the combination of MSE and mode "add" is better.

Overall, our SLPGE reduces the prediction accuracy of sensitive links to varying degrees, from which we can conclude that our model is effective. While protecting the privacy of sensitive links, some utility will be sacrificed, which may be structure information or attribute information. From the result analysis, it can be confirmed that SLPGE can retain most of the utility. In practical application, part of the structure of the model can be adjusted to meet different task requirements.

7. Conclusion

The problems of individual privacy under the interconnection of all things are ubiquitous. The research on link protection against link prediction in IoT is of great significance for entity privacy. Through the simulation of the datasets, the feasibility of our SLPGE is preliminarily verified. However, multifaceted challenges remain in the research on link protection. Our datasets are just static graphs, in which the nodes belong to different categories at the same level, and the edges only represent reference and social relationships. In heterogeneous scenarios, nodes can be of different levels, edges between the nodes may have diverse meanings, and the weight of the edges are no longer all equal to one. The weight of edges reflects the difference in the degree of communication between nodes.

Furthermore, in dynamic graphs, the entry and exit of nodes will affect the graph structure and the privacy information of sensitive links in real time. The attackers can collect more information for inference attacks. The greatest challenge is that the researches on resisting graph disturbance and enhancing the robustness of link prediction continue to emerge, which increases the difficulty of sensitive link protection. Therefore, we will emphasize the sensitive link protection in weighted graphs and dynamic graphs in our follow-up research.

Appendix

In Section 5.2, we use the two modes of "concatenate (cat)" and "add" to combine $\mathbf{Z}_f$ and $\mathbf{Z}_p$. The following is an explanation of these two operations. "cat" and "add" are two splicing modes of $\mathbf{Z}_f$ and $\mathbf{Z}_p$ in SLPGE. "cat" means stacking $\mathbf{Z}_f$ and $\mathbf{Z}_p$ in the horizontal direction, and "add" means that the elements in $\mathbf{Z}_f$ and $\mathbf{Z}_p$ are added correspondingly. Here, we will intuitively show how to get $\mathbf{Z}(\mathrm{cat})$ and $\mathbf{Z}(\mathrm{add})$ and explain their meanings. We assume that both $\mathbf{Z}_f$ and $\mathbf{Z}_p$ are 3 times 2 matrices,

$$\begin{aligned}\mathbf{Z}_f &= \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \\ c_1 & c_3 \end{bmatrix}, \\ \mathbf{Z}_p &= \begin{bmatrix} a_3 & a_4 \\ b_3 & b_4 \\ c_3 & c_4 \end{bmatrix},\end{aligned} \tag{A.1}$$

then

$$\begin{aligned}\mathbf{Z}(\mathrm{cat}) &= \left[\mathbf{Z}_f \mathbf{Z}_p\right] = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \end{bmatrix}, \\ \mathbf{Z}(\mathrm{add}) &= \mathbf{Z}_f + \mathbf{Z}_p = \begin{bmatrix} a_1 + a_3 & a_2 + a_4 \\ b_1 + b_3 & b_2 + b_4 \\ c_1 + c_3 & c_2 + c_4 \end{bmatrix}.\end{aligned} \tag{A.2}$$

In Part II, the reconstructed adjacency matrix $\widehat{\mathbf{A}}$ is obtained by the inner product of $\mathbf{Z}(\mathbf{Z}(\mathrm{cat})\,\mathrm{or}\,\mathbf{Z}(\mathrm{add}))$, i.e., $\widehat{\mathbf{A}}(\mathrm{cat}) = \mathbf{Z}(\mathrm{cat})\mathbf{Z}(\mathrm{cat})^T$ and $\mathbf{A}(\mathrm{add}) = \mathbf{Z}(\mathrm{add})\mathbf{Z}(\mathrm{add})^T$ whose detailed calculations are shown in the bottom.

We can see that $\widehat{\mathbf{A}}(\mathrm{add}) = \widehat{\mathbf{A}}(\mathrm{cat}) + \mathbf{B}$, and $\mathbf{B}$ is a cross-multiplying term matrix. Since $\mathbf{Z}_p$ is fixed, $(a_3, a_4, b_3, b_4, c_3$, and c_4 is fixed), the loss function will force Z_f to constantly

adjust so that $\widehat{\mathbf{A}}$ is close to $\mathbf{A}$. Because $\widehat{\mathbf{A}}(\text{add})$ has more cross-multiplying terms, $\widehat{\mathbf{A}}(\text{add})$ may exert greater pressure $\mathbf{Z}_f(a_1, a_2, b_1, b_2, c_1, \text{and } c_2)$. Based on the above analysis, we chose these two modes to get $\mathbf{Z}$:

$$\widehat{\mathbf{A}}(\text{cat}) = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \end{bmatrix} \times \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{4} a_i^2 & \sum_{i=1}^{4} a_i b_i & \sum_{i=1}^{4} a_i c_i \\ \sum_{i=1}^{4} b_i a_i & \sum_{i=1}^{4} b_i^2 & \sum_{i=1}^{4} b_i c_i \\ \sum_{i=1}^{4} c_i a_i & \sum_{i=1}^{4} c_i b_i & \sum_{i=1}^{4} c_i^2 \end{bmatrix},$$

$$\begin{aligned} \widehat{\mathbf{A}}(\text{add}) = \begin{bmatrix} a_1 + a_3 & a_2 + a_4 \\ b_1 + b_3 & b_2 + b_4 \\ c_1 + c_3 & c_2 + c_4 \end{bmatrix} \times \begin{bmatrix} a_1 + a_3 & b_1 + b_3 & c_1 + c_3 \\ a_2 + a_4 & b_2 + b_4 & c_2 + c_4 \end{bmatrix} &= \begin{bmatrix} \sum_{i=1}^{4} a_i^2 & \sum_{i=1}^{4} a_i b_i & \sum_{i=1}^{4} a_i c_i \\ \sum_{i=1}^{4} b_i a_i & \sum_{i=1}^{4} b_i^2 & \sum_{i=1}^{4} b_i c_i \\ \sum_{i=1}^{4} c_i a_i & \sum_{i=1}^{4} c_i b_i & \sum_{i=1}^{4} c_i^2 \end{bmatrix} \\ &+ \begin{bmatrix} 2a_1a_3 + 2a_2a_4 & a_1b_3 + a_3b_1 + a_2b_4 + a_4b_2 & a_1c_3 + a_3c_1 + a_2c_4 + a_4c_2 \\ b_1a_3 + b_3a_1 + b_2a_4 + b_4a_2 & 2b_1b_3 + 2b_2b_4 & b_1c_3 + b_3c_1 + b_2c_4 + b_4c_2 \\ c_1a_3 + c_3a_1 + c_2a_4 + c_4a_2 & c_1b_3 + c_3b_1 + c_2b_4 + c_4b_2 & 2c_1c_3 + 2c_2c_4 \end{bmatrix}. \end{aligned} \tag{A.3}$$

Data Availability

Cora [35] is a citation network composed of 7 categories of machine learning papers. Cora includes 2708 papers and 5278 citation relationships between papers. 1433 unique words appear in all papers as the attributes. Yale is a social network including 8578 people and 188 attributes. The class year attribute divides the nodes into 7 categories. Part of links and labels of the datasets are used as training sets. K. Li, "The data about the facebook friendships of yale university." [Online]. Available: https://github.com/KaiyangLi1992/Privacy-Preserving-Social-Network-Embedding.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2019JBZ001, in part by the Beijing Natural Science Foundation under Grant 4202054, and in part by the National Natural Science Foundation of China under Grant 61871023 and Grant 61931001.

References

[1] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science & Engineering*, vol. 7, no. 2, pp. 766–775, 2018.

[2] H. Yang, Y. Liang, J. Yuan, Q. Yao, and J. Zhang, "Distributed blockchain-based trusted multi-domain collaboration for mobile edge computing in 5g and beyond," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, 2020.

[3] Z. Cai and Z. He, "Trading private range counting over big iot data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, 2019.

[4] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial iots," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[5] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social

networks," *IEEE Transactions on Dependable & Secure Computing*, vol. 15, no. 4, pp. 577–590, 2016.

[6] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *in Acm Sigmod International Conference on Management of Data*, Vancouver, Canada, 2008.

[7] F. O. Rousseau, J. Casas Roma, and M. Vazirgiannis, "Community-preserving anonymization of graphs," *Knowledge & Information Systems*, vol. 54, no. 2, pp. 315–343, 2018.

[8] A. Milani Fard and K. Wang, "Neighborhood randomization for link privacy in social network analysis," *World Wide Web internet & Web Information Systems*, vol. 18, no. 1, pp. 9–32, 2015.

[9] P. Mittal, C. Papamanthou, and D. Song, "Preserving link privacy in social network based systems," *Computer Science*, 2012, https://arxiv.org/abs/1208.6189.

[10] K. Zhou, T. P. Michalak, T. Rahwan, M. Waniek, and Y. Vorobeychik, "Attacking similarity-based link prediction in social networks," 2018, https://arxiv.org/abs/1809.08368.

[11] J. Chen, Z. Shi, Y. Wu, X. Xu, and H. Zheng, "Link prediction adversarial attack," 2018, https://arxiv.org/abs/1810.01110.

[12] S. Yu, M. Zhao, C. Fu et al., "Target defense against link-prediction-based attacks via evolutionary perturbations," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 2, pp. 1–767, 2019.

[13] M. Waniek, K. Zhou, Y. Vorobeychik, E. Moro, and T. Rahwan, "How to hide one's relationships from link prediction algorithms," *Scientific Reports*, vol. 9, no. 1, p. 12208, 2019.

[14] K. Li, G. Luo, Y. Ye, W. Li, S. Ji, and Z. Cai, "Adversarial privacy preserving graph embedding against inference attack," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6904–6915, 2021.

[15] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[16] X. Ying and X. Wu, "Randomizing social networks: a spectrum preserving approach," in *in Proceedings of the SIAM International Conference on Data Mining, SDM 2008*, Atlanta, Georgia, USA, 2008.

[17] Y. Li, H. Shen, C. Lang, and H. Dong, "Practical anonymity models on protecting private weighted graphs," *Neurocomputing*, vol. 218, pp. 359–370, 2016.

[18] M. Yuan, C. Lei, P. S. Yu, and T. Yu, "Protecting sensitive labels in social network data anonymization," *IEEE Transactions on Knowledge & Data Engineering*, vol. 25, no. 3, pp. 633–647, 2013.

[19] S. Chester and G. Srivastava, "Social network privacy for attribute disclosure attacks," in *International Conference on Advances in Social Networks Analysis & Mining*, Kaohsiung, Taiwan, 2011.

[20] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing Social Networks," in *34th International Conference on Very large Data Bases (VLDB)*, Auckland, New Zealand, 2008.

[21] L. Liu, J. Wang, J. Liu, and J. Zhang, "Privacy preservation in social networks with sensitive edge weights," in *Siam International Conference on Data Mining*, Sparks, Nevada, USA, 2009.

[22] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," *International Journal of Computer Trends & Technology*, vol. 17, no. 1, 2014.

[23] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems*, vol. 13, no. 4, pp. 18–28, 1998.

[24] H. L. Jensen, "Using neural networks for credit scoring," *Managerial Finance*, vol. 18, no. 6, pp. 15–26, 1992.

[25] S. Yang, H. Jian, Z. Ding, H. Zha, and C. L. Giles, "Iknn: informative k-nearest neighbor pattern classification," in *European Conference on Knowledge Discovery in Databases: Pkdd*, Berlin, Heidelberg, 2007.

[26] J. Gu, Z. Wang, J. Kuen, L. Ma, and G. Wang, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[27] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," 2017, https://arxiv.org/abs/1801.01078.

[28] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," 2013, https://arxiv.org/abs/1312.6203.

[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, https://arxiv.org/abs/1609.02907.

[30] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, https://arxiv.org/abs/1611.07308.

[31] D. J. Im, S. Ahn, R. Memisevic, and Y. Bengio, "Auto-encoding variational Bayes," https://arxiv.org/abs/1312.6114.

[32] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18*, 2018https://arxiv.org/abs/1802.04407.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., *Generative adversarial nets*, MIT Press, 2014.

[34] Z. Cai, Z. Xiong, H. Xu, P. Wang, and Y. Pan, "Generative adversarial networks," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.

[35] S. Prithviraj, N. Galileo, B. Mustafa, G. Lise, and G. Brian, *Collective Classification in Network Data*, Ai Magazine, 2008.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5230594, 11 pages
https://doi.org/10.1155/2022/5230594

Research Article

Energy-Efficient Computational Offloading for Secure NOMA-Enabled Mobile Edge Computing Networks

Haiping Wang

School of Mechatronics and Mould Engineering, Taizhou Vocational College of Science and Technology, Zhejiang 318020, China

Correspondence should be addressed to Haiping Wang; 39552529@qq.com

Received 23 February 2022; Revised 27 March 2022; Accepted 7 April 2022; Published 27 April 2022

Academic Editor: Yan Huo

Computational offloading and nonorthogonal multiple access (NOMA) are two promising technologies for alleviating the problems of limited battery capacity, insufficient computational capability, and massive deployment of terminal equipment in the Internet of Things (IoT) era. However, offloading data may be threatened by malicious eavesdroppers, which leads to more energy consumptions to avoid being eavesdropped. In this work, we study the energy-efficient way of computational offloading under the condition of certain security requirement in a secure NOMA-enabled mobile-edge computing (MEC) networks, where K end users are intended to offload their data to the N-antenna access point (AP) through the same resource block under the threat of an eavesdropper. We first address energy-efficient local resource allocation by minimizing sum-energy consumption of end users, subject to CPU frequencies, offloading bits, secrecy offloading rate, and transmit power. We then optimize the local resources to obtain the minimum computation latency of task for each end user, with the constraint of certain energy budget. The solutions to the above two optimization problems are given and demonstrated numerically by a 3-user scenario.

1. Introduction

Research shows that the contradiction between the limited resources of mobile terminal equipment and its explosive development of business and application requirements has become the most challenging tasks in the field of mobile Internet and Internet of Things (IoT) [1]. The resources of the terminal equipment are mainly reflected in the computational capability of the processor and the battery capacity. According to the theory of integrated circuits, the power consumption of the CPU is proportional to the cube of its frequency [2]. The stronger computing power, the shorter battery life.

The newly developed mobile-edge computing (MEC) [3] technology helps to resolve the above contradiction. It allows terminal devices to migrate data to the MEC server for computing-computational offloading [4]. Although MEC effectively solves the contradictions faced by the terminal and improves the service experience, there are still problems of security and high energy consumption [5].

In recent years, most of the research on MEC computational offloading focuses on the optimization of energy consumption or/and time delay without considering the security threat. Some [6–9] investigated how to save energy, some [10–14] studied from the perspective of improving real-time performance, and [15] took into account both energy consumption and computation latency. Although many achievements have been received in the optimization of computational offloading performance under the condition of no security threat, in practical applications, data offloading often needs to consider information security issues. Whether it is IoT data, Internet of Vehicles (IoV) data, video streaming analysis data, or other data with high computational intensity [16], data leakage needs to be avoided. After security processing, such as encrypting and decrypting data in the terminal and MEC server, the computational offloading process inevitably increases energy consumption and computation latency, resulting in distortion of the existing performance optimization research results. To this end, security should be considered when studying the performance of computational offloading in practical applications.

The research on the security of the computational offloading process can mainly be divided into two categories:

one is based on the traditional cryptography; the other is based on the physical layer security. [17, 18] studied the energy efficiency of computational offloading based on cryptographic encryption, taking into account the delay and energy consumption caused by encryption, making energy efficiency evaluation closer to practical applications. [17] compared the energy consumption results of computational offloading with encryption and without encryption. As expected the energy consumption with encryption is greater than that without encryption. In turn, the authors further considered compressing the data to reduce the total amount of data, shorten the transmission delay and energy consumption, and achieve desired results.

Compared with traditional cryptography, computational offloading based on physical layer security is more attractive to researchers. The physical layer security is keyless, which avoids the network vulnerability problem caused by key distribution and management [19]; on the other hand, the physical layer security has been proved to be reliable from the perspective of information theory [20]. In addition, the physical layer security and traditional cryptography technology belong to different layers in the network system, and there is no replacement relationship between them, and the existing traditional security system can still be retained. Presently, the research combining physical layer security with MEC computational offloading energy efficiency is still in its infancy, and there are not many related studies. Most of them focus on secrecy offloading rate improvement and secrecy offloading resource allocation.

The most effective way to improve the energy efficiency of the terminal side is to increase the secrecy offloading rate without increasing the transmit power, shorten the data offloading time, and then reduce the offloading energy consumption of the terminal. Therefore, the related secrecy capacity enhancement techniques in the physical layer security can be applied, such as improving the main channel through relays; or deteriorating the eavesdropper's channel through artificial noise; or enhancing the main capacity through nonorthogonal multiple access technology. [21] studies the use of relay to improve the offloading rate. Since the relay is untrustworthy, the MEC server needs to interfere with it accordingly. [22] uses full-duplex artificial noise to deteriorate the eavesdropper's channel, that is, the MEC server sends out interference signals, which does not interfere with the legitimate receiver, but can influence the eavesdropper. Cooperative jamming assisted scheme via cooperative NOMA transmission was studied in [23]. A novel jamming signal scheme was designed to multiuser multiserver MEC-enabled IoT in [15, 24] adopts nonorthogonal multiple access technology to improve the terminal's antieavesdropping capability. [25–27] takes unmanned aerial vehicle (UAV) as the target of MEC data offloading and improves the secrecy capacity of end users by adjusting the position and power of UAV.

As for secure offloading resource allocation, there are two sides: terminal side and MEC side. The terminalside resources include CPU frequency, transmit power, transmission rate, and for the data "partial offloading" mode, also include the data partition ratio. MEC side resources include server CPU frequency, channel resources, and the like. [28] optimizes energy consumption by taking the proportion of data offloading and transmit power as resource allocation objects, that is, minimizing the energy consumption of the terminal by reasonably adjusting the data ratio of local computing and offloading computing and offloading power. The scenario is extended from single user to multiuser [29, 30], and the weighted total power is minimized by rationally distributing the data offloading ratio and transmit power of each user. Since the state-of-the-art CPU architecture used by most terminal devices adopts dynamic frequency and voltage scaling (DVFS) technique, the energy consumption can be controlled by flexibly adjusting the frequency or voltage of the CPU, so the local CPU frequency can be further included in the resource optimization object. [31–33] include channel resources (such as time slots and frequencies) into optimization objects while considering local resources. In addition, [34, 35] also studied dividing the data into several parts and offloading them to different MEC servers for computation.

In this work, we further investigate terminal-side resource allocation for multiterminal energy efficiency of a secure MEC system enabled with NOMA, which is of high spectral efficiency and can accommodate massive devices. Zero-forcing (ZF) combined with successive interference cancelation (ZF-SIC) is adopted on the side of AP. Different from MMSE, ZF is less complex and more energy-efficient. We consider "partial offloading" mode and take more practical secrecy performance metric of secrecy outage probability (SOP) during offloading. Although [30] has studied energy-efficient resource allocation in secure NOMA-enabled mobile MEC networks, the authors only addressed the scenario of two user NOMA and one single antenna at the AP associated with the MEC server for simplicity. In this paper, we generalize the model of secure NOMA-enabled MEC with multiple users and multiple antennas at AP and tackle the efficient problem of energy and latency via local resource allocation. The major challenges of this work are (1) modeling wiretap channel and partial offloading mode and giving the closed-form expressions; (2) formulating sum-energy consumption minimization problem and computation latency minimization problem; (3) transforming minimization problems with multiple variables into a single variable one and giving the optimal solutions.

We adopt SOP as the secure QoS metric rather than secrecy capacity or ergodic secrecy capacity based on the following considerations: (1) more favorable main channel than eavesdropper channel is not always available; (2) the eavesdropper is always passive, which implies the CSI of eavesdropper channel is not available; (3) ergodic secrecy capacity needs to encode over a long period of time over all channel realizations, which incurs long delay and is suitable only for delay-tolerant applications; (4) secrecy outage is suitable for encoding confidential message over a single coherence interval or channel block.

The main contributions of the paper are summarized as follows:

(1) We generalize the model of NOMA-enabled MEC networks against an external eavesdropper by K

users and N antennas at AP associated with MEC server. A novel design framework is introduced by adopting zero forcing plus successive interference cancelation at the AP, jointly optimizing the number of offloading bits, local CPU frequency, and transmit power, targeting at end users' energy and latency

(2) Aiming at energy-efficient design, we characterize the optimization problem of sum energy of end users under certain SOP and latency requirement in the secure NOMA-enabled MEC networks, subject to CPU frequency, offloading bits, secrecy offloading rate, and transmit power of each user. We transform the complex and nonconvex problem into a single-variable one and give solution by numerical-finding

(3) Aiming at latency-efficient design, we investigate the problem of minimizing the computation latency of each user's task under certain SOP and energy budget requirement, with the constraint of CPU frequency, offloading bits, secrecy offloading rate, and transmit power. Similarly, the algorithm of the solution is given by transforming the original complex problem into a single-variable one

The rest of the paper is organized as follows. In the following section, we model the multiuser secure NOMA-enabled mobile MEC system and derive the closed-form expression of the secrecy outage probability of an individual user. In the third section, the optimization problem of sum-energy consumption of end users under certain SOP and latency requirement is characterized, and the algorithm of the solution is given. The problem of minimizing the computation latency of each user's task under certain SOP and energy budget requirement is addressed and solved in the fourth section. Numerical results are presented in the section of "Numerical Results" to confirm the efficient design done in the previous sections. Finally, a conclusion is drawn in the last section.

2. System Model

As depicted in Figure 1, we consider a secure uplink NOMA-enabled MEC networks, where K single-antenna users are intended to offload their computation-intensive tasks to the N-antenna access point (AP) (with an MEC server integrated) by sharing the same radio resource (such as frequency and/or time) at the presence of an eavesdropper (Eve) which is equipped with a single antenna. Here, $N \geq$ K is assumed.

All the channels are experiencing quasistatic Rayleigh fading. To user $k(k \in \mathscr{K} \triangleq \{1, \cdots, K\})$, the symbols used hereinafter are listed in Table 1.

Therefore, the instantaneous composite signals received at the AP and Eve are given by

$$y_a = \sum_{k=1}^{K} \sqrt{p_k} h_{a,k} s_k + w_a, \tag{1}$$

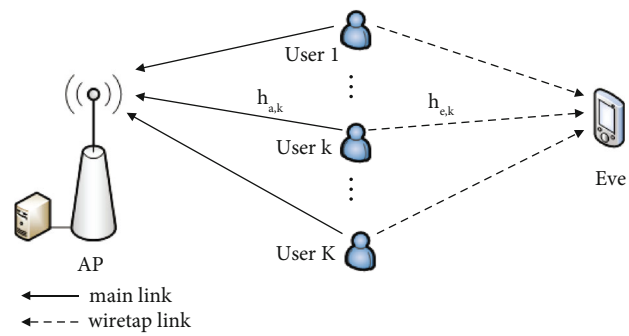


Figure 1: A model of secure uplink NOMA-enabled MEC networks with K users, one eavesdropper, and one N-antenna AP.

Table 1: Symbol description.

Symbol	Description
s_k	Complex information symbol from user k
p_k	Transmit power from user k
$h_{a,k}$	Channel gain vector from user k to AP
$h_{e,k}$	Channel gain from user k to Eve
w_a	Complex Gaussian noise vector at AP
w_e	Complex Gaussian noise at Eve
$d_{a,k}$	Distance from user k to AP
$d_{e,k}$	Distance from user k to Eve
α	Path-loss exponent
$g_{a,k}$	$g_{a,k} \sim \mathscr{CN}(0, I_N)$
$g_{e,k}$	$g_{e,k} \sim \mathscr{CN}(0, 1)$

$$y_e = \sum_{k=1}^{K} \sqrt{p_k} h_{e,k} s_k + w_e, \tag{2}$$

where $h_{a,k} = d_{a,k}^{-\alpha/2} g_{a,k}$ and $h_{e,k} = d_{e,k}^{-\alpha/2} g_{e,k}$; $\mathbb{E}[|s_k|^2] = 1$; $w_a \sim \mathscr{CN}(0, N_a I_N)$ and $w_e \sim \mathscr{CN}(0, N_e)$.

2.1. Wiretap Channel Model. Similar to [36], successive interference cancelation (SIC) combined with zero forcing (ZF) is employed at the AP, without loss of generality, to a specific user k, its aggregate signal left before decoding can be expressed as

$$y_{a,k}^{(\mathfrak{I})} = \sqrt{p_k} h_{a,k} s_k + \sum_{i \in \mathfrak{I}} \sqrt{p_i} h_{a,i} s_i + w_a, \tag{3}$$

where $\mathfrak{I} \subseteq \mathscr{K} \setminus \{k\}$, indicating U_k is decoded just before these users in set $\mathfrak{I}$ (index) during an SIC process.

After applying zero-forcing, the SINR of user k at the AP is given by

$$\gamma_k^{(\mathfrak{I})} = \frac{p_k \left\| Q_k^{(\mathfrak{I})} h_{a,k} \right\|^2}{N_a}, \tag{4}$$

where $Q_k^{(\mathfrak{I})}$ is a $N-|\mathfrak{I}|$ by N matrix whose rows are the orthonormal basis of the null space of the subspace spanned by the vector set $\{h_{a,i}|\forall i \in \mathfrak{I}\}$.

According to [37], $\gamma_k^{(\mathfrak{I})}$ obeys a chi-squared distribution with $2\times(N-|\mathfrak{I}|)$ degrees of freedom and has nothing to do with the specific users in set $\mathfrak{I}$ but its cardinality $|\mathfrak{I}|$. Specifically, $\lambda_k\gamma_k^{(\mathfrak{I})} \sim \chi^2_{2(N-|\mathfrak{I}|)}$, where $\lambda_k = N_a d_{a,k}^{\alpha}/p_k$. For the ease of presentation, we denote the cardinality of $\mathfrak{I}$ by n and rewrite $\gamma_k^{(\mathfrak{I})}$ by $\gamma_k^{(n)}$, whose probability density function (pdf) is given by

$$f_{\gamma_k^{(n)}}(\gamma_k) = \frac{\lambda_k^{N-n}}{\Gamma(N-n)}\gamma_k^{N-n-1}e^{-\lambda_k\gamma_k}, \gamma_k \geq 0. \tag{5}$$

As for the SINR of user k (denoted as ξ_k) at eve, we adopt the conservative assumption that all user interference can be canceled out by Eve. Therefore, $\xi_k = |h_{e,k}|^2 p_k/N_e$ obeys an exponential distribution with parameter $\mu_k = N_e d_{e,k}^{\alpha}/p_k$, i.e., $\xi_k \sim \mathbf{Exp}(\mu_k)$, whose cumulative distribution function (CDF) is given by

$$F_{\xi_k}(\xi_k) = 1 - e^{-\mu_k\xi_k}, \xi_k > 0. \tag{6}$$

2.2. Partial Offloading Model. We assume the task for each user can be partitioned into two parts with an arbitrary ratio. Let L_k and l_k be the total number of bits to be processed and the number of bits to be offloaded to the AP.

We consider a more practical scenario that the eavesdropper is passive and the AP only has the statistical characteristics of eavesdropper's channels. To this end, we adopt the secrecy outage probability (SOP) for quantifying the quality-of-service (QoS) of secure transmissions.

Suppose $R_{s,k}$ is the secrecy rate for transmit power p_k that satisfies the targeted secure QoS for user k. Therefore, the energy consumption of offloading can be formulated as

$$E_k^{\text{off}} = p_k t_k^{\text{off}} = p_k \frac{l_k}{BR_{s,k}}, \tag{7}$$

where t_k^{off} and B refer to offloading duration and the bandwidth of the channel, respectively.

We present the closed-form expression of SOP in form of theorem as follows.

Theorem 1. *The secrecy outage probability of user k for the required secrecy rate $R_{s,k}$ at the presence of n-user interference can be expressed as*

$$\begin{aligned} P_{so,k}(n, R_{s,k}) = &\frac{\boldsymbol{\gamma}\left(N-n, \lambda_k\left(2^{R_{s,k}}-1\right)\right)}{\boldsymbol{\Gamma}(N-n)} \\ &+ \left(\frac{\lambda_k}{\lambda_k + \mu_k 2^{-R_{s,k}}}\right)^{N-n} e^{\mu_k\left(1-2^{-R_{s,k}}\right)} \\ &\times \frac{\boldsymbol{\Gamma}\left(N-n, \left(\lambda_k + \mu_k 2^{-R_{s,k}}\right)\left(2^{R_{s,k}}-1\right)\right)}{\boldsymbol{\Gamma}(N-n)}, \end{aligned} \tag{8}$$

where

$$\boldsymbol{\gamma}(v, z) = \int_0^z u^{v-1}e^{-u}du, \tag{9}$$

and

$$\boldsymbol{\Gamma}(v, z) = \int_z^{\infty} u^{v-1}e^{-u}du = \boldsymbol{\Gamma}(v) - \boldsymbol{\gamma}(v, z), \tag{10}$$

are the incomplete Gamma function and its complement, respectively.

Proof. See Appendix. □

2.3. Local Computation Model. Let X_k be the computation intensity in CPU cycles per bit for user k. The total number of cycles for computing local part of task is $(L_k - l_k)X_k$. Each user adopts the advanced dynamic frequency and voltage scaling (DVFS) technique to control the energy consumption. According to [1], the energy consumption of a CPU cycle is given by $\kappa_k f_k^2$, where κ_k is a constant associated with the hardware architecture, and f_k is the CPU frequency for user k. For the local computation task of $(L_k - l_k)X_k$ cycles, the energy consumption can be derived:

$$E_k^{\text{loc}} = \kappa_k(L_k - l_k)X_k f_k^2. \tag{11}$$

The corresponding computation time for user k can be given by

$$t_k^{\text{loc}} = \frac{(L_k - l_k)X_k}{f_k}. \tag{12}$$

3. Sum-Energy Consumption Minimization

3.1. Problem Formulation. In this section, we focus on the problem of the sum-energy consumption minimization over transmit power p_k, offloading bits l_k, secrecy offloading rate $R_{s,k}$, and CPU frequency f_k, subject to the task $A_k(L_k, T, X_k)$ for each user.

By convention, we ignore the time of the data processing at the MEC as well as that of downlink transmission, due to the fact that the MEC processing speed is very fast and the processed result usually has fewer bits. Without loss of generality, the AP decodes the users' signals in the SIC order of user 1, user 2, ..., user K. As such, the number of user interferers for user k is $n = K - k$.

Mathematically, the minimization problem of sum-energy consumption can be formulated as

$$(P1):\ \min_{f,l,p,R_s} \sum_{k=1}^{K}\left(\kappa_k(L_k - l_k)X_k f_k^2 + p_k\frac{l_k}{BR_{s,k}}\right), \tag{13a}$$

$$\text{s.t.}\ \frac{(L_k - l_k)X_k}{f_k} \leq T, \forall k \in \mathscr{K}, \tag{13b}$$

$$\frac{l_k}{BR_{s,k}} \le T, \forall k \in \mathscr{K}, \tag{13c}$$

$$P_{so,k}(K-k, R_{s,k}) \le \epsilon, \forall k \in \mathscr{K}, \tag{13d}$$

$$0 \le l_k \le L_k, \forall k \in \mathscr{K}, \tag{13e}$$

$$f_k \le f_k^{\max} \forall k \in \mathscr{K}, \tag{13f}$$

where $f = [f_1, f_2, \cdots, f_K]$, $l = [l_1, l_2, \cdots, l_K]$, $p = [p_1, p_2, \cdots, p_K]$, and $R_s = [R_{s,1}, R_{s,2}, \cdots, R_{s,K}]$ refer to the CPU frequency vector, the offloading bit vector, the transmit power vector, and secrecy offloading rate vector, respectively; T is the computation latency, and $f_k^{\max}$ denotes the upper bound of CPU frequency for user k.

3.2. Solution to Problem (P1). The problem $(P1)$ is complicated and nonconvex due to the nonconvex nature of constraints (13d). Although the closed-form expression of the optimal solution is not available, we can obtain a suboptimal solution numerically by (1) simplifying constraints, (2) relaxing and transforming the multivariable problem into a single variable problem, and (3) giving the solution numerically.

We first simplify the constraints by the following lemma.

Lemma 2. *The constraint (13b) and (13d) are strictly binding for the optimal solution of problem (P1), i.e.,*

$$f_k = \frac{(L_k - l_k)X_k}{T}, \forall k \in \mathscr{K}, \tag{14}$$

$$P_{so,k}(K-k, R_{s,k}) = \epsilon, \forall k \in \mathscr{K}. \tag{15}$$

Proof. Observing the first term in (13a), the local computation energy consumption is an increasing function of f_k. Obviously, the lowest CPU frequency that satisfies the condition achieves the minimum energy consumption.

According to the property of the SOP, $P_{so,k}(n, R_{s,k})$ is an increasing function of $R_{s,k}$. The maximum $R_{s,k}$ that satisfies (13d) is the ϵ-outage secrecy capacity $C_{s,k}^{(n)}(\epsilon)$ which makes $P_{so,k}(n, C_{s,k}^{(n)}(\epsilon)) = \epsilon$ hold. This completes the proof. □

The problem $(P1)$ in (13a) is still complex and nonconvex. We continue to transform the problem by relaxing offloading duration with T, i.e.,

$$(P1.1):\ \min_{l,p,R_s} \sum_{k=1}^{K} \left(\frac{\kappa_k X_k^3 (L_k - l_k)^3}{T^2} + p_k T \right), \tag{16a}$$

$$\text{s.t.} R_{s,k} \ge \frac{l_k}{BT}, \forall k \in \mathscr{K}, \tag{16b}$$

$$P_{so,k}(K-k, R_{s,k}) = \epsilon, \forall k \in \mathscr{K}, \tag{16c}$$

$$\left[L_k - \frac{T f_k^{\max}}{X_k} \right]^+ \le l_k \le L_k, \forall k \in \mathscr{K}, \tag{16d}$$

where $[\cdot]^+ = \max(\cdot, 0)$.

Since the transmit power p_k increases with $R_{s,k}$, $R_{s,k} = l_k/BT$ achieves the minimum energy consumption for fixing other parameters. Moreover, p_k can be expressed as a function of $R_{s,k}$ from (16c). As such, we can further transform the problem into the one only having a single variable.

$$(P1.2):\ \min_l \sum_{k=1}^{K} \left(\frac{\kappa_k X_k^3 (L_k - l_k)^3}{T^2} + P_k\left(K-k, \frac{l_k}{BT}\right) T \right), \tag{17a}$$

$$\text{s.t.} \left[L_k - \frac{T f_k^{\max}}{X_k} \right]^+ \le l_k \le L_k, \forall k \in \mathscr{K}, \tag{17b}$$

where $P_k(n, R_{s,k})$ is the expression of the function for p_k derived from $P_{so,k}(n, R_{s,k}) = \epsilon$.

Although we cannot get a closed-form solution to this problem, the optimal solution can be found numerically. The numerical solution to problem $(P1.2)$ is shown in Algorithm 1. Obviously, the solution to problem $(P1.2)$ is equivalent to minimize the energy consumption of each user over offloading bit l_k independently. We will demonstrate it numerically in detail in Section.

3.3. Discussion

3.3.1. Local Computation Only Mode. The energy consumption of local computation only mode for user k is $\kappa_k X_k^3 L_k^3 / T^2$ by setting $l_k = 0$. However, the local computation only mode is not always available unless $f_k^{\max} \ge L_k X_k / T$ is satisfied.

3.3.2. Full Offloading Mode. Similarly, the full offloading mode is not always achievable as given the SOP requirement each user k has its asymptotic secrecy rate, which is explained by the following theorem.

Theorem 3. *Given the SOP of ϵ, the limited value of secrecy rate $R_{s,k}$ for user k at the presence of n-user interference is* $\log \mu_k/\lambda_k + 1/N - n \log \epsilon - \log(1 - \sqrt[N-n]{\epsilon})$.

Proof. $p_x \longrightarrow \infty$ makes μ_k and λ approach to 0 while their ratio keep as a limited value. Then, the asymptotic secrecy outage probability for (8) can be formulated as

$$P_{so,k}(n, R_{s,k}) \xrightarrow[p_x]{a.s.} \left(\frac{1}{1 + (\mu_k/\lambda_k) 2^{-R_{s,k}}} \right)^{N-n}. \tag{18}$$

In turn, the asymptotic -outage secrecy capacity is

$$C_{s,k}^{(n)}(\epsilon) \xrightarrow[p_x]{a.s.} \log \frac{\mu_k}{\lambda_k} + \frac{1}{N-n} \log \epsilon - \log\left(1 - \sqrt[N-n]{\epsilon}\right). \tag{19}$$

This completes the proof. □

```
Setting: T, B, ϵ, N, K, α, N_a, N_e;
Repeat
  Setting: f_k^max, κ_k, X_k, L_k, d_{a,k}, d_{e,k};
  Initialization: l_k;
  Repeat
    Repeat
      Search p_k with certain l_k via constraint of (13d);
    Until SOP converges to ϵ within a prescribed accuracy;
    Calculate E_k^loc + E_k^off;
  Until l_k = L_k
  Search l_k^* to make E_k^loc + E_k^off the least;
  Calculate f_k^*, R_{s,k}^*, p_k^*;
  Output: l_k^*, f_k^*, R_{s,k}^*, p_k^*;
Until k = K
```

Algorithm 1: Optimal solution to problem ($P1$).

If $N_a = N_e$, the asymptotic-outage secrecy capacity can be further expressed as

$$C_{s,k}^{(n)}(\epsilon) \xrightarrow[p_x]{a.s.} \alpha \log \frac{d_{e,k}}{d_{a,k}} + \frac{1}{N-n} \log \epsilon - \log\left(1 - \sqrt[N-n]{\epsilon}\right). \tag{20}$$

To achieve full offloading mode, the following inequality must be satisfied.

$$L_k < BT\left(\alpha \log \frac{d_{e,k}}{d_{a,k}} + \frac{1}{N-K+k} \log \frac{\epsilon}{1 - \sqrt[N-K+k]{\epsilon}}\right). \tag{21}$$

3.3.3. Computational Complexity. Observing Algorithm 1, there roughly exist five nesting steps in the process of numerical-finding: (1) calculate SOP with variable $R_{s,k}$ and find the ϵ-outage secrecy capacity; (2) calculate ϵ-outage secrecy capacity with variable transmit power and search optimal transmit power with certain l_k; (3) calculate total energy consumption and search for the optimal l_k to achieve minimum total energy consumption; (4) loop from user 1 to user K. Therefore, the computational complexity is $O(Km^3)$ (here, m refers to the computational complexity of SOP.).

4. Computation Latency Minimization

4.1. Problem Formulation. We continue to study the optimization problem of task computation latency over transmit power p_k, offloading bits l_k, secrecy offloading rate $R_{s,k}$, and CPU frequency f_k for user k with the constraint of certain energy budget in this section.

All the assumptions and notations are the same as in section. We form the problem as follows

$$(P2):\ \min_{f_k, l_k, R_{s,k}, p_k} \max\left(t_k^{\text{loc}}, t_k^{\text{off}}\right). \tag{22a}$$

$$\text{s.t.}\, \kappa_k (L_k - l_k) X_k f_k^2 + p_k \frac{l_k}{BR_{s,k}} \le E_k^{bu}, \tag{22b}$$

$$P_{so,k}(K-k, R_{s,k}) \le \epsilon, \tag{22c}$$

$$0 \le l_k \le L_k, \tag{22d}$$

$$f_k \le f_k^{\max}, \tag{22e}$$

where $t_k^{\text{loc}} = (L_k - l_k)X_k/f_k$ and $t_k^{\text{off}} = l_k/(BR_{s,k})$ according to the definitions in section, and E_k^{bu} refers to the energy budget of user k for its task.

4.2. Solution to Problem (P2). Before solving problem ($P2$) numerically, we need to simplify the problem via the following lemma.

Lemma 4. *The optimal solution of the variables f_k, l_k, $R_{s,k}$, and p_k should make $t_k^{loc} = t_k^{off}$.*

Proof. We prove this lemma via contradiction. Let f_k^*, l_k^*, $R_{s,k}^*$, and p_k^* be the jointly optimal values of f_k, l_k, $R_{s,k}$, and p_k. We assume $(L_k - l_k^*)X_k/f_k^* < l_k^*/(BR_{s,k}^*)$. One can find by keeping p_k and $R_{s,k}$ fixed, reducing l_k makes t_k^{off} and E_k^{off} decline. Although E_k^{loc} increases with l_k decreasing (f_k fixed), the increase in E_k^{loc} can be compensated by the decrease in E_k^{off}. Therefore, there exists another $\{f_k^*, l_k', R_{s,k}^*, p_k^*\}$, where $l_k' = l_k^* - \tau_k$ and τ_k is a small positive value, making $(L_k - l_k')X_k/f_k^* \le l_k'/(BR_{s,k}^*)$. Similarly, if $(L_k - l_k^*)X_k/f_k^* > l_k^*/(BR_{s,k}^*)$ is supposed, there exists another $\{f_k^*, l_k'', R_{s,k}^*, p_k^*\}$ that makes $(L_k - l_k'')X_k/f_k^* \ge l_k''/(BR_{s,k}^*)$ hold, where $l_k'' = l_k^* + \tau_k$.

We complete the proof. □

Combined with Lemma 2 and Lemma 4, we transform the problem ($P2$) into the following form.

$$(\text{P2.1}):\ \min_{l_k} T_k, \tag{23a}$$

$$\text{s.t.}\, \frac{\kappa_k X_k^3 (L_k - l_k)^3}{T_k^2} + P_k\left(K-k, \frac{l_k}{BT_k}\right) T_k \le E_k^{bu}, \tag{23b}$$

Setting: B, ϵ, N, K, α, N_a, N_e;
Repeat
  Setting: $f_k^{\max}$, κ_k, X_k, L_k, $d_{a,k}$, $d_{e,k}$, E_k^{bu};
  Initialization: l_k, T_k;
  Repeat
    Repeat
      Repeat
        Search p_k with certain l_k and T_k via constraint of (22c);
      Until SOP converges to ϵ within a prescribed accuracy;
      Calculate $E_k^{loc} + E_k^{off}$;
    Until T_k satisfies the condition in (23b)
  Until $l_k = L_k$
  Search l_k^* to make T_k the least;
  Calculate f_k^*, $R_{s,k}^*$, p_k^*;
  Output: l_k^*, f_k^*, $R_{s,k}^*$, p_k^*;
Until $k = K$

Algorithm 2: Optimal solution to problem ($P2$).

$$\left[L_k - \frac{T_k f_k^{\max}}{X_k}\right]^+ \le l_k \le L_k, \tag{23c}$$

The problem is changed into the form with a single variable, and we can find the optimal solution numerically, the algorithm of which is shown in Algorithm 2.

Different from Algorithm 1, Algorithm 2 has one more nesting step in the process of numerical-finding: Search for the minimum T_k for the fixed l_k, which makes its computational complexity be $O(Km^4)$.

5. Numerical Results

In this section, numerical results are presented to further validate the previous research results of the energy-efficient computational offloading design in this secure uplink NOMA-enabled MEC networks.

To demonstrate the numerical results more clearly, we take a 3-user scenario for instance, the layout of which is shown in Figure 2. The distances (grid) between each user and the AP as well as eve are listed in Table 2. Each grid is supposed to be 20 meters, thus, the distance from user 1 to the AP is 80 meters and about 152.3 meters to eve.

We assume the number of antennas equipped at the AP is 10, i.e., $N = 10$, the path loss in dB is expressed as $PL = 10\alpha \lg(d) + 43.5$, where the path loss exponent α is set to 3.76 according to 3GPP urban path loss model. The bandwidth $B = 10$ MHz and the equal AWGN power of the AP and eve are supposed, i.e., $N_a = N_e = N_0 B$, where $N_0 = -160$ dBm/Hz is the AWGN power spectral density.

For simplicity, we assume each end user has the same performance and tasks to process. The total bits of task for each user is $L_1 = L_2 = L_3 = 3$ Mbits. The computation intensity is $X_1 = X_2 = X_3 = 50$ cycles/bit. The maximum CPU frequency is 2 GHz. The effective capacitance coefficient is set as $\kappa_1 = \kappa_2 = \kappa_3 = 10^{-28}$. Without loss of generality, the SIC order is from user 1 to user 2 to user 3.

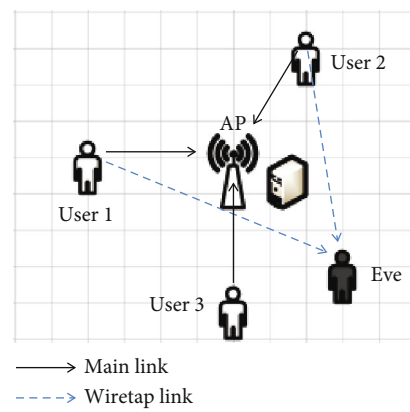


Figure 2: A 3-user scenario of secure uplink NOMA MEC system.

Table 2: Distances between each user and AP as well as eve.

Distance (grid)	User 1	User 2	User 3
AP	4	$\sqrt{13}$	4
Eve	$\sqrt{58}$	$\sqrt{37}$	$\sqrt{10}$.

5.1. Energy Consumption Minimization. Based on the above assumptions, the curves of the energy consumption versus offloading bits (Mbits) for each end user are shown in Figure 3, where the computation latency $T = 0.15$ s and the secrecy outage probability $\epsilon = 0.2$.

We note each curve starts from the same value (0.015 Joule) due to the similar parameter assumption for each user, goes down until the minimum energy consumption, then rises up, with the increase of offloading bits. $l_k = 0$ means local computation only mode, and $l_k = 3$ Mbits specifies full offloading mode. In this case, both modes cannot obtain the minimum energy consumption. The minimum energy consumptions for user 1, user 2, and user 3 are 0.0096 Joule, 0.0071 Joule, and 0.012 Joule, respectively. The corresponding offloading bits are $l_1 = 1$ Mbits, $l_2 = 1.3$

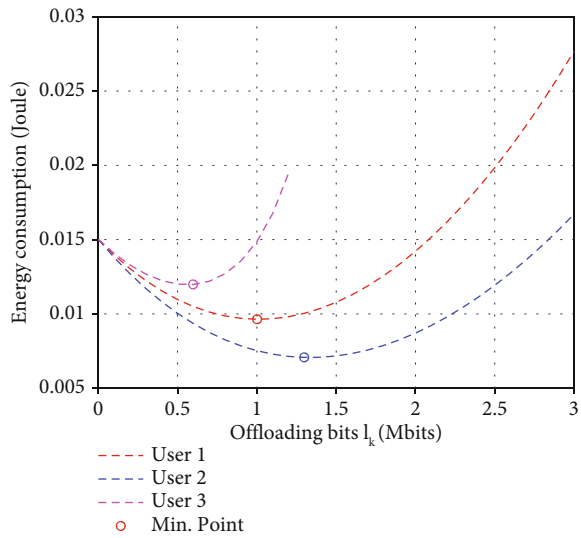


Figure 3: Energy consumption versus offloading bits for each user at the assumption of $T = 0.15$ s and $\epsilon = 0.2$

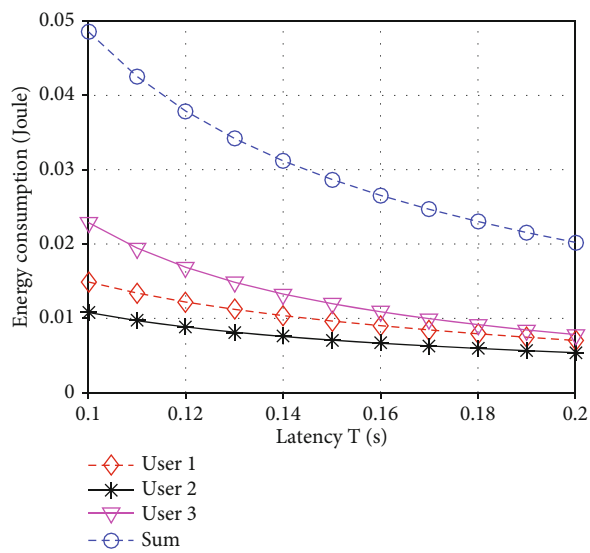


Figure 4: Minimum energy consumption versus latency for each user and their sum from 0.1 s to 0.2 s.

Mbits, and $l_3 = 0.6$ Mbits. Here, the resolution of offloading bits is 0.1 Mbits.

Although user 1 and user 3 have the same distances to the AP, due to the different distances to eve and the different SIC order, which result in the different secrecy outage capacities, user 1 has less energy consumption than user 3. Specifically, user 3 needs 0.0024 Joule more energy to overcome the distance difference of eve. Yet, user 2 has the least energy consumption even if it is not SIC decoded at last.

By the way, each user has an asymptotic ϵ-secrecy outage capacity. They are $C_{s,1}^{(2)}(\epsilon = 0.2) = 5.6589$ bps/Hz, $C_{s,2}^{(1)}(\epsilon = 0.2) = 5.1894$ bps/Hz, and $C_{s,3}^{(0)}(\epsilon = 0.2) = 1.2429$ bps/Hz. The asymptotic offloading bits for user 3 is about 1.86 Mbits, which indicates user 3 cannot conduct full offloading mode as the total bits of the task are 3 Mbits.

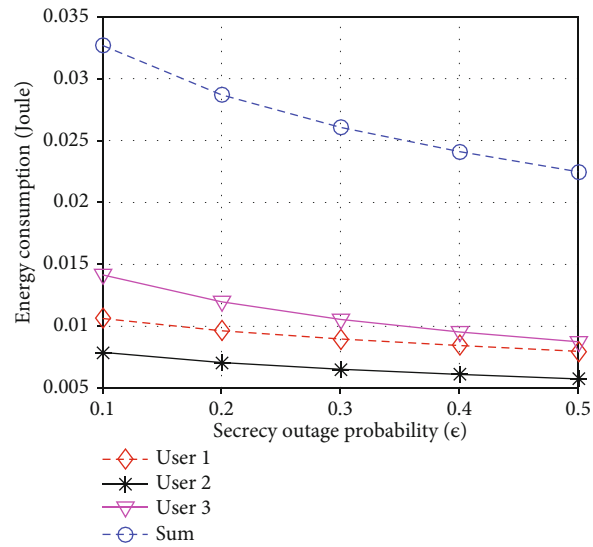


Figure 5: Minimum energy consumption versus secrecy outage probability for each user and their sum from $\epsilon = 0.1$ to $\epsilon = 0.5$.

We continue to study the impacts of the latency T and the secrecy outage probability ϵ on the minimum energy consumptions for each user and their sum.

We first illustrate the curves of the minimum energy consumption versus latency from $T = 0.1$ s to $T = 0.2$ s in Figure 4. For each user, the minimum energy consumption decreases with the increase of T. When $T = 0.1$ s, the minimum energy consumptions for user 1, user 2, user 3, and their sum are 0.015 Joule, 0.011 Joule, 0.023 Joule, and 0.049 Joule, respectively, while they are reduced to 0.007 Joule, 0.0054 Joule, 0.0078 Joule, and 0.0202 Joule when $T = 0.2$ s. The values for $T = 0.15$ s are the same as those achieved in Figure 3, which also confirms the reliability of the numerical results in Figure 4. As expected, user 2 has the least energy consumptions for the same T, while user 3 needs more energy consumption than user 1, just as in Figure 3.

Figure 5 depicts the curves of the minimum energy consumptions versus secrecy outage probability for each user and their sum from $\epsilon = 0.1$ to $\epsilon = 0.5$. It is easy to confirm the reliability of the numerical results in Figure 5 by observing the values for $\epsilon = 0.2$. Similar to the impact of latency T, the minimum energy consumptions decline with the secrecy outage probability increasing from 0.0106 Joule, 0.0079 Joule, 0.0142 Joule, 0.0327 Joule for $\epsilon = 0.1$ to 0.008 Joule, 0.0058 Joule, 0.0087 Joule, and 0.0225 Joule for $\epsilon = 0.5$. We note the minimum energy consumption for user 3 at $\epsilon = 0.1$ (0.0142 Joule) is very close to that of local computation only mode (0.015 Joule), which points out offloading is not always necessary especially when the eavesdropper's channel is much better than the legal channel.

5.2. Latency Minimization. Figure 6 shows the latency of the three end users versus offloading bits. The curves are generated on the assumption of the secrecy outage probability $\epsilon = 0.2$ and the energy budget $E^{bu} = 0.03$ Joule. The resolutions of the latency and offloading bits are 0.005 s and 0.05 Mbits, respectively. One can note each curve

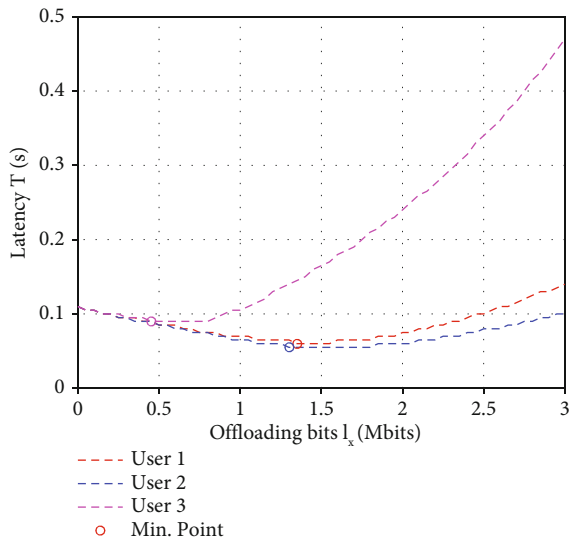


Figure 6: Latency versus offloading bits for each user at the assumption of $E^{bu} = 0.03$ Joule and $\epsilon = 0.2$.

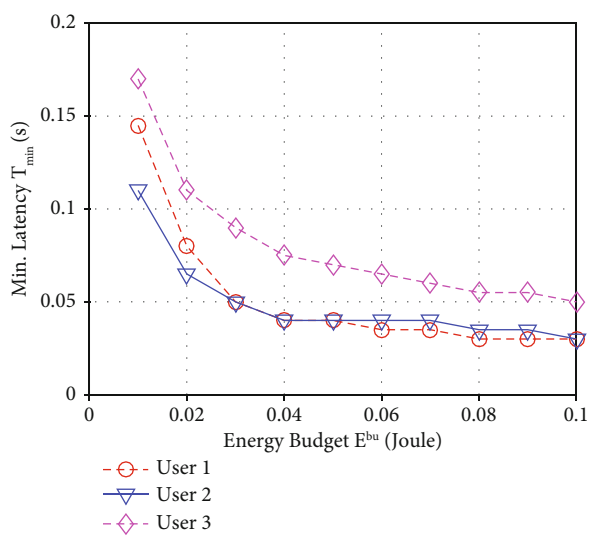


Figure 7: Minimum latency versus energy budget for each user from $E^{bu} = 0.01$ Joule to $E^{bu} = 0.1$ Joule.

declines first and then climbs up, which indicates there exists an optimal value. In this case, the minimum values of the latency for user 1, user 2, and user 3 are 0.06 s, 0.055 s, and 0.09 s, respectively, which are achieved by setting offloading bit $l_1^* = 1.35$ Mbits, $l_2^* = 1.30$ Mbits, and $l_3^* = 0.45$ Mbits, respectively. As such, $f_1^* = 1375$ MHz, $f_2^* = 1545.45$ MHz, $f_3^* = 1416.67$ MHz; $R_{s,1}^* = 2.25$ bps/Hz, $R_{s,2}^* = 2.36$ bps/Hz, $R_{s,3}^* = 0.5$ bps/Hz; and $p_1^* = 234.4$ mW, $p_2^* = 158.8$ mW, $p_3^* = 40.6$ mW.

Figure 7 shows the minimum latency versus the energy budget from $E^{bu} = 0.01$ Joule to $E^{bu} = 0.1$ Joule. As expected, the minimum latency decreases with the energy budget increasing. However, the decrease is significant from $E^{bu} = 0.01$ Joule to $E^{bu} = 0.04$ Joule, where the minimum latency for user 1 is reduced from 0.145 s to 0.05 s. For user 3 whose channel condition is not good enough, the reduction is still effective when E^{bu} is greater than 0.04 Joule. Meanwhile, the curves of user 1 and user 2 evert such that the minimum latency of user 2 is greater than that of user 1 from $E^{bu} = 0.05$.

Here so far, all these numeric results and observations in this section are consistent with the expectations.

6. Conclusion

In this work, we investigated a secure uplink NOMA-enabled MEC network under the threat of an eavesdropper, where K end users simultaneously offload their partial computation-intensive tasks to the MEC server in the same resource block and ZF-SIC is adopted at the multiantenna AP associated with MEC server. We first derived the closed-form expression of individual SOP for an arbitrary SIC order. We then characterize the optimization problem of sum-energy consumption subject to offloading bits, secrecy offloading rate, local CPU frequency, and transmit power and gave the solution. We further studied the problem of minimizing computation latency under condition of certain SOP requirement and energy budget through proving the equality of local computing time and offloading duration and transforming it into a single-variable problem. All the solutions are demonstrated and validated numerically by a 3-user case of a secure NOMA-enabled MEC network.

Appendix

Proof of Theorem 1

Given (5) and (6), jointly with the independence of $\gamma_k^{(n)}$ and ξ_k, the process of the derivation for the secrecy outage probability of user k is shown as follows,

$$\begin{aligned}
P_{so,k}(n, R_s) &= \Pr\left(\frac{1+\gamma_k^{(n)}}{1+\xi_k} < 2^{R_s}\right) \\
&= 1 - \int_{2^{R_s}-1}^{\infty} d\gamma_k \int_0^{2^{-R_s}\gamma_k + 2^{-R_s} - 1} f_{\gamma_k^n}(\gamma_k) f_{\xi_k}(\xi_k) d\xi_k \\
&= 1 - \int_{2^{R_s}-1}^{\infty} f_{\gamma_k^n}(\gamma_k) F_{\xi_k}\left(2^{-R_s}\gamma_k + 2^{-R_s} - 1\right) d\gamma_k \\
&= 1 - \int_{2^{R_s}-1}^{\infty} \frac{\lambda_k^{N-n}\gamma_k^{N-n-1}}{\Gamma(N-n)e^{\lambda_k\gamma_k}}\left(1 - e^{-\mu_k\left(2^{-R_s}\gamma_k + 2^{-R_s} - 1\right)}\right) \\
&\quad \cdot d\gamma_k = 1 - \frac{\lambda_k^{N-n}}{\Gamma(N-n)} \int_{2^{R_s}-1}^{\infty} \gamma_k^{N-n-1} e^{-\lambda_k\gamma_k} d\gamma_k \\
&\quad + \frac{\lambda_k^{N-n} e^{\mu_k\left(1-2^{-R_s}\right)}}{\Gamma(N-n)} \int_{2^{R_s}-1}^{\infty} \gamma_k^{N-n-1} e^{-\left(\lambda_k + \mu_k 2^{-R_s}\right)\gamma_k} d\gamma_k \\
&= 1 - \frac{\Gamma\left(N-n, \lambda_k\left(2^{R_s}-1\right)\right)}{\Gamma(N-n)} \\
&\quad + \frac{\lambda_k^{N-n} e^{\mu_k\left(1-2^{-R_s}\right)}}{\left(\lambda_k + \mu_k 2^{-R_s}\right)^{N-n}} \frac{\Gamma\left(N-n, \left(\lambda_k + \mu_k 2^{-R_s}\right)\left(2^{R_s}-1\right)\right)}{\Gamma(N-n)} \\
&= \frac{\gamma\left(N-n, \lambda_k\left(2^{R_s}-1\right)\right)}{\Gamma(N-n)} \\
&\quad + \frac{\lambda_k^{N-n} e^{\mu_k\left(1-2^{-R_s}\right)}}{\left(\lambda_k + \mu_k 2^{-R_s}\right)^{N-n}} \frac{\Gamma\left(N-n, \left(\lambda_k + \mu_k 2^{-R_s}\right)\left(2^{R_s}-1\right)\right)}{\Gamma(N-n)}.
\end{aligned} \tag{A.1}$$

The following integral result is applied [[38], Eq. (3.381.9)] during the above derivation,

$$\int_u^{\infty} x^m e^{-\beta x} dx = \frac{\Gamma(m+1, \beta u)}{\beta^{m+1}}. \tag{A.2}$$

The proof is completed.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the General Scientific Research Projects of Zhejiang Education Department (Grant no. Y202147949).

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE communications surveys & tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[2] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 13, no. 2-3, pp. 203–221, 1996.

[3] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[4] K. Guo, R. Gao, W. Xia, and T. Q. S. Quek, "Online learning based computation offloading in mec systems with communication and computation dynamics," *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 1147–1162, 2021.

[5] S. N. Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison, "The extended cloud: review and analysis of mobile edge computing and fog from a security and resilience perspective," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2586–2595, 2017.

[6] Y. Zhang, J. He, and S. Guo, "Energy-efficient dynamic task offloading for energy harvesting mobile cloud computing," in *2018 IEEE Int. Conf. Netw., Architecture and Storage (NAS)*, pp. 1–4, Chongqing, China, 2018.

[7] S. Chouhan, "Energy optimal partial computation offloading framework for mobile devices in multi-access edge computing," in *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1–6, Split, Croatia, 2019.

[8] C. He, R. Wang, and Z. Tan, "Energy-aware collaborative computation offloading over mobile edge computation empowered fiber-wireless access networks," *IEEE Access*, vol. 8, pp. 24662–24674, 2020.

[9] X. Li, Y. Dang, M. Aazam, X. Peng, T. Chen, and C. Chen, "Energy-efficient computation offloading in vehicular edge cloud computing," *IEEE Access*, vol. 8, pp. 37632–37644, 2020.

[10] R. M. Shukla and A. Munir, "A computation offloading scheme leveraging parameter tuning for real-time iot devices," in *2016 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS)*, pp. 208-209, Gwalior, India, 2016.

[11] R. M. Shukla and A. Munir, "An efficient computation offloading architecture for the internet of things (iot) devices," in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 728–731, Las Vegas, NV, USA, 2017.

[12] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in uav-enabled wireless-powered mobile-edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.

[13] Y. Shuai and R. Langar, "Collaborative computation offloading for multi-access edge computing," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 689–694, Arlington, VA, USA, 2019.

[14] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019.

[15] J. Xu, P. Zhu, J. Li, and X. You, "Secure computation offloading for multi-user multi-server mec-enabled iot," in *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6, Montreal, QC, Canada, 2021.

[16] L. Tang and Q. Li, "Energy and time optimization for wireless computation offloading," in *2015 International Conference on Wireless Communications & Signal Processing (WCSP)*, pp. 1–5, Nanjing, China, 2015.

[17] I. A. Elgendy, W.-Z. Zhang, Y. Zeng, H. He, Y.-C. Tian, and Y. Yang, "Efficient and secure multi-user multi-task computation offloading for mobile-edge computing in mobile iot networks," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2410–2422, 2020.

[18] U. A. Khan, W. Khalid, and S. Saifullah, "Energy efficient resource allocation and computation offloading strategy in a uav-enabled secure edge-cloud computing system," in *2020 IEEE Int. Conf. Smart Internet of Things (SmartIoT)*, pp. 58–63, Beijing, China, 2020.

[19] B. Schneier, "Cryptographic design vulnerabilities," *Computer*, vol. 31, no. 9, pp. 29–33, 1998.

[20] A. D. Wyner, "The wire-tap channel," *Bell Labs Technical Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.

[21] P. Zhao, W. Zhao, H. Bao, and B. Li, "Security energy efficiency maximization for untrusted relay assisted Noma-mec network with wpt," *IEEE Access*, vol. 8, pp. 147387–147398, 2020.

[22] X. He, R. Jin, and H. Dai, "Physical-layer assisted secure offloading in mobile-edge computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4054–4066, 2020.

[23] L. Qian, W. Wu, W. Lu, Y. Wu, B. Lin, and T. Q. Quek, "Secrecy-based energy-efficient mobile edge computing via cooperative non-orthogonal multiple access transmission," *IEEE Transactions on Communications*, vol. 69, no. 7, pp. 4659–4677, 2021.

[24] W. Wu, X. Wang, F. Zhou, K. K. Wong, C. Li, and B. Wang, "Resource allocation for enhancing offloading security in Noma-enabled mec networks," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3789–3792, 2021.

[25] D. Han and T. Shi, "Secrecy capacity maximization for a uav-assisted mec system," *China Communications*, vol. 17, no. 10, pp. 64–81, 2020.

[26] Y. Li, Y. Fang, and L. Qiu, "Joint computation offloading and communication design for secure uav-enabled mec systems," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Nanjing, China, 2021.

[27] X. Gu, G. Zhang, M. Wang, W. Duan, M. Wen, and P. H. Ho, "UAV-aided energy-efficient edge computing networks: security offloading optimization," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4245–4258, 2022.

[28] T. Bai, J. Wang, Y. Ren, and L. Hanzo, "Energy-efficient computation offloading for secure uav-edge-computing systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6074–6087, 2019.

[29] W. Zhao, B. Wang, H. Bao, and B. Li, "Secure energy-saving resource allocation on massive mimo-mec system," *IEEE Access*, vol. 8, pp. 137244–137253, 2020.

[30] W. Wei and F. Zhou, "Energy-efficient resource allocation for secure Noma-enabled mobile edge computing networks," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 493–505, 2020.

[31] H. Lin, Y. Cao, Y. Zhong, and P. Liu, "Secure computation efficiency maximization in Noma-enabled mobile edge computing networks," *IEEE Access*, vol. 7, pp. 87504–87512, 2019.

[32] J.-B. Wang, H. Yang, M. Cheng, J.-Y. Wang, M. Lin, and J. Wang, "Joint optimization of offloading and resources allocation in secure mobile edge computing systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8843–8854, 2020.

[33] S. Han, X. Xu, S. Fang et al., "Energy efficient secure computation offloading in Noma-based mmtc networks for iot," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5674–5690, 2019.

[34] F. Fang, K. Wang, and Z. Ding, "Optimal task assignment and power allocation for downlink Noma mec networks," in *2019 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, Waikoloa, HI, USA, 2019.

[35] M. Zhao, H. Bao, L. Yin, J. Yao, and T. Q. S. Quek, "Secrecy offloading rate maximization for multi-access mobile edge computing networks," *IEEE Communications Letters*, vol. 25, no. 12, pp. 3800–3804, 2021.

[36] K. Jiang, T. Jing, Z. Li, Y. Huo, and F. Zhang, "Analysis of secrecy performance in fading multiple access wiretap channel with sic receiver," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, 2017.

[37] K. Jiang, T. Jing, F. Zhang, Y. Huo, and Z. Li, "ZF-SIC based individual secrecy in simo multiple access wiretap channel," *IEEE Access*, vol. 5, pp. 7244–7253, 2017.

[38] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series and Products*, Academic Press, 7th edition, 2007.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6951849, 13 pages
https://doi.org/10.1155/2022/6951849

Research Article

Adaptive Differential Evolution Algorithm with Simulated Annealing for Security of IoT Ecosystems

Qianqian Liu,[1] **Xiaoyan Zhang**,[1] **Qiaozhi Hua**,[2] **Zheng Wen**,[3] **and Haipeng Li**[4]

[1]*College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China*
[2]*Computer School, Hubei University of Arts and Science, Xiangyang 441000, China*
[3]*School of Fundamental Science and Engineering, Waseda University, Tokyo 169-8050, Japan*
[4]*Capinfo Company Ltd, Beijing 100010, China*

Correspondence should be addressed to Xiaoyan Zhang; zhangxy@xust.edu.cn

Received 12 January 2022; Revised 11 February 2022; Accepted 28 February 2022; Published 12 April 2022

Academic Editor: Yan Huo

With the wide application of the Internet of Things (IoT) in real world, the impact of the security on its development is becoming incrementally important. Recently, many advanced technologies, such as artificial intelligence (AI), computational intelligence (CI), and deep learning method, have been applied in different security applications. In intrusion detection system (IDS) of IoT, this paper developed an adaptive differential evolution based on simulated annealing algorithm (ASADE) to deal with the feature selection problems. The mutation, crossover, and selection processes of the self-adaptive DE algorithm are modified to avoid trapping in the local optimal solution. In the mutation process, the mutation factor is changed based on the hyperbolic tangent function curve. A linear function with generation is incorporated into the crossover operation to control the crossover factor. In the selection process, this paper adopts the Metropolis criterion of the SA algorithm to accept poor solution as optimal solution. To test the performance of the proposed algorithm, numerical experiments were performed on 29 benchmark functions from the CEC2017 and six typical benchmark functions. The experimental results indicate that the proposed algorithm is superior to the other four algorithms.

1. Introduction

With the popularity of the Internet of Things (IoT) [1, 2], the applications of the IoT in various industries have gradually increased, e.g., autonomous vehicles [3, 4], the medical-care IoT [5], the satellite-based IoT [6], and the industrial Internet of Things [7]. The widespread deployments and advancements in IoT have simultaneously increased threats of security, such as online spamming [8], advanced persistent threats [9], and some malicious activities. IoT also adopts many protection measures for security, including intrusion detection system (IDS) [10], antiforensic technology [11], encryption [12–14], privacy-preserving [15] technology, etc. To further ensure the security of the IoT, various technologies such as artificial intelligence (AI), blockchain technology [16–22], security and detection mechanisms [23], and key management schemes [24] are used to enhance and optimize these protection measures. The Artificial Intelligence Internet of Things (AIoT) makes the intercommunication of various networks and systems more efficient [25–29]. Deep learning has also made many contributions to the realization of AIoT [30–32] and other field [33, 34]. Many scholars have adopted different computational intelligences, including fuzzy system, neural networks [35–37], swarm intelligence [38], differential evolution algorithm [39], and other evolutionary computation [40], to resolve differential optimization problems. The optimization and improvement of differential evolution (DE) algorithm have become a trend in the application of IoT. Xue et al. [41] adopted a self-adaptive differential evolution algorithm (SaDE) to deal with feature selection problems. In the nonuniform IoT node deployments, to solving nonlinear real-parameter problems, Ghorpade et al. [42] proposed an enhanced particle swarm optimization algorithm and adopted differential crossover quantum in this algorithm. In heterogeneous resource allocation, to minimize service cost

and service time, Fang et al. [43] proposed a dynamic multiobjective evolutionary algorithm to allocating IoT services. Iwendi et al. [44] proposed a metaheuristic optimization approach for energy efficiency in the IoT networks. Yang et al. [45] proposed an intelligent trust cloud management method for secure and reliable communication in Internet of Medical Things (IoMT) systems. Qureshi et al. [46] proposed enhanced differential evolution (EDE) and adaptive EDE algorithms to effectively improve the topology robustness of the IoT network while keeping the node degree distribution unchanged.

In 1995, Storn and Price first proposed the DE algorithm on the basis of the genetic algorithm to solve global optimization problems over continuous space [47]. As a metaheuristic algorithm, the DE algorithm utilizes the individuals in the population to present the solutions of problem and updates the individuals through the mutation operation, crossover operation, and selection operation. Due to its easy implementation, high convergence speed, and superior robustness, the DE algorithm has been widely used in many fields, including in solving dynamic optimization problems [48], constraint optimization problems [49], multiobjective optimization problems [50], and engineering design problems in practical applications [51]; it can also be used as scheduling algorithm in CPS system [52].

To achieve better improvement on the performance of the DE algorithm, many scholars have developed different optimized DE algorithms, which adopt adaptive mutation strategy and crossover strategy to optimize the mutation and crossover process. Mohamed and Suganthan [53] proposed an enhanced DE algorithm and introduced a new triangular mutation operator and two adaptive schemes to change the values of the mutation factor and crossover factor. Mohamed and Mohamed [54] proposed a new DE algorithm, namely, AGDE, to prepare two candidate pools of crossover factor and adaptively update the parameter value. Mohamed et al. [49] proposed an enhanced DE algorithm (EDDE) to solve constrained engineering optimization problems. EDDE uses individual information with different fitness function values in the population to generate a mutation vector. Wu et al. [55] realized an ensemble of three DE variants (EDEV). In each generation of the ensemble, the optimal variant is obtained by competition among three variants, and the final evolution is carried out by the optimal variant. Elquliti and Mohamed [56] proposed the nonlinear integer goal programming problem with binary and real variables and developed an improved real-binary differential evolution (IRBDE) algorithm for solving constrained optimization problems. In the developed algorithm, a new binary mutation strategy is introduced to deal with binary variables. Fu et al. [57] proposed an adaptive DE algorithm with aging leader and challenger mechanism to solve optimization problem. Sun et al. [58] proposed a hybrid adaptive DE algorithm, namely, HADE, which develops a mutation process with a disturbance factor and adjusts the crossover factor according to the fitness function values. Huynh et al. [59] added a Q-learning model to generate the values of the mutation factor and the crossover factor in the DE algorithm. Zeng et al. [60] combined the DE algorithm with the SA algorithm to generate a new individual with a Markov chain length of L time in the mutation operation.

In this study, we proposed an adaptive simulated annealing differential evolution (ASADE) algorithm based on the SA algorithm and DE algorithm. In the ASADE algorithm, the mutation factor is modified with reference to the hyperbolic tangent function curve, the crossover factor is changed to linear variation of generation, and we combine the selection operation and the Metropolis criterion of the SA algorithm. In the early evolution stages of the proposed algorithm, the mutation factor and crossover factor maintain a relatively large value, and the ability of the algorithm to get rid of a local optimal solution is enhanced. In the middle evolution stages, the values of the mutation factor and crossover factor are decreasing and the algorithm speeds up the convergence rate and obtains the trade-off between global and local abilities. In the later stage, the mutation factor and crossover factor maintain a relatively small value; the search continues until the optimal solution is found. ASADE was tested on the 2017 IEEE Evolutionary Computing Conference (IEEE CEC2017) [61] and six typical benchmark functions. The experiments and comparisons show that ASADE is superior to two typical population-based algorithms and two DE optimized algorithms.

This paper is arranged as follows: the DE algorithm and SA algorithm are introduced in Section 2. Section 3 proposes the ASADE algorithm. The experimental testing results are discussed in Section 4. Finally, the conclusions of this paper are presented in Section 5.

2. Related Algorithms

2.1. DE Algorithm. The DE algorithm is a direct search algorithm based on biological ideas to solve global optimization problems. It utilizes evolutionary process such as the mutation and crossover operation to obtain a new individual as a new solution to the optimization function. The DE algorithm focuses on the diversity of solutions and the effectiveness of convergence. Compared with other optimization algorithms, the DE algorithm has fewer control parameters, faster convergence speed, stronger robustness in optimization results, and wider application in various fields.

The DE algorithm includes four processes: initialization, mutation, crossover, and selection. In the initialization process, the initial parameters include the population size (NP), mutation factor (F), crossover factor (CR), maximum evolutionary generations (G_m), number of variables (D), and range of variables ($[x_{\min}, x_{\max}]$) in individuals. The fitness function for specific problem ($f(X)$) and the initial population (X^0) as the target population in the first generation are also obtained. In the mutation and crossover process, each individual in the target population is mutated and crossed to generate trial individual and the fitness function value of each individual can be obtained. In the selection process, by comparing the fitness function values, the better individuals between the target population and the trial population are selected to form the target population of the next

generation. Algorithm 1 presents the algorithmic process of the DE algorithm.

2.2. SA Algorithm. The SA algorithm is a stochastic intelligent optimization algorithm based on the Monte Carlo method to solve an optimized problem; the name of the SA algorithm comes from the annealing and cooling process in metallurgy [62]. The algorithm treats a feasible solution of the optimized problem as a particle in the solid. The particle will reach the final ground state in the process of cooling and annealing, and the internal energy will be reduced to the minimum value, which is similar to the process of finding the optimal value of the problem. In the process of particle cooling, at high temperature, a new state that differs significantly from the current temperature is more acceptable to be an important state, while at a low temperature, a new state with a smaller temperature difference from the current state is more inclined to accept as an important state. And as the temperature tends toward a constant, no new state can be accepted. The above criterion for accepting a new state is called the Metropolis criterion. According to the Metropolis criterion, the probability that a particle will go to equilibrium at temperature T is $e^{(-(\Delta E/(kT)))}$, where E represents the internal energy, ΔE is its changed energy, and k is the Boltzmann constant. A new solution is accepted or rejected according to this probability while finding solution of the problem. The SA process can be described in the following steps.

Step 1. Set the initial parameters, objective function f, initial temperature T, cooling function T_k, and Markov chain length L. In a single evolution, the number of iterations to generate new solutions is set by the Markov chain length.

Step 2. Choose an initial viable solution X randomly, which can be regarded as a particle in a solid. At present, the optimal solution of the objective function is $f(X)$.

Step 3. Perform the process of generating a new solution with a Markov chain length of L times, which is called the Markov process. The method of generating new solutions is as follows:

(1) Part 1

$$\begin{aligned} &X(x_1, x_2, \cdots, \cdot x_k, \cdots, x_l, \cdots x_n), \\ &X_{\text{new}}(x_1, x_2, \cdots, x_l, \cdots x_k, \cdots x_n) \end{aligned} \tag{1}$$

(2) Part 2

Assuming that the resulting new solution is $P(p_1, p_2, \cdots, p_k, \cdots, p_l, \cdots p_n)$, for each of the unknown variables, $p_i(i = 1, 2, \cdots, n)$ can be expressed as

$$p_i = \begin{cases} |x_i - 1|, & w > v, \\ x_i, & \text{otherwise}, \end{cases} \tag{2}$$

where $v \in [0, 1]$ and w is a random number between 0 and 1.

Step 4. After generating a new solution, decide whether or not to accept the new solution according to the Metropolis criterion. The criterion equation is as follows:

$$\begin{aligned} p &= \begin{cases} 1, & f(X_{\text{new}}) < f(X), \\ e^{(-(\Delta E/T_k))}, & \text{otherwise}, \end{cases} \\ T_k &= \begin{cases} -\dfrac{f(X)}{\log 0.2}, & k = 1, \\ T_{k-1} \cdot \mu, & k > 1, \end{cases} \end{aligned} \tag{3}$$

where $\Delta E = f(X_{\text{new}}) f(X)_|$ and μ is the cooling coefficient. If $p >$ rand $(0, 1)$, accept the new solution, otherwise, reject the new solution. Decrease the Markov chain length by 1, and repeat the process of producing new solutions until the Markov chain length equals to 0. Choose the solution corresponding to the greatest p value as the optimal solution of the iteration.

Step 5. Obtain the current optimal solution from Step 4, execute the cooling function T_k, and determine whether the temperature remains unchanged. If so, output the current optimal solution. If not, go to Step 3.

3. The ASADE Algorithm

The DE algorithm is a stochastic direct search evolutionary algorithm. In the process of evolution, the mutation operation and crossover operation greatly impact the diversity of the solutions. In the selection operation, different optimal solutions can affect the optimization process of the next evolution. Based on the literature [63], the ASADE algorithm is proposed.

3.1. Adaptive Mutation. Previous research [64] found that the mutation factor is closely related to the search step size. In the early stages of evolution, in the scope of the global feasible solution, a large mutation factor can search the solutions widely, the structure of the solution will be more directional and diversified, and it will be easy to get rid of the local optimal solution. In the middle and late stages of evolution, when the global optimal solution range has been found, a small mutation can help to accurately search better solutions, and the performance of the DE algorithm is more effective. According to the analysis of the range of the mutation factor, a hyperbolic tangent function between [-4,4] was adopted in this paper to adjust the value of the mutation factor, and its equation is as follows:

$$F = \frac{F_{\max} + F_{mix}}{2} + \frac{\tanh\left(-4 + 8 \cdot ((G_m - G)/G_m)\right)(F_{\max} - F_{\min})}{2}, \tag{4}$$

```
Input: the initialization parameter: NP, F,CR,G_m,D,[x_min, x_max] and f(X)
output: the optimal solution of the problem
population initialization
generation=0
for i =1 ⟶ NP do
    assign initial values to each variable in each individual X^0
    for j =1 ⟶ D do
        X^0_{i,j} = x_min + rand(0,1) · (x_max − x_min)
while generation < G_m do
    k=generation
    mutate the target population X^k
    for i =1 ⟶ NP do
        V^k_i = X^k_i + F(X^k_m − X^k_n)
    cross the mutant population V^k
    for i =1 ⟶ NP do
        k_i = rand(1,2,..., D)
        for j =1 ⟶ D do
            k=rand(0,1)
            if k < CR or j == k_i
                then U^k_{i,j} = V^k_{i,j}
            else
                U^k_{i,j} = X^k_{i,j}
    Select from the trial population U^k and the target population X^k
    if f(X^k_i) < f(U^k_i)
        then X^{k+1}_i = X^k_i
    else X^{k+1}_i = U^k_i
    generation = generation+1
obtain X^{Gm}_best and f(X^{Gm}_best)
```

Algorithm 1:The algorithmic description of DE.

where G_m is the maximum evolution generation, G is the current generation, $F_{\max}$ and $F_{\min}$ are the maximum and minimum ranges of the mutation factor, respectively. Taking the maximum evolution generation as 1000, the variation trend of the mutation factor in this paper is compared with that in the study by Sun et al. [58], and the graph is plotted in Figure 1.

In Figure 1, in the mutation process of the proposed algorithm, the mutation factor is maintained at approximately 0.75 in the first 350 generations. Through a period of global searching, the algorithm can get rid of the local optimal solution continually and find the range of the global optimal solution. In the other two cases, the rate of decreasing the mutation factors at the early stages of evolution is fast, which shortens the algorithm's global search time and makes the global search less effective. From 350 to 700 generations, the mutation factor of the proposed algorithm decreases from 0.75 to 0.25, and at the same time, the search scope also reduces from global to local. After 700 generations, the mutation factor remains at 0.25. After performing a local search for a sufficiently long period of time, the better optimal solution is found in the local scope.

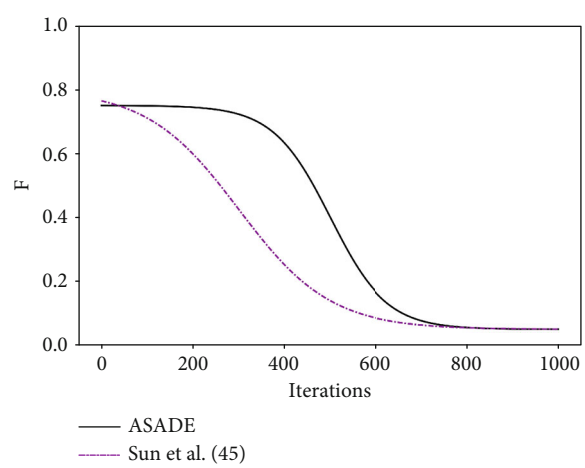


Figure 1: Change tendency curve of the mutation factor.

3.2. Adaptive Crossover. In the DE algorithm, the mutation factor has a great impact on the global search, while the crossover factor can increase the diversity of solutions and has the ability to affect the search range, but the influence of the crossover factor is slightly smaller. To improve the efficiency of the solution, we introduced a linear function with the number of generations in equation (5) to express the crossover factor. The value range of the crossover factor CR is $\mathrm{CR}_{\min} < \mathrm{CR} < \mathrm{CR}_{\max}$.

$$\mathrm{CR} = \mathrm{CR}_{\max} - \frac{G(\mathrm{CR}_{\max} - \mathrm{CR}_{\min})}{G_m}. \tag{5}$$

In the early stage of evolution, the value of the crossover factor is relatively large, and the diversity of feasible solutions is relatively rich, while in the later stage of evolution, the crossover factor gradually decreases. According to the crossover process in the DE algorithm, the probability that the value randomly generated between [0, 1] is smaller than the crossfactor decreases, the probability that the crossover vector selects the parent vector increases, and the diversity of the population decreases.

3.3. SA Selection. In the selection process of the DE algorithm, the fitness function values of individuals are compared between the trial population and the target population, and individuals with small fitness function values are selected to form the target population in next generation. After the selection process, the individual with the smallest fitness function value is the optimal value obtained by this evolution. In the process of selection, it is easy to ignore a poor solution in the trial population. To consider the impact of a poor solution could produce more diverse mutation vectors, affecting the value of optimal solution. In this paper, the Metropolis criterion of the SA algorithm is introduced in selection process to accept a poor solution as the individual in the next population. The selection of the poor solution can make the algorithm get rid of a local optimal solution in the evolutionary process and mutate in a wider direction in the next evolutionary process. The selection operation with Metropolis criterion in the proposed algorithm is defined as follows:

If $p_i > \text{rand}\,(0, 1)$,

$$X_i^{k+1} = U_i^k. \tag{6}$$

Else

$$X_i^{k+1} = X_i^k \tag{7}$$

End where the equation of p_i is as follows:

$$p_i = \begin{cases} 1 & f\left(X_i^k\right) < f\left(U_i^k\right), \\ e^{\left(-\frac{\Delta E}{T_k}\right)}, & \text{otherwise}, \end{cases} \tag{8}$$

where $E = |f(X_i^k) - f(U_i^k)|$ and T_k is the temperature in the kth generation; the initial temperature and cooling function are set as follows:

$$T_k = \begin{cases} -\dfrac{f(X_{\text{best}})}{\log 0.2}, & k = 1, \\ T_{k-1} \cdot \mu, & k > 1, \end{cases} \tag{9}$$

where k is the number of generation and μ is the cooling coefficient. The temperature decreases with evolution increment until the temperature remains constant, and each drop is related only to the value of the previous temperature.

4. Numerical Experiment and Result Analysis

The experiment in this paper used a 64-bit Windows 10 operating system. The processor is an Intel(R) Core (TM) i5-5200U CPU @ 2.20 GHz with an Intel(R) HD Graphics 5500 GPU. Python 3.5.2 is selected as the experimental code language, and the experiment is run in PyCharm software to complete the experimental process.

4.1. Experiment Setup. The performance of the ASADE algorithm in this paper was tested on the IEEE Congress on Evolutionary Computation17 test suite (CEC2017). The CEC2017 test includes 29 benchmark functions, a detailed introduction, and description of CEC2017, and its specific functions can be found in [61]. The test dimensions (D) of functions are $D = 10$, $D = 30$, $D = 50$, and $D = 100$, respectively, and the solution error $fi = f(x) - f(x^*)$ is regarded as the objective function and the fitness function, with $f(x)$ as the calculated optimal solution and $f(x^*)$ as the known optimal solution. The maximum number of fitness evaluations was $D * 10000$. The objective function values of each benchmark function are calculated over 51 runs.

The statistics results including the best, mean, and standard deviation (Std) for all dimensions are presented in Tables 1 and 2.

To compare the results of different algorithms on test functions, this paper used the Friedman test and Wilcoxon signed-rank test to analyze and compare the solution quality. The two tests use $\alpha = 0.05$ as the significance level. The Friedman test generates the final ranks of different algorithms on test functions' the Wilcoxon signed-rank test compares the specific differences between two algorithms for test functions of CEC2017. Comparing the solution solved by the former algorithm and the comparison algorithm, "R+" is the sum of ranks for the functions in the first algorithm solutions that are more than the second algorithm solutions in the row, "R-" is the sum of ranks for the opposite situation, a plus (+) sign indicates the function number of CEC2017 in which the first algorithm solutions are more than the second algorithm solutions, a minus (-) sign indicates the function number of CEC2017 in the opposite situation, and the approximation (≈) presents the number of the remaining functions. p values less than the significance level are marked in italic. SPSS 26.0 was used as an experimental tool for the statistical tests.

4.2. ASADE Parametric Study. The ASADE algorithm optimized the mutation, crossover, and selection processes in the DE algorithm. To analyze the impact of the adaptive mutation, adaptive crossover, and SA selection on the performance of the ASADE algorithm, experiments were conducted. Three different versions of ASADE were tested and compared against the proposed version on 29 functions of CEC2017 on $D = 10$, $D = 30$, $D = 50$, and $D = 100$.

(1) *Version 1.* To test the individual effect of adaptive mutation on the performance of the ASADE algorithm, an ASADE version with adaptive crossover, SA selection, and a basic mutation strategy was

Table 1: The results of the ASADE algorithm for $D = 10$ and $D = 30$ on CEC2013.

Function	$D = 10$				$D = 30$			
	Best	Worst	Mean	Std	Best	Worst	Mean	Std
f1	6.68E − 05	2.61E + 00	2.21E − 01	4.67E − 01	4.02E − 01	1.91E + 01	3.21E + 01	4.64E + 01
f2	1.13E − 10	3.17E − 02	1.86E − 03	6.00E − 03	1.59E + 00	1.17E + 02	5.58E + 01	2.25E + 00
f3	1.48E + 00	3.42E + 00	2.48E + 00	4.98E − 01	8.57E + 01	1.18E + 02	8.88E + 01	5.50E + 00
f4	9.95E − 01	1.19E + 01	4.99E + 00	2.22E + 00	1.80E + 01	7.47E + 01	3.89E + 01	1.20E + 01
f5	0.00E + 00	4.09E − 10	1.39E − 11	5.70E − 11	1.14E − 13	1.10E − 06	2.93E − 08	1.53E − 07
f6	1.10E + 01	2.01E + 01	1.36E + 01	1.91E + 00	2.24E + 01	1.32E + 02	6.39E + 01	2.20E + 01
f7	1.00E + 00	9.01E + 00	4.15E + 00	1.99E + 00	2.01E + 01	7.06E + 01	4.09E + 01	1.30E + 01
f8	0.00E + 00	9.23E − 08	3.13E − 09	1.35E − 08	0.00E + 00	1.35E − 06	3.02E − 08	1.86E − 07
f9	1.02E + 02	1.07E + 03	4.46E + 02	1.28E + 02	1.61E + 03	5.16E + 03	3.28E + 03	8.81E + 02
f10	2.07E − 01	4.24E + 00	2.09E + 00	9.82E − 01	5.67E + 00	9.44E + 01	4.39E + 01	3.29E + 01
f11	1.99E + 01	2.86E + 02	1.20E + 02	6.74E + 01	2.45E + 04	7.75E + 05	1.62E + 05	1.65E + 05
f12	1.15E + 00	8.32E + 00	5.93E + 00	1.29E + 00	4.52E + 01	2.92E + 02	1.95E + 02	5.47E + 01
f13	0.00E + 00	1.99E + 00	3.31E − 01	5.07E − 01	3.14E + 01	7.60E + 01	5.66E + 01	1.09E + 01
f14	5.80E − 02	1.53E + 00	4.29E − 01	3.91E − 01	1.11E + 01	4.76E + 01	2.90E + 01	8.78E + 00
f15	2.42E − 02	3.81E + 01	2.89E + 00	6.33E + 00	2.77E + 01	1.43E + 03	6.28E + 02	2.86E + 02
f16	3.67E − 06	2.05E + 01	1.73E + 00	4.24E + 00	2.77E + 01	5.56E + 02	2.38E + 02	1.40E + 02
f17	2.29E − 03	8.63E − 01	1.66E − 01	1.76E − 01	9.92E + 01	6.16E + 02	2.19E + 02	8.37E + 01
f18	1.68E − 06	2.54E − 01	1.89E − 02	3.40E − 02	1.20E + 01	2.98E + 01	2.23E + 01	3.23E + 00
f19	0.00E + 00	3.12E − 01	1.20E − 02	6.00E − 02	2.18E + 01	6.73E + 02	2.37E + 02	1.44E + 02
f20	1.00E + 02	2.17E + 02	1.60E + 02	5.37E + 01	2.14E + 02	2.67E + 02	2.41E + 02	1.34E + 01
f21	8.28E − 11	1.01E + 02	8.89E + 01	3.15E + 01	1.00E + 02	5.32E + 03	3.09E + 03	1.32E + 03
f22	3.04E + 02	3.16E + 02	3.09E + 02	2.66E + 00	3.67E + 02	4.15E + 02	3.90E + 02	1.15E + 01
f23	1.00E + 02	3.49E + 02	3.36E + 02	3.32E + 01	4.40E + 02	5.49E + 02	4.71E + 02	2.22E + 01
f24	3.98E + 02	3.98E + 02	3.98E + 02	1.07E − 01	3.83E + 02	3.87E + 02	3.87E + 02	4.66E − 01
f25	3.00E + 02	3.00E + 02	3.00E + 02	2.07E − 07	1.18E + 03	1.70E + 03	1.45E + 03	1.21E + 02
f26	3.87E + 02	3.91E + 02	3.89E + 02	5.80E − 01	4.62E + 02	5.37E + 02	4.95E + 02	1.28E + 01
f27	3.00E + 02	5.84E + 02	3.13E + 02	4.58E + 01	4.03E + 02	1.83E + 03	4.63E + 02	2.24E + 02
f28	2.31E + 02	2.50E + 02	2.38E + 02	5.04E + 00	3.74E + 02	8.81E + 02	5.39E + 02	1.10E + 02
f29	4.22E + 02	8.83E + 05	1.54E + 05	3.14E + 05	2.83E + 03	9.25E + 03	5.84E + 03	1.49E + 03

experimentally investigated. This version was called ASADE-1

(2) *Version 2*. To test the individual effect of adaptive crossover on the performance of the ASADE algorithm, an ASADE version with adaptive mutation, SA selection, and a basic crossover strategy was experimentally investigated. This version was called ASADE-2

(3) *Version 3*. To test the individual effect of SA selection on the performance of the ASADE algorithm, an ASADE version with adaptive mutation, adaptive crossover, and basic selection process was experimentally investigated. This version was called ASADE-3

The statistical test results of the ASADE algorithm against its alternate versions (ASADE-1, ASADE-2, and ASADE3) on CEC2017 are presented in Tables 3 and 4. Table 3 shows the average ranks of four ASADE versions calculated by the Friedman test. In the table, the p values obtained by the Friedman test for each dimension are 0.003, 0.000, 0.000, and 0.000, which are all less than 0.05. It can be drawn that the performance of these ASADE versions has a significant difference. Compared with ASADE-1, ASADE-2, and ASADE-3, the ranks of 2.02, 1.24, 1.17, and 1.10 obtained by the ASADE algorithm for all dimensions are all the smallest, and the mean rank value 1.38 is also the smallest, which proves that ASADE is better than the other three algorithms in all dimensions. In addition, ASADE-3 ranks second, followed by ASADE-2 and ASADE-1. This proves that the adaptive modified mutation factor in the mutation process plays a key role in the ASADE algorithm. The ASADE algorithm integrates three optimization strategies to obtain the best optimization effect.

Table 2: The results of the ASADE algorithm for $D = 50$ and $D = 100$ on CEC2013.

Function	$D = 50$				$D = 100$			
	Best	Worst	Mean	Std	Best	Worst	Mean	Std
f1	$1.26E+02$	$3.03E+03$	$9.53E+02$	$1.02E+02$	$9.95E+03$	$1.11E+03$	$1.05E+03$	$4.15E+02$
f2	$2.62E+01$	$5.43E+02$	$4.02E+02$	$8.57E+01$	$3.94E+02$	$4.96E+02$	$4.35E+02$	$4.39E+01$
f3	$4.84E+01$	$2.08E+02$	$1.54E+02$	$5.81E+01$	$7.81E+02$	$8.49E+02$	$8.22E+02$	$2.44E+01$
f4	$8.46E+01$	$1.17E+02$	$1.03E+02$	$1.26E+01$	$7.19E-02$	$1.51E-01$	$9.22E-02$	$2.70E-02$
f5	$5.68E-13$	$2.20E-01$	$4.43E-02$	$7.92E-02$	$9.47E+02$	$9.75E+02$	$9.62E+02$	$1.02E+01$
f6	$1.13E+02$	$1.81E+02$	$1.49E+02$	$2.48E+01$	$7.94E+02$	$8.64E+02$	$8.28E+02$	$2.66E+01$
f7	$5.17E+01$	$1.36E+02$	$8.67E+01$	$2.66E+01$	$4.40E+01$	$8.31E+02$	$4.15E+02$	$2.67E+02$
f8	$1.34E-08$	$4.64E+02$	$8.68E+01$	$1.70E+02$	$3.12E+04$	$3.17E+04$	$3.14E+04$	$2.12E+02$
f9	$3.37E+03$	$8.03E+03$	$5.61E+03$	$1.79E+03$	$8.64E+04$	$1.60E+05$	$1.22E+05$	$2.54E+04$
f10	$4.58E+01$	$6.04E+01$	$5.35E+01$	$4.38E+00$	$4.12E+04$	$5.45E+05$	$1.73E+05$	$1.83E+04$
f11	$1.86E+05$	$4.45E+06$	$2.27E+06$	$1.53E+06$	$1.74E+06$	$2.31E+07$	$1.10E+07$	$8.12E+06$
f12	$2.95E+03$	$1.92E+04$	$1.01E+04$	$6.02E+03$	$4.49E+06$	$1.53E+07$	$9.21E+06$	$3.41E+06$
f13	$1.23E+02$	$1.67E+02$	$1.51E+02$	$1.43E+01$	$6.84E+03$	$1.12E+05$	$3.62E+04$	$3.46E+04$
f14	$5.68E+01$	$2.04E+02$	$1.24E+02$	$5.71E+01$	$7.85E+03$	$8.91E+03$	$8.39E+03$	$3.85E+02$
f15	$4.96E+02$	$2.12E+03$	$1.24E+03$	$5.96E+02$	$4.97E+03$	$5.45E+03$	$5.22E+03$	$1.53E+02$
f16	$5.92E+02$	$9.40E+02$	$7.71E+02$	$1.08E+02$	$2.63E+07$	$5.46E+07$	$3.94E+07$	$1.14E+07$
f17	$2.05E+04$	$6.49E+04$	$3.70E+04$	$1.53E+04$	$6.20E+03$	$1.18E+06$	$2.18E+05$	$4.29E+05$
f18	$6.45E+01$	$1.37E+02$	$8.33E+01$	$2.54E+01$	$5.49E+03$	$6.06E+03$	$5.80E+03$	$1.79E+02$
f19	$4.73E+02$	$1.64E+03$	$1.00E+03$	$4.05E+02$	$9.58E+02$	$1.08E+03$	$1.02E+03$	$4.26E+01$
f20	$2.76E+02$	$3.18E+02$	$2.99E+02$	$1.69E+01$	$3.04E+04$	$3.28E+04$	$3.18E+04$	$8.73E+02$
f21	$4.98E+03$	$8.62E+03$	$7.37E+03$	$1.25E+03$	$6.45E+02$	$1.10E+03$	$8.61E+02$	$1.72E+02$
f22	$4.94E+02$	$5.67E+02$	$5.24E+02$	$2.36E+01$	$1.47E+03$	$1.72E+03$	$1.63E+03$	$8.55E+01$
f23	$5.39E+02$	$7.02E+02$	$6.11E+02$	$5.45E+01$	$1.03E+03$	$1.37E+03$	$1.22E+03$	$1.15E+02$
f24	$4.81E+02$	$5.80E+02$	$5.36E+02$	$3.51E+01$	$9.64E+03$	$1.17E+04$	$1.08E+04$	$7.72E+02$
f25	$1.86E+03$	$2.57E+03$	$2.20E+03$	$2.82E+02$	$7.47E+02$	$9.07E+02$	$8.31E+02$	$5.70E+01$
f26	$5.28E+02$	$8.99E+02$	$6.51E+02$	$1.34E+02$	$1.17E+04$	$1.44E+04$	$1.31E+04$	$9.65E+02$
f27	$4.59E+02$	$5.14E+02$	$4.83E+02$	$2.22E+01$	$3.87E+03$	$6.19E+03$	$5.29E+03$	$7.38E+02$
f28	$4.29E+02$	$9.16E+02$	$7.37E+02$	$1.71E+02$	$1.41E+04$	$1.01E+05$	$5.78E+04$	$2.85E+04$
f29	$6.26E+05$	$8.53E+05$	$7.00E+05$	$7.39E+04$	$9.95E+05$	$1.11E+06$	$1.05E+06$	$4.15E+04$

Table 4 presents the comparison results between ASADE and the other three versions on different dimensions according to the Wilcoxon test. The best results are distinguished in italic. In the table, the p value of the comparison between ASADE and ASADE-1 is less than 0.05 in each dimension, "R+" is less than "R-" and the function number of "-" is more than the function number of "+." ASADE is significantly better than ASADE-1 in performance. Additionally, the p values of the comparisons of ASADE with ASADE-3 and ASADE-2 on $D = 10$ are 0.545 and 0.940, respectively, both of which are greater than 0.05. However, as the dimension increases, the p value tends toward 0.000. This indicates that the optimization of the adaptive crossover process and the SA selection process has an increasing influence on the performance of the ASADE algorithm. The final comparison results are recorded in the last column of the table; a plus (+) sign indicates that the former algorithm is superior to the compared algorithm. According to the last column, ASADE is improved to the compared algorithm in 83% of the rows.

4.3. Comparison against State-of-the-Art Algorithms. The proposed ASADE algorithm was compared with four evolutionary algorithms, i.e., PSO, DE, HADE [58], and adaptive DE with disturbance factor algorithm (ADE-D) [65] on six typical benchmark functions; the functions are presented in Table 5. These benchmark functions have many local optimal values, a large search space, and strong deception. The function values are all positive, and the global optimal values of these functions are all zero; then, the fitness

Table 3: Average ranks calculated by the Friedman test for ASADE, ASADE-1, ASADE-2, and ASADE-3 across all problems and all dimensions using CEC2017.

Algorithm	$D = 10$	$D = 30$	$D = 50$	$D = 100$	Mean rank	Rank
ASADE	2.02	1.24	1.17	1.10	1.38	1
ASADE-1	3.21	3.83	3.79	3.79	3.66	4
ASADE-2	2.52	2.76	2.72	2.76	2.69	3
ASADE-3	2.26	2.17	2.31	2.34	2.27	2
Friedman p value	*0.000*	*0.000*	*0.000*			

Table 4: Results of the Wilcoxon signed-rank test between ASADE and the three compared versions across all problems and all dimensions using CEC2017.

D	ASADE vs.	R+	R-	p value	+	≈	-	Dec.
	ASADE-1	72.00	334.00	*0.003*	5	1	23	+
10	ASADE-2	221.00	214.50	0.940	13	0	16	≈
	ASADE-3	140.00	185.00	0.545	9	4	16	≈
	ASADE-1	0.00	435.00	*0.000*	0	0	29	+
30	ASADE-2	2.00	433.00	*0.000*	1	0	28	+
	ASADE-3	68.00	367.00	*0.001*	6	0	23	+
	ASADE-1	2.00	433.00	*0.000*	1	0	28	+
50	ASADE-2	28.00	407.00	*0.000*	2	0	27	+
	ASADE-3	29.00	406.00	*0.000*	2	0	27	+
	ASADE-1	2.00	433.00	*0.000*	1	0	28	+
100	ASADE-2	3.00	432.00	*0.000*	1	0	28	+
	ASADE-3	5.00	430.00	*0.000*	1	0	28	+

Table 5: Six typical benchmark functions.

Function name	Expression	Variable range
Sphere	$f_1(x) = \sum_{i=l}^{D} x_i^2$	$x_i \cdot \in [-5.12, 5.12]$
Rastrigin	$f_2(x) = \sum_{i=l}^{D} \left(x_1^2 - 10\cos(2\pi x_i) + 10\right)$	$x_i \cdot \in [-5.12, 5.12]$
Salomon	$f_3(x) = -\cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2} + 1$	$x_i \in [-100, 100]$
Griewank	$f_4(x) \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$x_i \in [-600, 600]$
Ackley	$f_5(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	$x_i \in [-32, 32]$
Levy	$f_6 = (\sin \pi w_1)^2 + \sum_{i=1}^{D-1} (w_i - 1)^2\left[1 + 10\sin^2(\pi w i + 1)\right] + (w_D - 1)^2\left[1 + \sin^2(2\pi w_D)\right], w_i = 1 + \frac{x_i - 1}{4}$	$x_i \in [-10, 10]$

function is defined as the function itself. The closer the solution is to zero, the closer it is to the global optimal value. The results including the best (BST), worst (WST), and average (AVG) values and the number of evolution generations (NEG) that reach the specified convergence precision for each function are recorded in Table 6, the values smaller than 10^{-8} are taken as zero, and the smallest values are marked in italic. The parameter setting of DE, PSO, HADE, and ADE-D can be found in original paper, and the parameter values of ASADE are NP = 100, $G_m = D * 1000$, $F_{\max} = 0.8$, $F_{\min} = 0.05$, $\mathrm{CR}_{\max} = 1$, and $\mathrm{CR}_{\min} = 0.9$.

Table 6: Comparative results of benchmark functions for $D = 30$.

Function	Value type	DE	PSO	HADE	ADE-D	ASADE
Sphere	BST	$7.71E-04$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	WST	$2.23E-03$	$1.34E-08$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	AVG	$1.35E-03$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	NEG	589	379	615	165	134
Rastrigin	BST	$7.77E+01$	$7.06E+01$	$6.92E-04$	$2.81E-06$	$0.00E+00$
	WST	$1.20E+02$	$1.57E+02$	$3.99E+00$	$9.95E-01$	$8.96E-01$
	AVG	$9.82E+01$	$8.78E+01$	$1.37E+00$	$1.99E-02$	$0.69E-02$
	NEG	980	787	969	661	576
Salomon	BST	$1.01E+00$	$1.10E+00$	$2.00E-01$	$2.00E-01$	$2.00E-01$
	WST	$1.33E+00$	$1.70E+00$	$3.00E-01$	$3.00E-01$	$2.00E-01$
	AVG	$1.17E+00$	$1.18E+00$	$2.60E-01$	$2.22E-01$	$2.00E-01$
	NEG	997	680	885	937	285
Griewank	BST	$5.74E-01$	$3.20E-02$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	WST	$8.69E-01$	$3.20E-02$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	AVG	$7.26E-01$	$3.20E-02$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	NEG	962	598	838	302	236
Ackley	BST	$1.53E+01$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	WST	$1.87E+01$	$1.34E+00$	$4.91E-07$	$0.00E+00$	$0.00E+00$
	AVG	$1.77E+01$	$8.08E-01$	$1.49E-07$	$0.00E+00$	$0.00E+00$
	NEG	978	745	975	272	191
Levy	BST	$1.85E-02$	$2.64E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	WST	$6.30E-02$	$1.62E+01$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	AVG	$4.11E-02$	$7.38E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	NEG	882	908	758	221	121

Table 7: Average ranks calculated by the Friedman test for ASADE, DE, PSO, HADE, and ADE-D on $D = 10$, $D = 30$, and $D = 50$.

Algorithm	$D = 10$	$D = 30$	$D = 50$	Mean rank	Rank
ASADE	2.00	1.92	1.75	1.89	1
DE	4.67	3.67	3.08	3.81	4
PSO	3.83	5.00	5.00	4.61	5
HADE	2.33	2.50	3.00	2.61	3
ADE-D	2.17	1.92	2.17	2.09	2
Friedman p value	*0.001*	*0.001*	*0.001*		

In Table 6, the solutions for all benchmark functions of the ASADE algorithm are the smallest and only ASADE obtains the optimal solutions on the Rastrigin and Salomon function. The NEG of the ASADE algorithm on six functions are 134, 576, 236, 191, 121, and 285; among them, the probability of finding the global optimal solution before 500 generations is 83.33%, while the probability of DE, PSO, HADE, and ADE-D is 0%, 16.67%, 0%, and 66.67%. Table 7 shows the outcome of the Friedman test. The average ranks of ASADE on 10, 30, and 50 dimensions are 2.00, 1.92, and 1.75, respectively. The mean rank of ASADE is 1.89, which is the smallest among the five algorithms. The second and third best algorithms are ADE-D and HADE, with the mean rank as 2.09 and 2.61.

When $D = 30$, the convergent tendency curves of ASADE and other four compared algorithms for six benchmark functions are depicted in Figure 2. In Figure 2, the rate of searching the best objection function value of ASADE is faster than other algorithms; compared with ADE-D, ASADE firstly finds the optimal value in the solutions of all functions. On Rastrigin and Ackley functions, DE, PSO, and HADE easily fall into local optimal solutions, while ASADE continually presents monotonic downward trend until it find the smallest solution. Therefore, we can conclude that compared with DE, PSO, HADE, and ADE-D, ASADE has faster convergence speed and more accurate solution in the process of solving these functions.

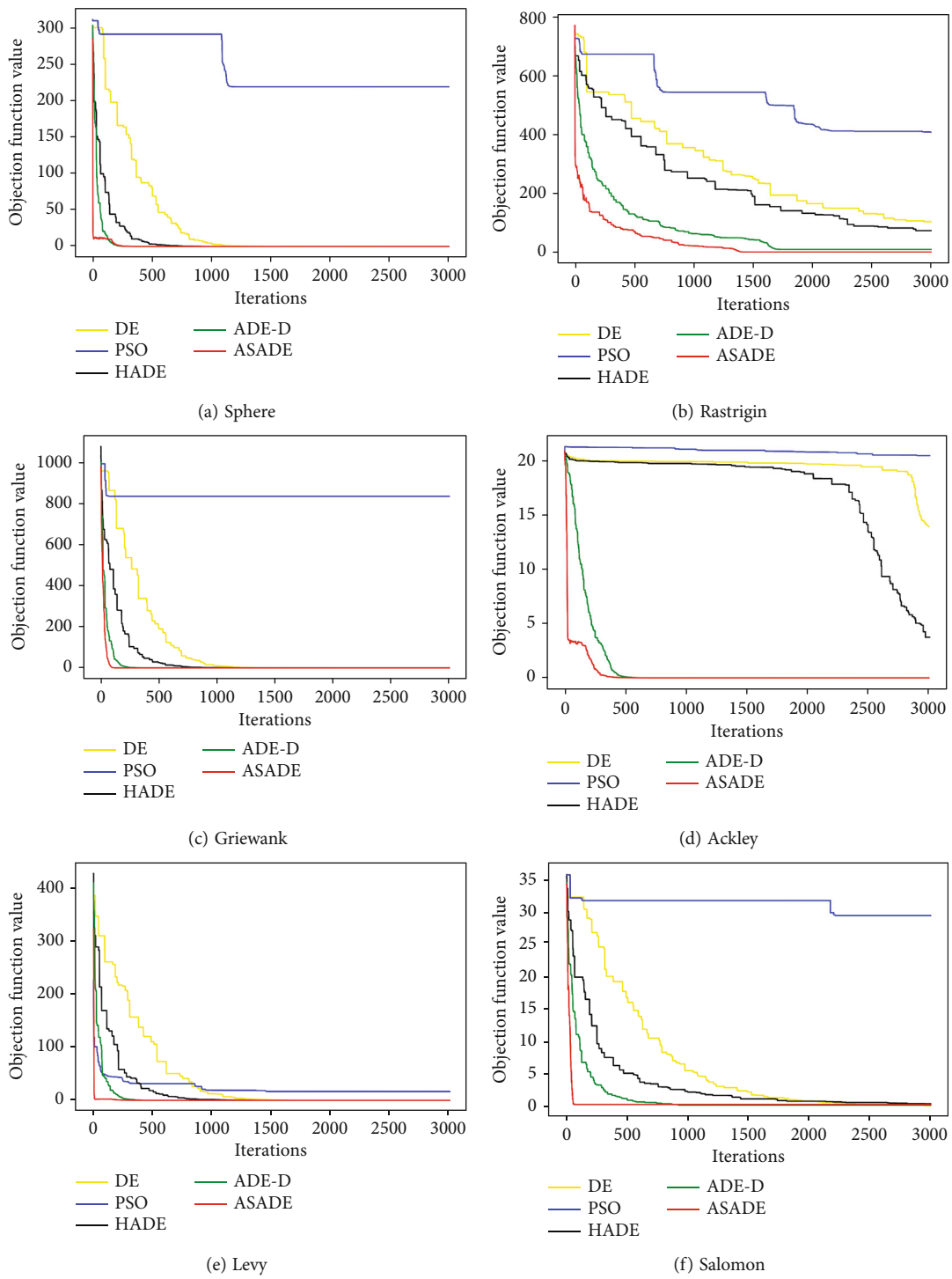


Figure 2: The convergent tendency curve of differential algorithms on typical benchmark functions for $D = 30$.

5. Conclusion

In the development of Internet of Things (IoT), intrusion detection systems (IDS) play a vital role in data security. The IDS dataset has dimensional problems of irrelevant and redundant, and feature selection is employed to reduce dimensions. An adaptive simulated annealing differential evolution algorithm (ASADE) is proposed to generate multiple candidate solutions to find the global optima in the feature selection process. The ASADE algorithm optimized the basic DE algorithm in three aspects. First, in the process of mutation, the hyperbolic tangent function is used as a variable-factor change trend function to balance global exploration and local exploitation abilities in the evolution process. Second, we adapt a linearly varied crossover factor in the crossover operation; with the increase in evolutionary

time, the crossover ability gradually changes from strong to weak. Finally, in selection process, the Metropolis criterion of the SA algorithm is used to accept a poor solution as optimal solution, which gives the DE algorithm an enhanced ability to enrich population diversity and get rid of the local optimum. To test the performance of the ASADE algorithm, we analyze the effectiveness of three ASADE versions on CEC2017 test and compare it with four improved evolution algorithms on six typical benchmark functions. The experimental results demonstrate that the performance of the ASADE algorithm is improved compared with other algorithms. In the future, the population reduction strategy, success-history slots, and a uniform distribution or a Cauchy distribution for the parameters can be employed to the ASADE algorithm to improve the performance of the algorithm. In addition, more and more security problems of IoT Ecosystem could be transformed into nonlinear real-parameter optimization problems and the ASADE algorithm could be applied to resolve them with high accuracy and fast convergence.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Fundamental Research on Safe and High-Efficiency Mining at Large Inclined Longwall Faces (No. 51634007), the Hubei Natural Science Foundation under grant 2021CFB156, and the Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research (KAKENHI) under grant JP21K17737.

References

[1] N. M. Balamurugan, S. Mohan, M. Adimoolam, A. John, and W. Wang, "DOA tracking for seamless connectivity in beamformed IoT-based drones," *Computer Standards & Interfaces*, vol. 79, article 103564, 2022.

[2] H. Li, Q. Zheng, W. Yan, R. Tao, X. Qi, and Z. Wen, "Image super-resolution reconstruction for secure data transmission in Internet of Things environment," *Mathematical Biosciences and Engineering*, vol. 18, no. 5, pp. 6652–6671, 2021.

[3] L. Tan, K. Yu, L. Lin, G. Srivastava, J. C. Lin, and W. Wei, "Speech emotion recognition enhanced traffic efficiency solution for autonomous vehicles in a 5G-enabled space-air-ground integrated intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2830–2842, 2022.

[4] F. Ding, K. Yu, Z. Gu, X. Li, and Y. Shi, "Perceptual enhancement for autonomous vehicles: restoring visually degraded images for context prediction via adversarial training," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 1–12, 2021.

[5] D. Wang, Y. He, K. Yu, G. Srivastava, L. Nie, and R. Zhang, "Delay sensitive secure NOMA transmission for hierarchical HAP-LAP medical-care IoT networks," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[6] L. Zhen, Y. Zhang, K. Yu, N. Kumar, A. Barnawi, and Y. Xie, "Early collision detection for massive random access in satellite-based internet of things," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 5184–5189, 2021.

[7] Y. Gong, L. Zhang, R. Liu, K. Yu, and G. Srivastava, "Nonlinear MIMO for industrial internet of things in cyber–physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5533–5541, 2021.

[8] Z. Guo, K. Yu, A. Jolfaei, F. Ding, and N. Zhang, "Fuz-Spam: label smoothing-based fuzzy detection of spammers in Internet of Things," *IEEE Transactions on Fuzzy Systems*, p. 1, 2021.

[9] K. Yu, L. Tan, S. Mumtaz et al., "Securing critical infrastructures: deep-learning-based threat detection in IIoT," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 76–82, 2021.

[10] A. Fatani, M. A. Elaziz, A. Dahou, M. A. A. AI-Qaness, and S. Lu, "IoT intrusion detection system using deep learning and enhanced transient search optimization," *IEEE Access*, vol. 9, pp. 123448–123464, 2021.

[11] F. Ding, G. Zhu, Y. Li, X. Zhang, P. K. Atrey, and S. Lyu, "Anti-forensics for face swapping videos via adversarial training," IEEE Transactions on Multimedia, 2021.

[12] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin, and G. Srivastava, "An efficient ciphertext-policy weighted attribute-based encryption for the Internet of Health Things," *IEEE Journal of Biomedical and Health Informatics*, p. 1, 2021.

[13] C. Feng, K. Yu, M. Aloquail, M. Alazab, Z. Lv, and S. Mumtaz, "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13784–13795, 2020.

[14] D. Xu, K. Yu, and J. A. Ritcey, "Cross-layer device authentication with quantum encryption for 5G enabled IIoT in Industry 4.0," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[15] Y. Sun, J. Liu, K. Yu, M. Alazab, and K. Lin, "PMRSS:privacy-preserving medical record searching scheme for intelligent diagnosis in IoT healthcare," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1981–1990, 2022.

[16] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K. K. R. Choo, "Blockchain-based cross-domain authentication for intelligent 5G-enabled internet of drones," *IEEE Internet of Things Journal*, p. 1, 2021.

[17] W. Wang, H. Xu, M. AlazabT, T. R. Gadelallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for IIoT devices," *IEEE transactions on industrial informatics*, p. 1, 2021.

[18] L. Tan, K. Yu, N. Shi, C. Yang, W. Wei, and H. Lu, "Towards secure and privacy-preserving data sharing for COVID-19 medical records: a blockchain-empowered approach," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 271–281, 2022.

[19] K. Yu, L. Tan, C. Yang et al., "A blockchain-based Shamirs threshold cryptography scheme for data protection in industrial Internet of Things settings," *IEEE Internet of Things Journal*, p. 1, 2021.

[20] W. Wang, C. Qiu, Z. Yin et al., "Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet of Things Journal*, p. 1, 2021.

[21] L. Tan, N. Shi, K. Yu, M. Aloqaily, and Y. Jararweh, "A blockchain-empowered access control framework for smart devices in green Internet of Things," *ACM Transactions on Internet Technology*, vol. 21, no. 3, pp. 1–20, 2021.

[22] C. Feng, B. Liu, K. Yu, S. K. Goudos, and S. Wan, "Blockchain-empowered decentralized horizontal federated learning for 5G-enabled UAVs," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3582–3592, 2022.

[23] M. Vashishtha, P. Chouksey, D. S. Rajput et al., "Security and detection mechanism in IoT-based cloud computing using hybrid approach," *International Journal of Internet Technology and Secured Transactions*, vol. 11, no. 5/6, pp. 436–451, 2021.

[24] K. Yu, M. Arifuzzaman, Z. Wen, D. Zhang, and T. Sato, "A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 8, pp. 2072–2085, 2015.

[25] T. Guo, K. Yu, M. Aloqaily, and S. Wan, "Constructing a prior-dependent graph for data clustering and dimension reducton in the edge of AIoT," *Future Generation Computer Systems*, vol. 128, pp. 381–394, 2022.

[26] K. Yu, Z. Guo, Y. Shen, W. Wang, J. C. Lin, and T. Sato, "Secure artificial intelligence of things for implicit group recommendations," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2698–2707, 2021.

[27] L. Xu, X. Zhou, X. Li, R. H. Jhaveri, T. R. Gadekallu, and Y. Ding, "Mobile collaborative secrecy performance prediction for artificial IoT networks," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[28] L. Tan, K. Yu, F. Ming, X. Cheng, and G. Srivastava, "Secure and resilient artificial Intelligence of Things: a honeyNet approach for threat detection and situational awareness," *IEEE Consumer Electronics Magazine*, p. 1, 2021.

[29] X. Shang, L. Tan, K. Yu, J. Zhang, K. Kaur, and M. M. Hassan, "Newton-interpolation-based zk-SNARK for Artificial Internet of Things," *Ad Hoc Networks*, vol. 123, p. 102656, 2021.

[30] K. Yu, L. Tan, L. Lin, X. Cheng, Z. Yi, and T. Sato, "Deep-learning-empowered breast cancer auxiliary diagnosis for 5GB remote e-health," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 54–61, 2021.

[31] F. Ding, G. Zhu, M. Alazab, X. Li, and K. Yu, "Deep-learning-empowered digital forensics for edge consumer electronics in 5G HetNets," *IEEE Consumer Electronics Magazine*, vol. 11, no. 2, pp. 42–50, 2022.

[32] L. Tan, K. Yu, A. K. Bashir et al., "Toward real-time and efficient cardiovascular monitoring for COVID-19 patients by 5G-enabled wearable medical devices: a deep learning approach," *Neural Computing and Applications*, pp. 1–14, 2021.

[33] H. Li, M. Zhang, Z. Yu, Z. Li, and N. Li, "An improved pix2pix model based on Gabor filter for robust color image rendering," *Mathematical Biosciences and Engineering*, vol. 19, no. 1, p. 86101, 2022.

[34] H. Li, Q. Zheng, J. Zhang, Z. Du, Z. Li, and B. Kang, "Pix2-Pix-based grayscale image coloring method," *Journal of Computer-Aided Design & Computer Graphics*, vol. 33, no. 6, pp. 929–938, 2021.

[35] H. I. Ahmed, N. A. Elfeshawy, S. F. Elzoghdy, H. S. Elsayed, and O. S. Faragallah, "A neural network-based learning algorithm for intrusion detection systems," *Wireless Personal Communications*, vol. 97, no. 2, pp. 3097–3112, 2017.

[36] Q. Zhang, K. Yu, Z. Guo et al., "Graph neural networks-driven traffic forecasting for connected Internet of Vehicles," *IEEE Transactions on Network Science and Engineering*, p. 1, 2021.

[37] H. Li, Q. Zheng, X. Qi et al., "Neural network-based mapping mining of image style transfer in big data systems," *Computational Intelligence and Neuroscience*, vol. 2021, 11 pages, 2021.

[38] I. Ahmad and F. E. Amin, "Towards feature subset selection in intrusion detection," in *Proceedings of 2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference*, pp. 69–74, Beijing, 2014.

[39] A. A. Aburomman and M. B. Ibne Reaz, "A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems," *Information Sciences*, vol. 414, pp. 225–246, 2017.

[40] S. Rastegari, P. Hingston, and C.-P. Lam, "Evolving statistical rulesets for network intrusion detection," *Applied Soft Computing*, vol. 33, pp. 348–359, 2015.

[41] Y. Xue, W. Jia, X. Zhao, and W. Pang, "An evolutionary computation based feature selection method for intrusion detection," *Security and Communication Networks*, vol. 2018, 10 pages, 2018.

[42] S. N. Ghorpade, M. Zennaro, B. S. Chaudhari, R. A. Saeed, H. Alhumyani, and S. Abdel-Khalek, "Enhanced differential crossover and quantum particle swarm optimization for IoT applications," *IEEE Access*, vol. 9, pp. 93831–93846, 2021.

[43] S. S. Fang, Z. Y. Chai, and Y. L. Li, "Dynamic multi-objective evolutionary algorithm for IoT services," *Applied Intelligence*, vol. 51, no. 3, pp. 1177–1200, 2021.

[44] C. Iwendi, P. K. R. Maddikunta, T. R. Gadekallu, K. Lakshmanna, A. K. Bashir, and M. J. Piran, "A metaheuristic optimization approach for energy efficiency in the IoT networks," *Software: Practice and Experience*, vol. 51, no. 12, pp. 2558–2571, 2021.

[45] L. Yang, K. Yu, S. X. Yang, C. Chakraborty, Y. Liu, and T. Guo, "An intelligent trust cloud management method for secure clustering in 5G enabled Internet of Medical Things," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[46] T. N. Qureshi, N. Javaid, A. Almogren, A. U. Khan, H. Almajed, and I. Mohiuddin, "An adaptive enhanced differential evolution strategies for topology robustness in internet of things," *International Journal of Web and Grid Services*, vol. 18, no. 1, pp. 1–33, 2022.

[47] R. Storn and K. Price, *Differential evolutiona simple and efficient adaptive scheme for global optimization over continuous spaces*, Technical Report TR-95-012. ICSI, 1995.

[48] S. Das, A. Mandal, and R. Mukherjee, "An adaptive differential evolution algorithm for global optimization in dynamic environments," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 966–978, 2014.

[49] A. W. Mohamed, A. K. Mohamed, E. Z. Elfeky, and M. Saleh, "Enhanced directed differential evolution algorithm for solving constrained engineering optimization problems," *International Journal of Applied Metaheuristic Computing*, vol. 10, no. 1, pp. 1–28, 2019.

[50] M. H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, and H. Faris, "MTDE: an effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems," *Applied Soft Computing Journal*, vol. 97, p. 106761, 2020.

[51] X. Yu, C. Li, and J. Zhou, "A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios," *Knowledge-Based Systems*, vol. 204, p. 106209, 2020.

[52] Y. Peng, A. Jolfaei, and K. Yu, "A novel real-time deterministic scheduling mechanism in industrial cyber-physical systems for energy internet," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[53] A. W. Mahamed and P. N. Suganthan, "Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation," *Soft Computing*, vol. 22, no. 10, pp. 3215–3235, 2018.

[54] A. W. Mahamed and A. K. Mahamed, "Adaptive guided differential evolution algorithm with novel mutation for numerical optimization," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 2, pp. 253–277, 2019.

[55] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, and P. N. Suganthan, "Ensemble of differential evolution variants," *Information Sciences*, vol. 423, pp. 172–186, 2018.

[56] S. A. H. ElQuliti and A. W. Mahamed, "A large-scale nonlinear mixed-binary goal programming model to assess candidate locations for solar energy stations: an improved real-binary differential evolution algorithm with a case study," *Journal of Computational and Theoretical Nanoscience*, vol. 13, no. 11, pp. 7909–7921, 2016.

[57] C. Fu, C. Jiang, G. Chen, and Q. Liu, "An adaptive differential evolution algorithm with an aging leader and challengers mechanism," *Applied Soft Computing*, vol. 57, pp. 60–73, 2017.

[58] Z. Sun, N. Wang, Y. Bi, and D. Strinivasan, "Parameter identification of PEMFC model based on hybrid adaptive differential evolution algorithm," *Energy*, vol. 90, pp. 1334–1341, 2015.

[59] T. Huynh, D. Do, and J. Lee, "Q-learning-based parameter control in differential evolution for structural optimization," *Applied Soft Computing*, vol. 107, no. 11, article 107464, 2021.

[60] Y. Zeng, J. Zhang, L. Peng, and L. Wang, "Hybrid differential evolution supported by simulated annealing and its application in integrated joint replenishment-delivery problems," *Application Research of Computers*, vol. 35, no. 4, pp. 1037–1041, 2018.

[61] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, *Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-Parameter numerical optimization*, Nanyang Technological University, Jordan University of Science and Technology and Zhengzhou University, Tech. Rep, 2016.

[62] Y. Lu, Y. Lin, Q. Peng, and Y. Wang, "Summary of simulated annealing algorithm improvement and parameter exploration," *College Mathematics*, vol. 31, no. 6, pp. 96–103, 2015.

[63] Q. Yan, R. Ma, Y. Ma, and J. Wang, "Adaptive simulated annealing particle swarm optimization algorithm," *Journal of Xidian University*, vol. 48, no. 4, pp. 1–9, 2021, (in Chinese).

[64] W. Deng, J. Xu, Y. Song, and H. Zhao, "Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem," *Applied Soft Computing*, vol. 100, p. 106724, 2020.

[65] Z. Sun, Y. Ling, H. Qu, Z. Sun, and F. Wu, "An adaptive DE algorithm based fuzzy logic anti-swing controller for overhead crane systems," *International Journal of Fuzzy Systems*, vol. 22, no. 6, pp. 1905–1921, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9584894, 13 pages
https://doi.org/10.1155/2022/9584894

Research Article

Convolution Neural Network-Based Sensitive Security Parameter Identification and Analysis

Hyunki Kim,[1] **Donghyun Kim**,[2] **and Okyeon Yi**[3]

[1]*Department of Financial Information Security at Kookmin University, Seoul 02707, Republic of Korea*
[2]*Department of Computer Science at Georgia State University (GSU), Atlanta, GA, USA*
[3]*Department of Information Security Cryptology and Mathematics at Kookmin University, Seoul 02707, Republic of Korea*

Correspondence should be addressed to Okyeon Yi; oyyi@kookmin.ac.kr

Received 9 December 2021; Revised 2 February 2022; Accepted 10 February 2022; Published 1 April 2022

Academic Editor: Liran Ma

When the logic of a pseudo-random number generator is followed by standards, the security of the generator depends on the entropy sources used in the seed construction, which are constructed through noise sources; thus, the noise sources are extremely important in cryptographic applications. The operation process of a cryptographic random number generator can be divided into an entropy collection stage and a pseudo-random number generation stage. To ensure the security, the noise sources used to construct the entropy source must be securely collected. If a cryptographic analyst physically acquires a cryptographic module in the form of a black box, it may be possible to predict the cryptographic random number to be used in the future by analyzing the random number generation process of the module. Thus, even if the cryptographic algorithms and protocols are securely designed and implemented without vulnerabilities, if the random number generation process is analyzed, the cryptographic system may be exposed. Noise sources can be identified through information generated when they are collected. If the noise sources can be identified during the stage of collecting sources that provide unpredictability to the cryptographic module during random number generation, future values may be predicted according to the data characteristics. We identify noise sources that are used as entropy sources with our convolution neural network model. Therefore, we establish attack scenarios in which random numbers can be analyzed by identifying data used in the model learned through this study when a random cryptographic module is obtained.

1. Introduction

Modern cryptography is applied under the assumption that secure random numbers are used. Random numbers are used as security parameters in cryptographic modules or cryptosystems such as encryption keys, nonces, and cryptographic salts [1–3]. Therefore, if the security of random numbers is not guaranteed, the entire cryptographic system is vulnerable. Because the vulnerability of the algorithm for generating random numbers significantly affects the security of the module or the entire system, it is more useful than the vulnerability of the encryption algorithm itself in terms of attacking the actual cryptographic system. That is, the random number is an essential element for the secure use of cryptographic functions such as confidentiality, authentication, access control, and nonrepudiation [4].

A "random number" used for cryptographic purposes has to be generated from the output of a cryptographic random number generator consisting of an entropy source collection stage and a pseudo-random number generation stage. The entropy source is the data used to generate the seed, which is the input to the deterministic random number generator.

Assuming that a secure deterministic random number generator (also known as the pseudo-random number algorithm) is used, the security of the cryptographic random number generator depends on the "unpredictability of the entropy sources" [5]. Therefore, to evaluate its security, it is necessary to verify the security of the input (seed) combined with the entropy sources. Typically, the security of a noise source to be used as an entropy source is verified based on entropy, independence, and statistical

tests. In addition, according to international random number-related standards, the seed, which is the input of the deterministic random number generator, has to be used in combination with various noise sources, and thus, the security of the noise sources is extremely important [4, 5].

By contrast, cryptographic random number generators can be classified into an entropy collection stage and a pseudo-random number generation stage, as mentioned above. To ensure the security of each stage, an evaluation of the noise sources used in the construction of the entropy sources applied to generate the pseudo-random number should be conducted [6]. If a cryptographic analyst physically acquires a cryptographic module in the form of a black box whose inside is unknown, it may be possible to predict the cryptographic random number to be used in the future by analyzing the random number generation process of the module. Then, even if all cryptographic algorithms and protocols are securely designed and implemented without vulnerabilities, all cryptographic systems using that random number generation process may be exposed to vulnerabilities. Accordingly, we present methods for identifying various entropy sources used as inputs to generate cryptographically secure random numbers by analyzing them with convolution neural network (CNN) models. This paper shows that CNN models can identify various data acquired during the collection stage of various entropy sources of cryptographic modules. Therefore, we can establish attack scenarios that can analyze a random number by identifying the entropy sources used in the model learned through this study as a cryptographic module under specific environments.

We establish three attack scenarios by considering the case where the cryptographic module to be analyzed is physically obtained and the case where it is not. If the attacker has physically acquired the module, the security parameters are identified by observing the output values. Otherwise, they are identified by observing the power consumption of the module.

The contributions to this paper are as follows.

1.1. Proposal of Cryptographic Module Analysis Methodology. This paper presents a method to analyze cryptographic modules from a new perspective. The main directions of the existing cryptographic module analysis methods were full investigation of implementation vulnerabilities, memory leak inspection, and reverse engineering. However, a new direction can be added to the analysis method through the CNN model-based security parameter identification methods proposed in this paper.

1.2. Suggestion of the Security Parameter Identification Model according to the Output Values of Cryptographic Modules. We accumulate several sensitive security parameters generated by software modules of Windows 10 and image them. Then, the security parameters are identified by predicting the images with the CNN model. As a result of that, we establish an attack scenario.

1.3. Suggestion of the Security Parameter Identification Model according to the Power Waveform Generated by the Cryptographic Module. We image the power consumption waveforms when security parameters are generated in the firmware cryptographic module in the Arduino board. And then, by predicting the images with the CNN model, parameters are identified. As a result of that, we establish an attack scenario.

2. Background

2.1. Random Number. Cryptographic random numbers are used as various security parameters. They must satisfy the unpredictability, unbiasedness, and interbit independence and provide more than the security strength recommended in [1, 7]. A deterministic random number generator (DRBG) is used to ensure unbiasedness and independence between bits, and various entropy sources (noise sources) are collected and mixed to satisfy the unpredictability and security strength recommended by the country and then input into the DRBG [6, 8].

2.2. Randomness. Randomness in cryptography (and statistics) means that it is difficult to find predictability or patterns of certain events that are randomly selected within a defined range. A random number is equivalent to the result of tossing a coin in an unbiased and fair manner. Each side of this coin can be considered 0 or 1 bit, and once tossed, the probability of obtaining a 0 or 1 is the same, with each being executed independently [2, 3]. Therefore, tossing a coin in succession can be considered the result of a perfect random number generator. Because all parts of a sequence are independent of each other, already generated parts cannot affect the generation of the next sequence, and it is also impossible to predict the next sequence from the current sequence. If a sequence is independent for each bit and has the same probability (also known as independent and identically distributed), it is said to be random.

2.3. Unpredictability. In an environment using cryptography, random numbers should never be predictable. If even a slight predictability will result in a different outcome. Depending on the situation, it is possible to attack the security protocol by predicting parameters such as encryption key, nonce, and salt. In addition, the ciphertext of a secure standard algorithm can be decrypted with the predicted key. For this reason, countries set strict standards to prevent random numbers from being predicted. Random numbers generated through cryptographic algorithms should not only be unable to predict future values based on current values, they should also be unable to infer past records based on such values. In general, random number generation algorithms are open to the public, and thus, the input values of the algorithms have to remain secret to ensure their unpredictability [2, 3].

2.4. Security Strength (Bits of Security). Security strength is closely related to the number of operations required to break a cryptographic algorithm or system, expressed in bits, and the National Institute of Standards and Technology (NIST)

standard stipulates that one should be selected from the set of {80, 112, 128, 192, 256} the selected [7]. A security strength of 80 bits should not be used, however, as it is no longer considered sufficiently secure. In addition, it is currently stipulated that more than 112 bits should be provided.

2.5. Random Number Generation. Because cryptosystems frequently use random numbers, they should be able to generate them extremely quickly. It is impossible to obtain a random number for cryptographic use without a specific algorithm, and even if the output value changes every time without an algorithm and it is possible to generate an unpredictable value (true random number), the speed is sufficiently slow to avoid satisfying the availability. For this reason, in the cryptography field, a deterministic algorithm is used to generate random numbers at a high speed [4].

The algorithm used in this situation is called DRBG, which receives seeds and generates random numbers according to the determined logic. Therefore, when the same seed is set in the same DRBG, these two algorithms output random numbers that match each other exactly. Because of this characteristic, if the user of an algorithm applies predictable values as seeds, an attacker can accurately determine the random number that the user generates. Thus, the seed that initializes or updates the DRBG must be different each time to be unpredictable. In general, these consist of noise sources that are commonly obtained in the operating environment [6]. Because most noise sources have entropy bias, it is inappropriate to use the collected noise source as a seed without any significant change. NIST and the Telecommunications Technology Association (TTA) recommend applying conditioning to equally distribute the biased entropy. The lower bound of the security of the DRBG depends on the entropy provided by the noise sources and the way in which the seed is constructed. It is therefore necessary to configure the seeds with entropy sources that guarantee the target security strength of the cryptographic module [9, 10].

2.6. Noise Sources, Entropy Sources, and Seed. When using the DRBG as a secure standard algorithm, its security depends entirely on the seeds used as input. Therefore, the composition of the seed is extremely important. What is needed are noise sources or entropy sources, because these must consist of values that change unpredictably and nondeterministically. The "noise source" encompasses a function or device that generates nondeterministic data, and the "entropy source" includes a function or device that combines the noise source and an algorithm (as the conditioning) that mitigates its poor characteristics [1, 8].

As shown in Figure 1, noise sources have various collection methods (digital or analog data) and types (e.g., function calls and sensing). In general, as noise sources that can be collected by the operating system, aspects such as the running time, mouse cursor coordinates, keyboard stroke, disk status, and interrupt request information can be used. The entropy of nondeterministic functions provided by the operating system may change depending on the environment, such as the current state of the operating system, user intervention (e.g., mouse/keyboard), and the number of active background processes. Therefore, it is recommended to use not only one type of noise source provided by the operating system but also various types together [4]. If a certain noise source has entropy characteristics, it can be used as a component of the entropy source for seed construction. If not, however, it cannot be used to configure the entropy source but can be used for the purpose of supplementing the stability of the seed.

In the server and PC level equipment, it is not difficult to accumulate more entropy than the security level because there are many noise sources that can be used as entropy sources. However, small devices such as Internet of Things (IoT) devices with firmware only or sensors that cannot install an operating system lack the ability to accept user intervention or generate unpredictable elements. Therefore, the available noise sources are extremely limited [11–13].

3. Related Work

3.1. Data Identification Using Deep Learning. Deep learning refers to machine learning technologies that construct a model to have a large number of neural layers for learning a pattern recognition problem or feature points. This is an artificial neural network (ANN) technology consisting of several hidden layers between an input layer and an output layer. With artificial intelligence (AI) and ANNs in the spotlight, deep learning is used in various fields. Representative neural network structures include an autoencoder [14] and a restricted Boltzmann machine (RBM) [15], convolutional neural network (CNN) [16], and recurrent neural network (RNN) [17]. An autoencoder and RBM are models presented in the early days when deep learning technology was proposed and have the advantage of being easy to use for self-learning. However, they exhibit limited performance owing to their characteristics. Most currently used deep learning systems are based on a CNN or an RNN, and although they require supervised learning, they show a more powerful performance and a wider range of applications. CNNs are becoming an essential technology in the field of image signal processing or computer vision [18, 19], and RNNs have shown a good performance in speech signal processing and speech recognition. CNNs are neural networks designed using the concept of the receptive field of human visual neurons [16]. They have convolutional layers, pooling layers, and rectified linear units (ReLU) as key elements. They are characterized by kernels that take inputs and outputs in a specific form and represent the weights as small-sized filters. That is, regardless of the number of dimensions of the input data, if the kernel size is small, the neural layers can be defined with an extremely small number of weights. The pooling neural layer reduces the size of the data by summarizing several output values, and through this process, it can eliminate the distortion of the input data. A ReLU is a nonlinear neuron with ramp function characteristics and has the effect of solving the computational burden of the "sigmoid" function and the disappearance of gradients in the backpropagation algorithm.

Although the concept of a CNN was proposed at the end of the 20th century, it was grafted with the deep learning

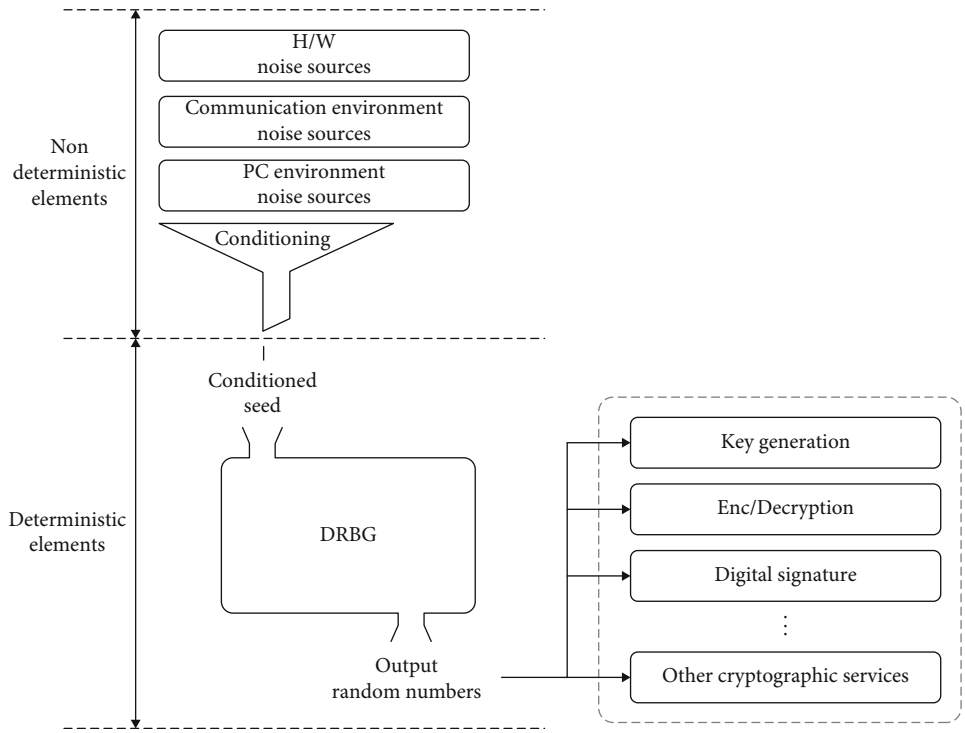


Figure 1: Roles of randomness. In the environment in which the cryptographic module operates, nondeterministic data are variously combined and passed through deterministic algorithms. The generated random numbers are then used as different security parameters.

Table 1: Windows OS noise source providing functions and their information.

Collected noise source information	Function name
Windows random number generator	CryptGenRandom()
System configuration information	GetSystemInfo()
Operating system version information	GetVersionEx()
Current process ID	GetCurrentProcessId()
Current thread ID	GetCurrentThreadId()
Memory state information	GlobalMemoryStatus()
Mouse cursor position	GetCursorPos()
CPU clock accumulated after booting	QueryPerformanceCounter()
CPU clock generated per second	QueryPerformanceFrequency()
Process heap handle	GetProcessHeap()
Elapsed time since boot	GetTickCount()
Create GUID	CoCreateGuid()

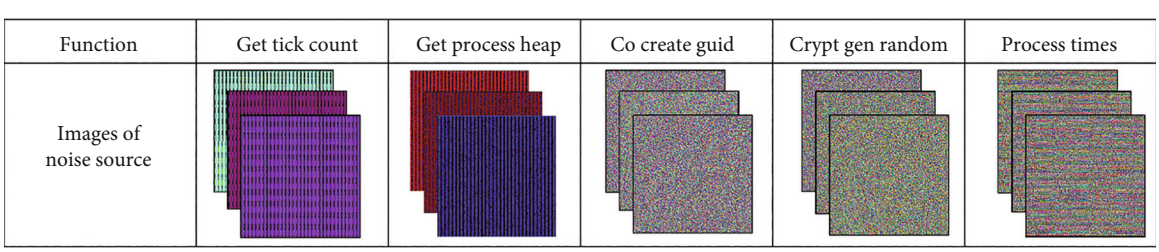


Figure 2: Samples of noise sources. The noise source functions were called several times to form each bit stream, and through them, ".bmp" files of 255×255 in size were made. Although GetTickCount and GetProcessHeap are easy to identify because of their distinct characteristics, it is difficult to tell the difference between CoCreateGuid, CryptGenRandom, and ProcessTimes.

Table 2: Estimated entropy values for noise sources by byte index through SP800-90B. Target noise sources are mainly used as entropy sources for software cryptographic modules in Windows 10. The smaller the entropy per byte (sum total/number of bytes), the more patterns appear in the image, and vice versa; it becomes increasingly difficult to recognize the pattern. The noise sources CryptGenRandom and CoCreateGuid, commonly used in Windows's software cryptographic modules, have quite high entropy values and do not differ much. Therefore, even in their images, it seems indistinguishable by the eyes.

Target noise		GetTickCount	GetProcessHeap	CoCreateGuid	CryptGenRandom	GetProcessTimes
Byte index	0	0.303	0.000	7.863	7.883	7.882
	1	0.027	0.000	7.885	7.881	7.008
	2	0.002	7.876	7.884	7.877	0.587
	3	0.000	7.852	7.882	7.882	0.001
	4		7.882	7.878	7.888	
	5		0.996	7.880	7.871	
	6		0.000	7.886	7.865	
	7		0.000	3.975	7.880	
	8			5.939	7.879	
	9			7.881	7.861	
	10			7.869	7.879	
	11			7.883	7.874	
	12			7.862	7.864	
	13			7.880	7.889	
	14			7.883	7.879	
	15			7.881	7.865	
Sum total of H		0.332	24.606	120.211	126.017	15.478
Average of min-entropy H		0.083	3.076	7.513	7.876	3.870

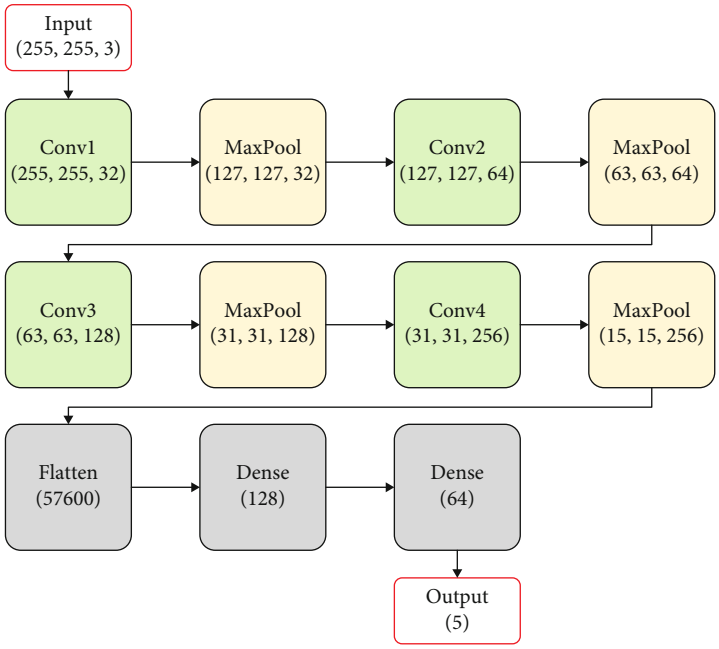


Figure 3: Structure of $CNN_{NSBased}$. It consists of four Conv2D layers, and Maxpooling is applied to each layer. The shape of the data output from each step of the model is indicated in parentheses.

paradigm during the 2012 ImageNet challenge contest and has become the mainstream of both deep learning and machine learning. Until 2011, a large number of images were classified using the existing machine learning method, and with the 2012 release of the CNN-based AlexNet [20] , it has achieved significant results, becoming the mainstream tool in image-related problems. CNN models proposed after 2012 have had an increasing number of neural layers, and

Table 3: Hyperparameters of CNN models.

Hyperparameter	$CNN_{NSBased}$	$CNN_{TraceBased}$
Filter size	{32, 64, 128, 256}	{4, 8, 16, 32}
Epoch	20	15
Output size (number of classes)	5	4
Batch size	100	100
Kernel size	5×5	5×5
Kernel initializer	he_normal	he_normal
Conv strides	(1, 1)	(1, 1)
Pooling strides	2	2
Output activation type	Softmax	Softmax
Optimizer	rmsprop	rmsprop
Loss function	categorical_crossentropy	categorical_crossentropy
Padding	Same	Same
Pooing sizes after each conv layer	(2, 2)	(2, 2)

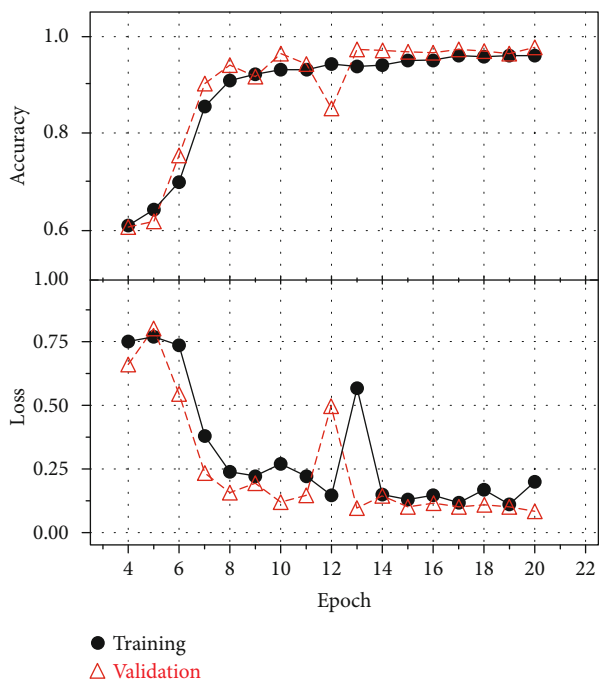


Figure 4: Result of identifying noise sources of Windows 10. In up to 8 out of 20 epochs in total, the accuracy sharply increases to approximately 0.90, and the loss rapidly decreases to approximately 0.2. After that, it gradually increases during the period and finally reaches 0.97. From epoch 21, the training loss decreases and the validation_loss tends to increase, resulting in an overfitting, and thus, the training is stopped at epoch 20.

through this process, techniques for applying many neural layers into a single CNN have been applied. In 2013, the network-in-network (NIN) concept [21] was proposed in which the filter kernel is composed of a neural network rather than a set of weights. In addition, in 2014, VGG [22] and GoogLeNet [23], which apply the technique of stacking many small convolutional layers rather than stacking few large convolutional layers, were proposed. In 2015, ResNet was proposed [24], which reduces the number of computations by learning the difference between outputs and inputs, and constructs a deeper neural network. The approach recorded an image classification error rate of 3.5% in the ImageNet challenge competition and was evaluated as achieving a higher performance than the human cognitive ability of 5%. Meanwhile, a neural network in the form of a combination of GoogLeNet and ResNet has been released, and the structure of CNNs will continue to become deeper and more complex in the future [25].

3.2. Security Evaluation of Cryptographically Secure Random Number. Methods used to evaluate the security of cryptographic random numbers can be classified into statistical randomness tests and stochastic entropy estimations [26]. In general, a statistical randomness test is a method for checking whether the final output of a random number generator is distinguished from ideal random numbers, and the entropy estimation method estimates the minimum entropy of a noise source providing unpredictability [27]. The security evaluation of cryptographic random numbers is being studied mainly by major standard organizations such as NIST in the US and the Federal Office for Information Security (BSI) in Germany [28, 29]. Appropriate probabilistic entropy estimation methods can theoretically prove the security of the entropy source. However, they are based on the assumption that the target of the estimated source and their behavior are both known [6, 27]. However, the statistical randomness test is unsuitable for discriminating noise sources, and it only judges whether the data are random numbers by focusing on the statistical properties regardless of the entropy of the target source.

4. Materials and Methods

4.1. Image Data Identification of Noise Sources Based on CNN. The final goal of this study is to identify random noise sources generated by the black box cryptographic module

True label \ Predicted label	A	B	C	D	E
A	900 (0.900)	100 (0.100)	0 (0.000)	0 (0.000)	0 (0.000)
B	58 (0.058)	942 (0.942)	0 (0.000)	0 (0.000)	0 (0.000)
C	0 (0.000)	0 (0.000)	1,000 (1.000)	0 (0.000)	0 (0.000)
D	0 (0.000)	0 (0.000)	0 (0.000)	1,000 (1.000)	0 (0.000)
E	0 (0.000)	0 (0.000)	0 (0.000)	0 (0.000)	1,000 (1.000)

The number of predicted label/ (normalized value)

A: CoCreateGuid
B: CryptGenRandom
C: GetProcessHeap
D: GetProcessTimes
E: GetTickCount

Confusion matrix

Figure 5: Confusion matrix for $CNN_{NSBased}$. This model correctly identified GetProcessHeap, GetProcessTimes, and GetTickCount for each 1,000 pieces of data. Although incorrect answers existed for CoCreateGuid and CryptGenRandom, they were identified with high probability. Each value in the table means the number of predicted labels and their normalized value.

whose interior is unknown when using a trained model. To identify the noise source functions used in the acquired module, the model has to be trained with the noise source values provided under various environments in advance. Because this paper focuses on identifying the noise sources generated by the cryptographic module, we start by examining the noise sources that can be obtained in a general PC environment and are mainly adopted in a cryptographic module.

4.1.1. Selection and Collection of Target Data. First, the target of the training is noise sources commonly used as entropy sources in the Windows 10 OS environment. Table 1 shows a part of the Windows noise source functions provided [1].

Each bit stream was made by repeatedly calling some of these functions, which were imaged and used as training data for the model. This can be visualized by converting the noise source bit streams collected several times into a bitmap form. As shown in Figure 2, there are noise sources whose patterns can be distinguished even with the naked eye (GetTickCount() and GetProcessHeap()), whereas others are difficult to find patterns (e.g., CoCreateGuid() and CryptGenRandom()). When operating the cryptographic module in various environments, even if digital noise sources generated by software or analog signals generated from hardware are gathered, any noise source data must also be digitized because they must eventually go through the logic operation of the algorithm, and thus, the bit sampled per noise source is determined within a finite range. Then, the maximum and minimum values at which the noise source is output are determined, and accordingly, when the same type of noise source is repeatedly collected several times, a specific pattern may be repeated. This feature becomes more prominent when the bit stream collected several times is imaged. Therefore, to detect patterns that humans are unable to distinguish, a CNN is applied to an image analysis of various noise sources to identify them. To train the model, a total of five noise sources (CoCreateGuid(), CryptGenRandom(), GetProcessHeap(), GetTickCount(), and GetProcessTimes()) were selected for analysis. For this purpose, 50,000 samples of $255 \times 255 \times 3$ size (width × length × number of channels) were generated for each noise source. A total of 25,000 samples (50% of the total) were used as training data, and 20,000 samples (40%) were used as validation data. Finally, 5,000 samples (10%) were used for a prediction test to check the accuracy of the trained model.

(1) CryptGenRandom. This is a random number generator provided by Microsoft Windows that generates cryptographically random data; however, a vulnerability in the function was discovered in 2007 [30, 31], and it can no longer be used as a random number generator. However, because a different value is generated every time the function is called, the data are generally used as a noise source [4].

(2) CoCreateGuid. This creates a globally unique GUID each time it is called.

(3) GetProcessHeap. When this function is called, the number of heap handles that are active in the calling process is returned.

(4) GetTickCount. When this function is called, the elapsed time in milliseconds since the system was started is returned.

(5) GetProcessTime. Various types of time can be returned, such as the start or end time of the process being used and the time the process was executed in the kernel mode.

Meanwhile, Table 2 shows the min-entropy values of the target noise sources. These are the results of the entropy estimation software [6] published in 2020 by NIST. This tool has limitations that the size of one sample can only estimate the entropy from the minimum 1-bit to the maximum 1-byte (8-bit). Since all of the target noise sources are larger than 1 byte, the final entropy "H" was determined by obtaining the entropy corresponding to the byte position of each data and adding all the values. For example, if there are a total of 3 bytes with $\text{Noise} = \text{noise}[0] \| \text{noise}[1] \| \text{noise}[2]$, each $H_{\text{noise}[0]}$, $H_{\text{noise}[1]}$, and $H_{\text{noise}[2]}$ is derived through the tool. Then, the final entropy is obtained as $H_{\text{Noise}} = H_{\text{noise}[0]} + H_{\text{noise}[1]} + H_{\text{noise}[2]}$.

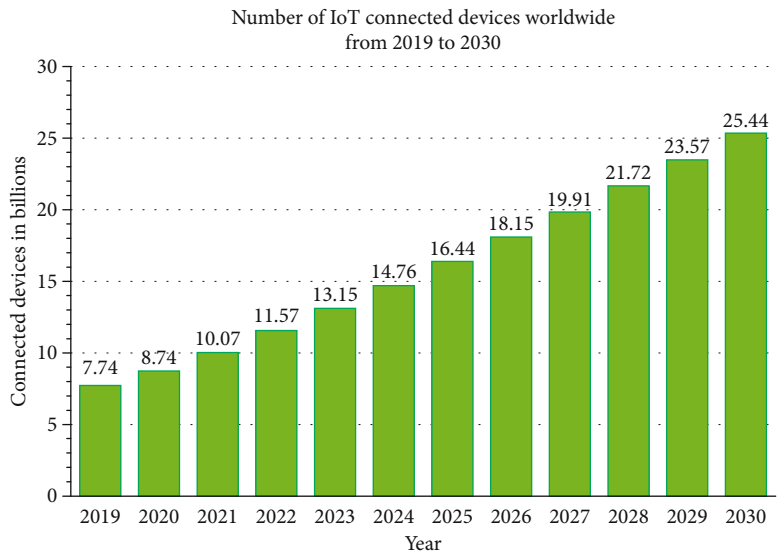


Figure 6: Number of IoT-connected devices worldwide from 2019 to 2030. It is predicted that the global use of IoT devices will nearly triple within 10 years.

Table 4: Functions that are supported by the Arduino board.

Category	Deterministic functions	Nondeterministic functions
Digital I/O	digitalRead()	
	digitalWrite()	—
	pinMode()	
Analog I/O	analogReference()	analogRead()
	analogWrite()	
T5ime	delay()	micros()
		millis()
Random numbers	random()	—
	randomSeed()	
Interrupts	—	timer0_overflow_count

4.1.2. Model Training. As shown in Figure 3, this noise source-based CNN ($CNN_{NSBased}$) goes through Conv2D four times, Maxpooling four times, and Dense three times. The filter consists of 32, 64, 128, and 256 square pixels for each conv layer, and Maxpooling is applied to reduce the number of features that can occur every time each layer is passed by half. In addition, ReLU is applied to the activation function of Conv, and in the last density layer, a softmax function is adopted to suit single index multiclassification. Moreover, RMSprop and categorical_crossentropy are used in the optimizer and loss functions, respectively, to construct a model. The model was trained to identify the noise source image through a total of 20 epochs with a batch size of 100. The details of the hyperparameters of this model are shown in Table 3.

4.1.3. Analysis of Results. As a result of training the model with images of entropy sources, the patterns of bit strings that are difficult to identify are distinguished with the naked eye with high accuracy. To check the possibility of identifying the entropy sources of this CNN model, as shown in Figure 2, bit strings (GetTickCount(), GetProcessHeap()) that are easy to detect with the naked eye and bit strings (CryptGenRandom(), CoCreatedGuid(), and ProcessTimes ()) that are difficult to identify were used as target data for learning and verification.

Assuming that the accuracy of choosing the correct answer by randomly selecting one out of five classes is 20%, the identification result of the trained model is shown in Figure 4. The identification accuracy started to increase from epoch 8 and reached approximately 97.24% at epoch 20. An overfitting occurred when epoch 21 was passed, and learning was stopped at epoch 20.

To evaluate this trained multiclass identification model, a dataset of 5,000 uncontaminated through model training was used for the prediction test. Figure 5 shows the normalized confusion matrix for the multiclass identification model results.

The results of the prediction test indicate that this model can identify noise sources with consistently high precision for GetProcessHeap, GetProcessTimes, and GetTickCount. This indicates that there are feature points that can be distinguished by the model between each noise source class. In addition, there is a percentage of instances with a misclassification between CoCreateGuid and CryptGenRandom. From this result, although the model can distinguish the two, it is inferred that their distributions are extremely similar.

Meanwhile, the $F1$ score can be derived by calculating the precision and recall, which are performance evaluation indicators of the identification model, based on the values in Figure 5. The precision and recall of CoCreateGuid (CCG) are $\text{precision}_{CCG} = 900/(900 + 58) = 0.94$ and $\text{recall}_{CCG} = 900/(900 + 100) = 0.90$, respectively, whereas those of CryptGenRandom (CGR) are $\text{precision}_{CGR} = 942/(100 + 942) = 0.90$ and $\text{recall}_{CGR} = 942/(58 + 942) = .94$, and

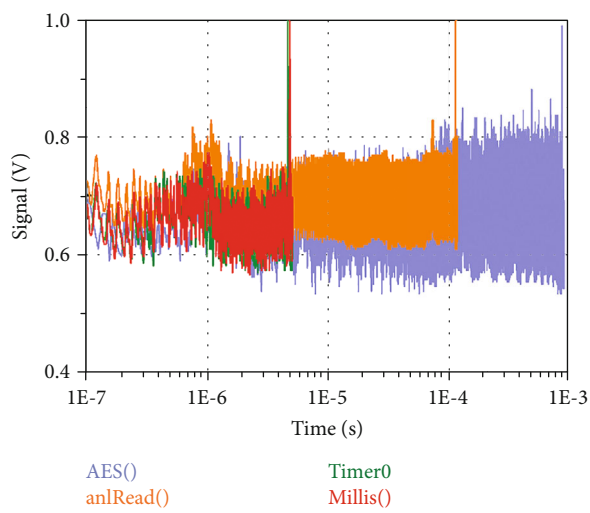


Figure 7: How to encode input for use. This is a graph in which the time of the waveform is expressed as a logarithmic scale. AES, analogRead, millis, and timer0_overflow_count (called Timer0) have the longest running times in order. Because millis and timer0 use similar clock-related interrupts, they tend to have similar shapes.

Table 5: Number of valid points in trace.

Target noise function	Total number of points in trace	Number of valid points between trigger signals
AES()	100,000	About 86,690
analogRead()	60,000	About 54,900
millis()	6,000	About 4,700
timer0_ overflow_ count	6,000	About 4,570

those of the other three classes are 1000/1000 = 1.00. The $F1$ score derived from these indicators is proximately 0.97, and thus, the model effectively identifies the noise sources.

4.2. Identification of Power Waveform Data When Noise Sources Are Generated Based on CNN. As shown in Figure 6, IoT devices are used worldwide from industrial control to consumer markets, and their number is expected to nearly triple from 8.74 billion units in 2020 to over 25.4 billion units in 2030 [32]. In particular, among the IoT devices available in 2021, Raspberry Pi and Arduino board occupy the first and second places with 44% and 28% shares, respectively, in the industrial control sector [33]. Industrial IoTs often share data between multiple parties [34, 35]. In particular, among the devices, they may upload or share privacy and sensitive information. At this time, security protocols are carried out to prevent inference attacks, etc. In this process, IoT requires random number generation [36, 37].

Even if some cryptographic analysts have not yet physically acquired a cryptographic module whose internals are unknown, if the power consumption waveform generated when the module is operating can be determined from a distance, the module can be analyzed through such waveforms [38, 39]. When applying specific functions in any module or device, the generated power consumption waveforms all differ. Some of these specific functions also include the process of generating important security parameters (resulting from noise-source generating functions). The model identified in this section is the power trace generated when generating sensitive security parameters (entropy sources) on an Arduino board. If an attacker can catch the power waveforms of a target board (e.g., a cryptographic module that has not been physically seized) from a distance, the attacker can reveal the types of security parameters used by the device and further identify them.

4.2.1. Selection and Collection of Target Data. The target board of our experiment is an Arduino Uno; in addition, the MCU is an ATmega328P based on an 8-bit operator, the clock speed is 16 MHz, and the board operates at a voltage of 5 V. Most IoT devices provide only limited functions that can only perform specific tasks to minimize the hardware volume, power, and memory, among other factors. Therefore, the random elements are so limited that they cannot be compared with the PC environment, and thus, there are few available functions. The noise sources including random elements that can be used on the Arduino board are at most functions that read the input voltage from the analog pin and the built-in timer-related functions. Thus, as a training target for the CNN model used in this experiment, power waveforms generated by the target board (when calling the noise source functions and encryption algorithms) were applied.

Table 4 shows some of the basic functions provided by the Arduino board [40]. Since the functions supported are extremely limited, good noise sources (high entropy) cannot be obtained unless there are added-on modules and sensors. Various sensors such as gyroscope, position, inertia, vibration, and acceleration can be added to the Arduino board according to the application used in the industry. There is also a method of digitizing analog values generated from newly added devices and using them as noise sources. However, since the combination of sensors added-on varies extremely depending on the board usage in each industry, this paper deals with only the functions that are basically supported. The method of obtaining a noise source by adding a module other than Arduino is out of scope of the experiment. The power consumption that occurs when three out of the four functions whose resulting value is not deterministic (analogRead(), millis(), and timer_overflow_count) are operated was collected. The representative encryption algorithm AES() was also collected for comparison [41]. In addition, the power consumption generated at that time was collected. First, in the target board, after two channels (waveform and trigger channel) are connected to the oscilloscope, a rising edge is activated just before the target algorithm operates, and a falling edge is activated immediately after operation. The learning target data of the model is collected by extracting the waveform during the trigger signal. Figure 7 is the graph, which is expressed on a logarithmic scale of the power consumption used when the target functions operate. And Table 5 shows the number of points

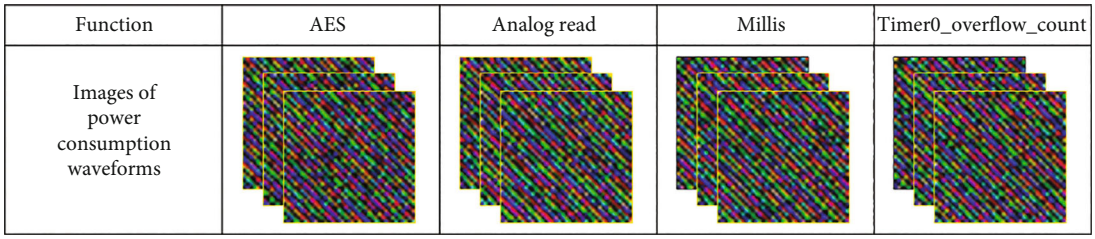


Figure 8: Image samples of each trace. Unlike when the output bit streams in Section 4.1 were imaged, the images of the traces were uncharacteristic. Therefore, it is impossible to identify with the naked eye.

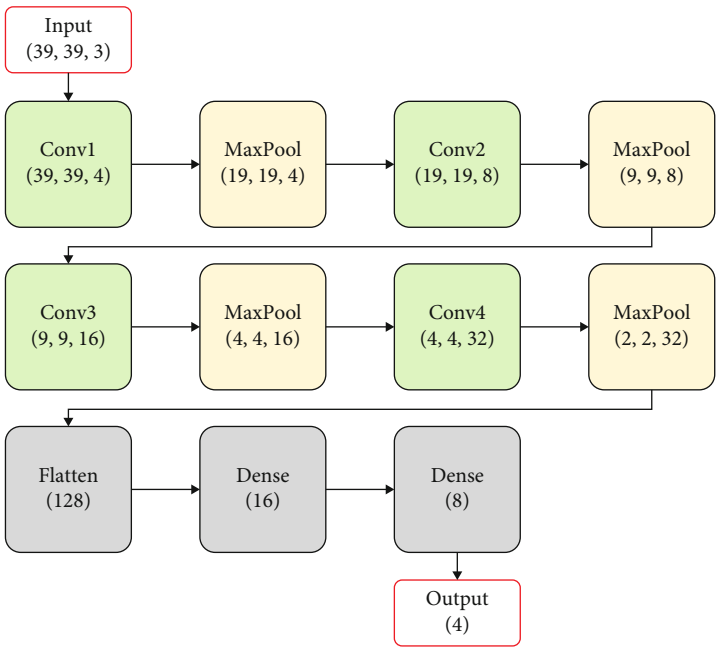


Figure 9: Structure of $CNN_{TraceBased}$. It consists of four Conv2D layers, and Maxpooling is applied to each layer. The shape of the data output from each step of the model is indicated in parentheses. The details of the hyperparameters of this model are shown in Table 3.

measured in the target power consumption waveform during the trigger signal. Because AES() and analogRead() perform many operations inside the corresponding algorithm, the running time is relatively long and there are numerous points in the trace. In addition, timer0_overflow_count and millis() require less running time and contain a small number of points compared to AES() and analogRead(). As shown in Figure 8, to train the power waveform identification model with the data of the same length, the number of the sampled data of each target noise function is determined by timer0_overflow_count because it has the minimum number of valid points during the pulse width of the trigger signal.

In addition, to train the waveform identification model in a similar way as the model $CNN_{NSBased}$, the extracted data were postprocessed as a square color bitmap file. Using $39 \times 39 \times 3 = 4{,}563$ (<4,570, which is the number of points in timer0) points for each waveform, a total of 10,116 data samples of size (39, 39, 3) (horizontal, vertical, and number of channels) were generated. Square bitmap files of color are represented as shown in Figure 8. We used 6,068 pieces (60% of the total) as training data and 3,032 pieces (30%) as validation data. In addition, 1,016 pieces (10%) were used for the prediction test to check the accuracy of the trained model.

4.2.2. Model Training. As shown in Figure 9, this CNN model ($CNN_{TraceBased}$) applies Conv2D four times, Maxpooling four times, and Dense three times, which are the same numbers as the noise source identification model $CNN_{NSBased}$. Because the pixel resolution of the image is 39×39, the filter is composed of 4, 8, 16, and 32 square pixels for each Conv layer to avoid exceeding the width and height of the image. For the same reason as the previous model, Maxpooling is applied to halve the features that can occur each time each layer is passed. The activation function of Conv applied ReLU, the last Dense layer adopted a softmax function suitable for single-index multiclassification, and the optimizer and loss functions constructed a model using RMSprop and categorical_crossentropy, respectively. It was trained to identify the image of the power consumption waveform through a total of 15 epochs with a batch size of 100.

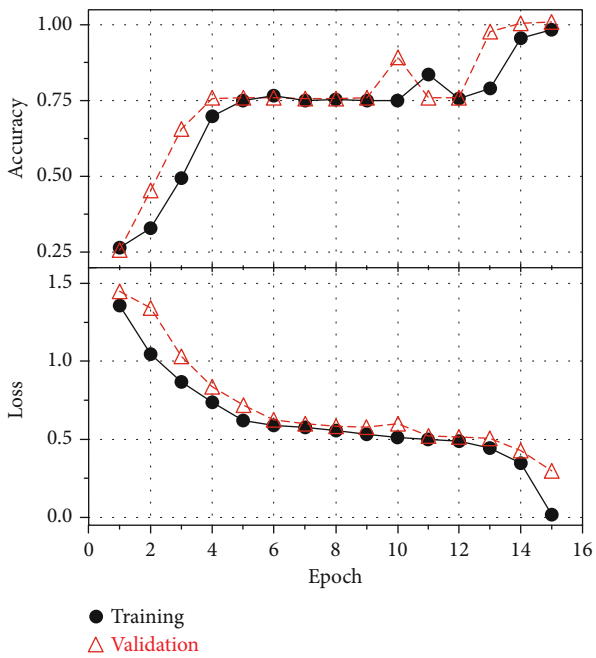


Figure 10: Results of identifying power consumption waveform of Arduino. For up to 4 out of the total 15 epochs, the accuracy sharply increases to approximately 0.75, and the loss rapidly decreases to approximately 0.75. After that, it gradually increases during the period and finally reaches 0.98. From epoch 16, the training loss decreases and the validation_loss tends to increase, resulting in an overfitting, and thus, the training was stopped at epoch 20.

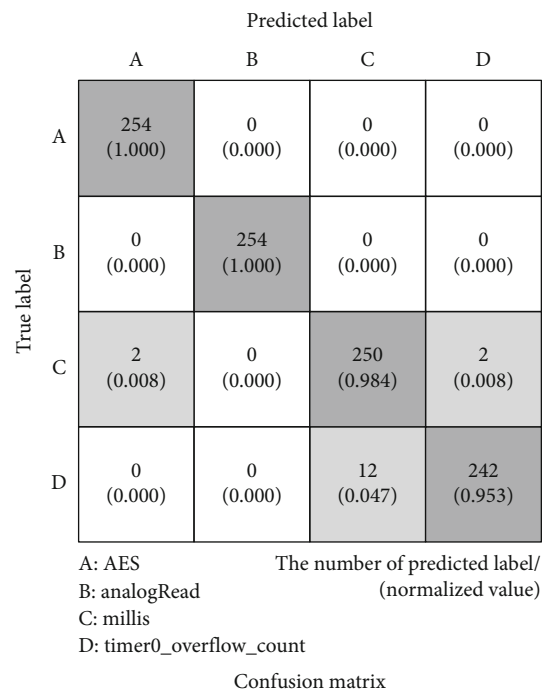


Figure 11: Confusion matrix for $CNN_{TraceBased}$. This model correctly identified analogRead for all 254 pieces of data. Although incorrect answers existed for AES, millis, and timer0, they were identified with high probability. Each value in the table indicates the number of predicted labels and their normalized value.

4.2.3. Analysis of Results. As a result of training the CNN model with the image of the power waveform generated when the noise source functions on the target board are called, the model distinguishes the pattern of bit strings that are difficult to identify with the naked eye with high accuracy. Here, AES() and analogRead() have distinct waveforms because they contain completely different logical operations internally, whereas millis() and timer0_overflow_count use similar types of clock interrupt handlers, and thus, the waveforms have similar shapes.

Therefore, the CNN model used in this experiment showed a high identification rate for AES() and analogRead() and accurately identified all prediction datasets corresponding to them. If one of the four classes is randomly selected and the accuracy of correcting the correct answer is 25%, the identification result of the trained model is shown in Figure 10. The identification accuracy gradually increased from epoch 3, and the accuracy was over 90% from epoch 14; in addition, learning was stopped at epoch 15 because an overfitting occurred from epoch 16 or higher. To evaluate this multiclass identification model, 1,016 datasets uncontaminated by training were used for the prediction tests. Figure 11 shows a confusion matrix of the results of this identification model.

The results of the prediction test indicate that this model can identify the target waveform with consistently high precision. Through this result, it can be seen that there are feature points that the CNN can distinguish among each power waveform class. In addition, there is a proportion of instances that misclassified between millis() and timer0_overflow_count. This leads to the fact that, although the model can identify the two, the power consumption according to the algorithm operation is extremely similar.

Meanwhile, based on the values in Figure 11, the precision, recall, and $F1$ score, which are performance evaluation indicators of the identification model, can be derived. The precision and recall of millis() are $\text{precision}_{\text{millis}} = 250/(250 + 12) = 0.95$ and $\text{recall}_{\text{millis}} = 250/(2 + 250 + 2) = 0.98$, respectively, and those of timer0_overflow_count (called timer0) are $\text{precision}_{\text{timer0}} = 242/(2 + 242) = 0.99$ and recal $\text{l}_{\text{timer0}} = 242/(12 + 242) = 0.95$. The $F1$ score derived by calculating the other two classes in this way is approximately 0.98, and the model effectively identifies the noise sources.

4.3. Attack Scenarios. If some cryptographic analysts physically acquire a black box-type cryptographic module whose inside is unknown, they may try to analyze it in various ways, such as through reverse engineering, a brute force attack, and a side-channel attack, to understand the cryptographic mechanism through which the module operates. Under this situation, any information related to security, such as cryptographic algorithms, protocols, and cryptographic keys used within the module, can be targeted for analysis. This also includes sensitive security parameters (such as cryptographically random numbers or the noise sources needed to generate them). By contrast, even if the

analyst did not physically acquire the cryptographic module, if the power waveform generated when the module is operating can be caught from a distance, the module can be side-channel analyzed.

If analysts do not know what noise sources are used as entropy sources in the module, the noise source identification model of this study can be used to derive the types of noise sources used in the cryptographic module. If the noise sources used among the parameters derived from the target encryption module are revealed by physically acquiring or catching a remote signal, the noise sources can be reproduced or predicted. If the noise sources found are the time- and clock-related parameters, they can be monotonically increasing and repeat the same values within a finite range ($\{0, 1\}^n$ at most).

Meanwhile, analysts can know the operating environments if the cryptographic module is physically obtained or if the target module of a remote analysis is a universally used public device. As a result, the rate of increase in clock noise sources in the environment can also be obtained without difficulty. Therefore, analysts can predict the output of a noise source that monotonically increases at a constant rate within a finite range, which can affect the security of the cryptographic system. Also, let us suppose that some IoT devices sense surrounding environmental information (e.g., temperature and illuminance) and use it as noise sources. Then, from a distance, the attacker already knows the noise source information of the IoT device (through the method of this paper), and as a result, he can manipulate the surrounding environment (such as the temperature of a heater and the brightness of a fluorescent lamp in a building) in which the target device is installed to a desired value.

4.4. Study Limitations. In order to obtain the identification effects presented in the attack scenarios section, the models must be trained in advance regardless of which cryptographic module is attacked. For the $\mathrm{CNN}_{\mathrm{NSBased}}$ model to be trained, the type of the attack target module must be clearly known. If the nationally verified cryptographic modules (e.g., CMVP, KCMVP, and JCMVP) are the target of the analysis, even if the module is not physically acquired, the device type of the module can be found on the website of the relevant verification program. However, when the target of analysis is a module used for private business or defense, it is not easy to know the type of device in advance. Meanwhile, to train the $\mathrm{CNN}_{\mathrm{TraceBased}}$ model, it must be able to remotely detect the power waveform when the attack target module is operating. This is an inherent limitation of all subchannel analysis methods. In addition, it is quite difficult to specify the waveform (e.g., the power/electromagnetic wave) of the module, and the sensors are expensive.

5. Conclusions

This paper presents a new perspective for analyzing cryptographic modules. We propose that the cryptographic system can be affected by identifying sensitive security parameters that are the starting point of cryptographic module security. We do not present a new deep learning identification model but show that anyone can analyze cryptographic modules through image identification models that are already commonly used in various fields. This is because even being able to identify a critical security parameter can lead to a fatal vulnerability.

In this paper, entropy sources, which are the basis of cryptographic random numbers, were analyzed by type through CNN image identification models. As a result, the existence of patterns that are difficult to detect with the human eye and the possibility of detecting and identifying entropy sources of unknown origin were presented.

As future studies, the following can be applied to improve the identification performance of deep learning for any noise source. The identification of noise sources in a Raspberry Pi, which has the largest market share among industrial IoT devices, suggests that additional problems are caused by such sources, and the internal configuration of the learning model should be changed. In particular, in order to improve the identifiability of security parameters, it can also be considered to apply the Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM), which are models that have different identification methods from CNNs.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the Republic of Korea's MSIT (Ministry of Science and ICT), under the High-Potential Individuals Global Training Program (2021-0-01516) supervised by the IITP (Institute of Information and Communications Technology Planning & Evaluation).

References

[1] *Information Technology-security Techniques-security Requirements for Cryptographic Modules, KS X ISO/IEC 19790*, Korean Agency for Techonology and Standards, Korea, 2015.

[2] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, FL, USA, 1st edition, 1996.

[3] D. R. Stinson, *Cryptography: Theory and Practice*, Chapman and Hall/CRC, USA, 3rd edition, 2005.

[4] H. Choi, W. Ju, H. Kim, and Y. Yeom, *Guideline for the Collection and Application of Noise Sources on Operating Systems*, TTAS.KO-12.0235/R2, Telecommunication Technology Association, Korea, 2020.

[5] H. Kim, W. Ju, H. Choi, and Y. Yeom, *Guideline for Testing noise Sources used in Software Cryptographic Modules*, TTAK.KO-12.0341, Elecommunication Technology Association, Korea, 2018.

[6] M. Turan, E. Barker, J. Kelsey, M. L. Baish, and M. Boyle, *Recommendation for the Entropy Sources used for Random Bit Generation*, NIST SP800-90B, National Institute of Standards and Technology, USA, 2018.

[7] E. Barker, *Recommendation for Key Management*, NIST SP800-57, National Institute of Standards and Technology, USA, 2020.

[8] E. Barker and J. Kelsey, *Recommendation for Random Bit Generator (RBG) Constructions*, NIST SP800-90C, National Institute of Standards and Technology, USA, 2016.

[9] Z. Gutterman, B. Pinkas, and T. Reinman, "Analysis of the Linux random number generator," IEEE S&P 2006, Berkeley, California, USA, 2006.

[10] M. Vanhoef and F. Piessens, "Predicting, decrypting, and abusing WPA2/802.11 group keys," 25th USENIX Security Symposium, Austin, TX, USA, 2016.

[11] M. P. Pawlowski, A. Jaraand, and M. Ogorzalek, "Harvesting entropy for random number generation for Internet of things constrained devices using on-board sensors," *MDPI, Sensors*, vol. 15, no. 10, pp. 26838–26865, 2015.

[12] G. Souaki and K. Halim, *Random number generation based on MCU sources for IoT application*, ATSIP'2017, Fez, Morroco, 2017.

[13] K. Wallace, K. Moran, E. Novak, G. Zhou, and K. Sun, "Toward sensor-based random number generation for mobile and IoT devices," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1189–1201, 2016.

[14] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[15] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[17] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, *Extensions of recurrent neural network language model*, IEEE ICASSP, 2011.

[18] C. Dong, C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE TPAMI*, vol. 38, no. 2, pp. 295–307, 2016.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proceedings of the IEEE CVPR*, pp. , 20153431–3440, 2015.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[21] M. Lin, Q. Chen, and S. Yan, "Network in network," (2013), https://arxiv.org/abs/1312.4400.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," (2014), https://arxiv.org/abs/1409.1556.

[23] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," (2015), https://arxiv.org/abs/1512.03385.

[25] S. Mun, S. Jang, J. H. Lee, and J. S. Lee, "Machine learning and deep learning technology trends," *Korean Institute of Communications and Information Sciences*, vol. 33, no. 10, pp. 49–56, 2016.

[26] H. Park, *A study on the security evaluation for cryptographically secure random number generators, [Ph.D. thesis]*, Dept. Financial Information Security, Kookmin Univ, Seoul, KR, 2020.

[27] J. Yang, S. Zhu, T. Chen, Y. Ma, N. Lv, and J. Lin, "Neural network based min-entropy estimation for random number generators," *Security and Privacy in Communication Networks, SecureComm*, 2018Springer, 2018.

[28] A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST SP800-22. (2010).

[29] W. Killmann and W. Schindler, "A proposal for functionally classes and evaluation methodology for true (physical) random number generators," *AIS*, 2001.

[30] L. Dorrendorf, Z. Gutterman, and B. Pinkas, "Cryptanalysis of the random number generator of the Windows operating system," *ACM Transactions on Information and System Security*, vol. 13, no. 1, pp. 1–32, 2009.

[31] "CryptGenRandom function (wincrypt.h)," Microsoft (2021), http://docs.microsoft.com/en-us/windows/win32/api/wincrypt/.

[32] A. Host, "Number of Internet of things (IoT) connected devices worldwide from 2019 to 2030Statista(2021), http://statista.com/statistics/1183457/iot-connected-devices-worldwide/.

[33] S. Higginbotham, "IoT news of the weekStacey on IoT(2021), http://staceyoniot.com/iot-news-of-the-week-for-sept-10-2021/.

[34] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019.

[35] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications (JSAC).*, vol. 38, no. 5, pp. 968–979, 2020.

[36] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[37] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering (TNSE).*, vol. 7, no. 2, pp. 766–775, 2020.

[38] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," *Advances in Cryptology*, vol. 1109, pp. 104–113, 1996.

[39] A. Golder, D. Das, J. Danial, S. Ghosh, S. Sen, and A. Raychowdhury, "Practical approaches toward deep-learning-based cross-device power side-channel attack," *IEEE Transactions on VLSI Systems*, vol. 27, no. 12, pp. 2720–2733, 2019.

[40] Language reference. Arduino, https://www.arduino.cc/reference/en/".

[41] Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, (2001).

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3096961, 8 pages
https://doi.org/10.1155/2022/3096961

Research Article

2PN: A Unified Panoptic Segmentation Network with Attention Module

Jianwen Wang and Zhiqin Liu

Southwest University of Science and Technology, Mianyang, China

Correspondence should be addressed to Zhiqin Liu; lzq@swust.edu.cn

Received 26 January 2022; Revised 6 March 2022; Accepted 14 March 2022; Published 30 March 2022

Academic Editor: Yan Huo

Comprehensive and accurate surveillance of the environment forms the basis of secure Internet of things (IoTs), the threats can be observed, and the AI services of IoT systems can be preserved. Panoptic segmentation is an efficient and popular approach for environmental surveillance based on images captured by smart sensing devices. This approach can jointly detect stuffs and things within an image and feed subsequent tasks like image detection. So far, there are many methods for panoptic segmentation which focus on extracting sophisticated visual features for segmentation. However, these efforts are both heavy on their workload and cannot clearly distinguish essential features useful for surveillance in an open environment. Therefore, this paper proposes a novel deep learning model 2PN for panoptic segmentation. The model includes a 2-way pyramid network and an attention module to learn in a more concentrated and reasonable way which enhances the feature extraction part. It strikes a balance between the computing complexity and the power of model capability. Finally, 2PN (2-way pyramid network) results are reflected on the Cityscapes dataset.

1. Introduction

Securing the functionalities and services of the Internet of things (IoT for short) systems usually request a clear awareness of the environment, such that potential threats can be observed and the whole system can be guarded. Recently, AI-powered IoTs proposed both novel services and approach-secured IoTs; IoT systems can supply the multimedia data collected by intelligent sensing devices to perform environmental surveillance. Among these pioneering attempts, image segmentation is believed to be an essential and basic aspect of surveillance and also acts as an important research direction of computer vision. The panoptic segmentation [1], which is a combination of semantic segmentation and instance segmentation, is considered a novel frontier of image segmentation. Each pixel of the image must be obtained with a semantic label or an instance label, which may jointly contribute to the understanding of the environment. This segmentation method can bring new opportunities and challenges to computer vision, especially when dealing with complicated open environments.

Generally, the scene of image segmentation consists of "stuff" and "thing." "stuff" usually defines uncountable objects or an object without a fixed shape such as sky and building. At the same time, "thing" usually defines countable objects such as cars, bikes, and pedestrians. The main object of panoptic segmentation is to jointly and wisely detect and distinguish both parts as they are usually correlated. A panoptic segmentation image is shown in Figure 1.

Current trends of panoptic segmentation usually follow the paradigms of deep convolutional networks, which can be divided into three parts: feature extraction, semantic and instance segmentation branch, and subtask fusion. Feature extraction is the head part of panoptic segmentation which receives the input image and provides information for subsequent panoptic segmentation tasks. However, current methods for feature extraction may lose some information, resulting in poor results. Fortunately, the feature

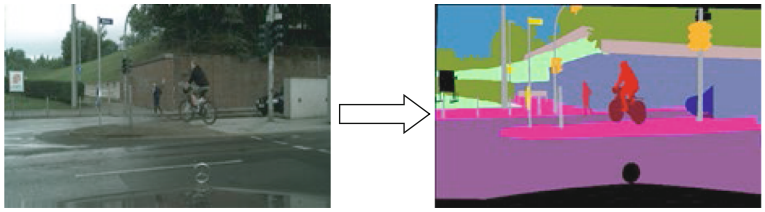


Figure 1: A panoptic segmentation image: (a) is the original image and (b) is the panoptic image. Background without fixed shapes such as sky, traffic light, and wall is "stuff." Pedestrians, cars, and bikes in this figure are "thing."

pyramid network (FPN) [2] is a method used in feature extraction. Current panoptic segmentation models such as BBFNet, PCV, and Panoptic-FPN [3–5] all adopt FPN to mitigate feature extraction. The function of FPN is to extract multiscale features, which can improve the effect of panoptic segmentation.

However, this method still has limitations. Due to the increase in the complexity of the image, only a single FPN cannot obtain more effective features. The FPN is a one-way network, which will lose local information and affect the accuracy of the thing detection part. Moreover, panoptic segmentation will change the branches of semantic and instance segmentation into different directions, and when the transmission is two-way, information will be lost due to different emphases of semantic and instance segmentation.

This paper proposes a two-way pyramid network to solve this problem. The two-way pyramid network will carry out two-way propagation of information. Compared with regular FPN, the feature pyramid model in the upsampling direction is added, which can reduce the information loss caused by the convolution of the network and improve the segmentation effect of the "thing" part. In addition, due to the bidirectional propagation of information, the two-way pyramid network can integrate multiscale features better than FPN, which will also improve the effect of the "stuff" part. Specifically, in order to collect multiscale context information, we use Atrous Spatial Pyramid Pooling (ASPP) [6].

Moreover, the distribution and importance of various features are different in image segmentation. The attention module can play an important role in panoptic segmentation. The attention module has not been applied in these panoptic segmentation models. In this work, we include the attention-based methods to enhance feature extraction and get context information based on Panoptic-DeepLab [7]. Because the feature distribution of the image is unequal, the attention module focuses on more significant features. As a result, we get 60.4%PQ on Cityscapes with the ResNet-50 backbone, getting better performance better than the baseline Panoptic-DeepLab with the ResNet-50 backbone [8].

In summary, the main contribution of the paper is as follows:

(i) A two-way pyramid network is introduced, and a novel panoptic segmentation network is designed, by which reasonable and comprehensive visual features can be extracted and applied

(ii) An attention module is designed for getting multiscale context information concentrating on pivotal parts useful for environmental surveillance

(iii) Experimental results on benchmarks show the advancement of the proposed model in an open environment

2. Materials and Methods

2.1. Related Work. Panoptic segmentation is a concept proposed by Kirillov et al. [1]. It combines the characteristics of semantic segmentation and instance segmentation. In recent years, many methods have been proposed to improve the results of panoptic segmentation.

Semantic segmentation: semantic segmentation distinguishes the regions of different categories of the input image by distinguishing the category of each pixel. Early segmentation algorithms usually used traditional algorithms such as the conditional random field and random forest. The Fully Convolutional Network (FCN) proposed by Long et al. [9] is the semantic segmentation network based on CNN. FCN replaces the full connection layer with the convolution layer. U-Net proposed by Ronneberger et al. [10] is based on FCN and effectively obtains multiscale features through the encoder-decoder structure. Zhao et al. propose PSPNet [11] network structures, which adopt the Pyramid Pooling Module (PPM). The pyramid pooling structure uses four layers of pooling, which is easier to aggregate context information than a single pooling layer. Chen et al. [6] propose the Atrous Spatial Pyramid Pooling (ASPP) module. It samples the given input in parallel at different sampling rates, and the effect is to obtain the context information of the image in different scales. Chen et al. also propose DeepLabv3+ [12] to get better semantic segmentation performance; DeepLabv3+ takes an encoder-decoder structure as a whole, which can obtain more context feature information.

Instance segmentation: instance segmentation includes object detection and semantic segmentation. Instance segmentation is proposed by Hariharan et al. [13]. Instance segmentation generates segmentation results and then detects the segmentation results. Girshick et al. [14] propose the regional convolutional neural network (R-CNN), which first makes regional candidates and then classifies objects in the selected region. Fast R-CNN [15] has greatly improved the training speed of R-CNN.

Ren et al. propose Faster R-CNN [16]. As a continuation of Fast R-CNN, this method proposes the region proposal

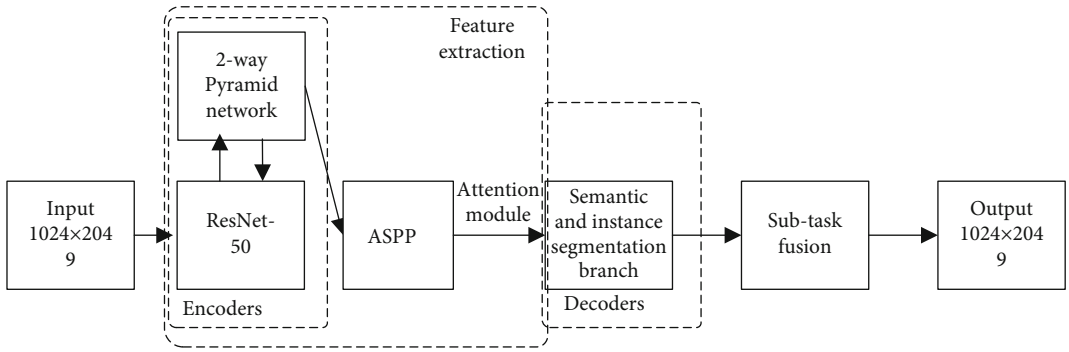


Figure 2: Overview of our architecture: ResNet-50, 2-way pyramid network, and ASPP consist of the feature extraction part of the attention module. The 2-way pyramid network as encoders transforms feature information to decoders in the semantic and instance segmentation branch.

network, which functions similar to the attention mechanism, generates target candidate boxes, and optimizes target detection. Then, Mask R-CNN proposed by He et al. [17] adds the mask mechanism on the basis of Faster R-CNN to perform parallel computing with Faster R-CNN added with FPN. Because of its accuracy and speed, this model is often used in the instance segmentation branch of panoptic segmentation. [18, 19] also introduce the semantic segmentation branch, so that each pixel can be marked. This branch of semantic segmentation is also very similar to the later semantic segmentation branch of panoptic segmentation.

Panoptic segmentation: BlitzNet proposed by Dvornik et al. [20] is considered to be the prototype of a single-stage panoptic segmentation model. It cascades object detection and semantic segmentation. Deeperlab proposed by Yang et al. [21] uses the bottom-up method and uses three parts of the mainstream panoptic segmentation. Then, Panoptic-DeepLab proposed by Cheng et al. [7] gets the best performance of panoptic segmentation. The structure of ASPP is added before Panoptic-DeepLab's semantic and instance segmentation branch, and Panoptic-DeepLab has strong expansibility. The performance of the model can be further improved by modifying the semantic and instance segmentation branch of feature extraction. FPSNet [22] uses a heuristic algorithm to make the model simpler and easier to implement. These methods are one-stage methods without using RPN. Some panoptic segmentation methods which use RPN are called two-stage methods. JSIS-Net proposed by De Geus et al. [23] uses a shared feature extractor to provide features for semantic and instance segmentation branches. TASC-Net proposed by Li et al. [24] reduces the fusion loss by adding a mask mechanism to align the "thing" categories of the semantic and instance segmentation branch. Panoptic-FPN proposed by Kirillov et al. [5] adds the feature pyramid network to help extract context information. Our model proposes two opposite FPNs to get more features from the segmentation input part.

Attention module: the spatial attention module and channel attention module are the two most commonly used modules. The channel attention module enables the neural network to automatically determine which channel is important or unimportant and then assign appropriate weight. SE (Squeeze-and-Excitation) [25] is based on the channel attention module. The spatial attention module is to find the most important part of the network for processing. Our attention module combines the spatial attention module and channel attention.

AI-empowered IOT: some research in different fields on the Internet of things focuses on datasets. [26] presents an out-of-core 3D segmentation method for large-scale image datasets on medical service. [26] introduces the novel concept of ϵ-Kernel Dataset on Wireless Sensor Networks (WSNs) and designs a distributed algorithm to satisfy the ϵ requirement. [27–29] also propose algorithms for WSNs, while our approach focuses on the Cityscapes dataset, which is the representative of the open environment of street scenes.

3. Architecture

This section first introduces the overview of our proposed model. Then, each component of the model is separately introduced in each part.

As is shown in Figure 2, the size of our model's input from the Cityscapes is 1024×2049. Our model focuses on the improvement of feature extraction and consists of the following parts: a ResNet-50 backbone. The feature will be passed into a 2-way pyramid network, which sends large-scale feature images to decoders from semantic and instance segmentation branches and produces feature maps for the ASPP part. ASPP is used for getting multiscaled features. The attention module is used to get the most important features from ASPP. The semantic and instance segmentation branch and subtask fusion part are similar to Panoptic-DeepLab. Our semantic segmentation branch and instance branch are similar to DeepLabv3+ [12]. Subtask fusion obtains the loss function and gets the results of the model.

3.1. 2-Way Pyramid Network. Figure 3 is the 2-way pyramid network, which will adopt 4 or 8x downsampling for the input large-scale feature image to obtain more detailed information, and 16 or 32x downsampling for small-scale

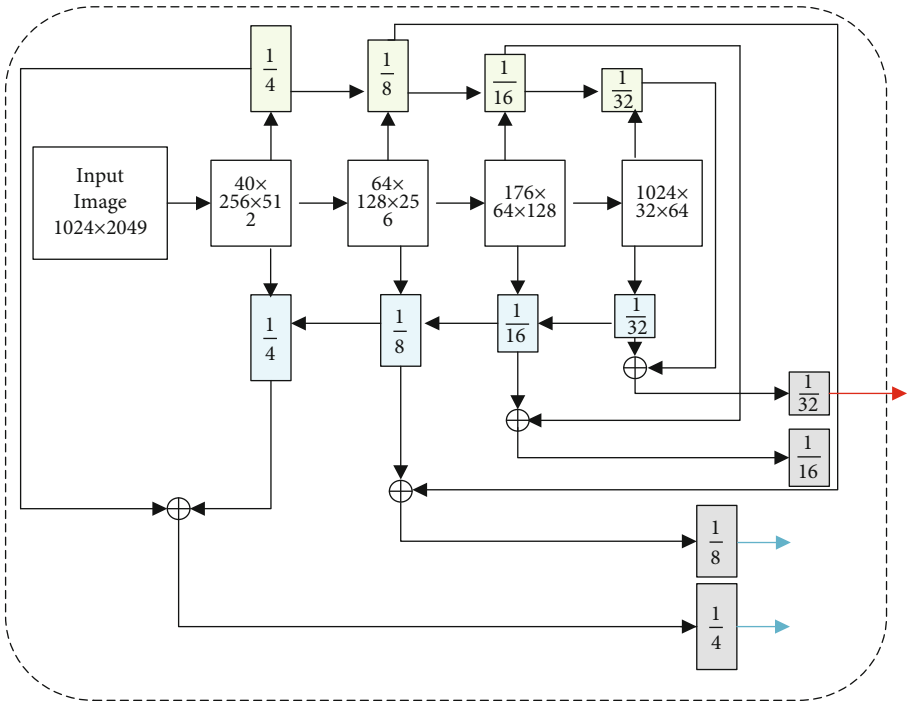


Figure 3: The 2-way pyramid network's information transmission is bidirectional. Four outputs are got from addition by two-way feature blocks.

features. The low-resolution features will be upsampled to the high-resolution features for fast operation and reduce the amount of calculation. The output of the last two branches corresponds to the sum and calculates through 3 × 3 separable convolutions with 256 output channels. After calculating, the results of 4, 8, 16, and 32x downsampling are obtained.

The 2-way pyramid network combined with the backbone network is divided into two parts, the light gray part for forward propagation and the light blue part for back propagation. The backbone network of the white block inputs the information into the two-part pyramid network with 4, 8, 16, and 32x downsampling, respectively, in which the light gray part propagates downward from the features with larger size and the light blue part propagates upward from the features with smaller size. At the same time, the gray and blue blocks are also fused with each other. In this way, the features of the obtained white blocks combine forward propagation and back propagation information. Among them, 32x of the output is used for Atrous Spatial Pyramid Pooling (ASPP), and 8x and 4x of the output will be sent to the decoder part.

3.2. Atrous Spatial Pyramid Pooling. Figure 4 is the architecture of ASPP. Four kinds of atrous convolutions with sampling rates will be input for sampling, which are atrous convolutions with the rate of 1, 6, 12, and 18, respectively. When the interval is 6, 12, and 18, 3 is adopted ×3. If the rate is too large, the context information obtained will be too rare and will not help feature extraction. If the rate is too small, too much feature information obtained will lead to a significant decrease in computing speed. Therefore, the rate of 6, 12, and 18 is the best combination of speed and precision. When the interval is 1, 3 × 3 kernels will become 1 × 1 because there is no rate. This method directly extracts the corresponding features. A total of four atrous convolutions and one pooling form the ASPP model. Finally, the two ASPP structures extract semantic, instance, and multiscale context information, respectively.

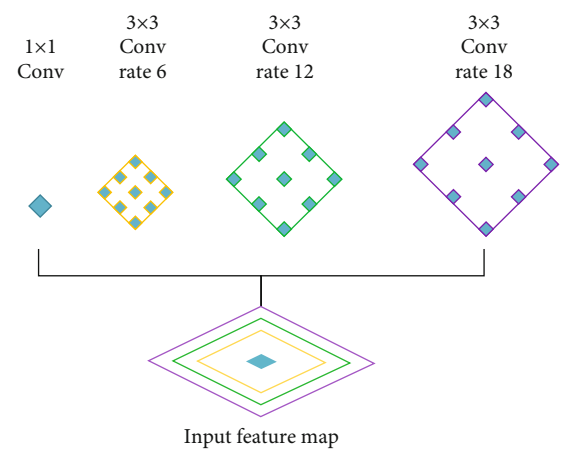


Figure 4: A feature map is divided into four parts. These four parts will be combined into an input feature map which is sent to the attention module.

3.3. Attention Module. Figure 5 shows our attention module. We propose the attention module which combines the channel attention module and spatial attention module in Figure 4. The channel attention module performs the

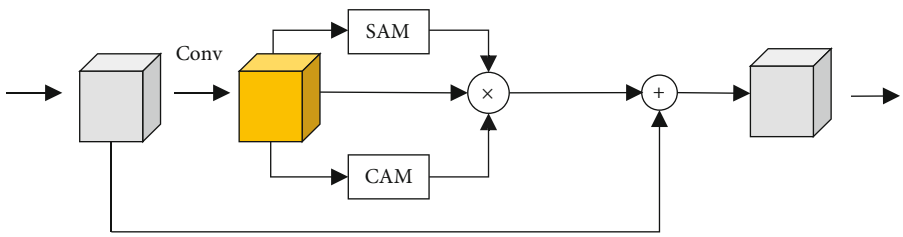


Figure 5: The attention module using spatial attention module (SAM) and channel attention module (CAM) focuses on the spatial and channel part which determines the weight of the feature.

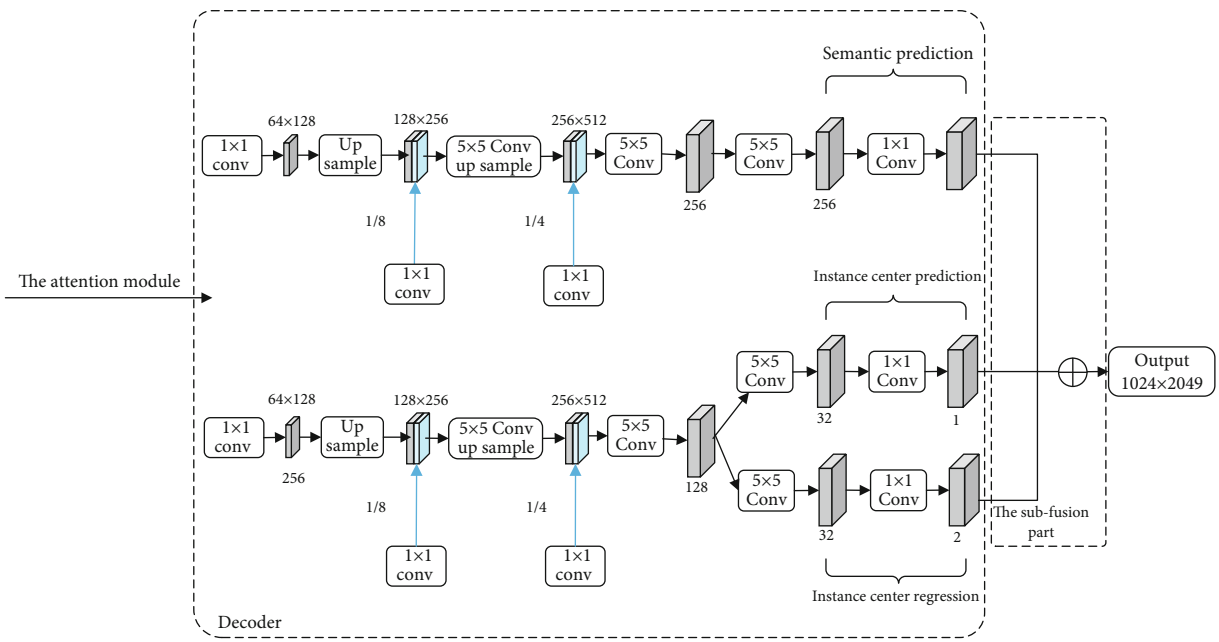


Figure 6: The structure of the image segmentation part. The semantic and instance segmentation branch is the decoder of our model. The subfusion part will obtain the output of semantic segmentation prediction, instance center prediction, and instance center regression.

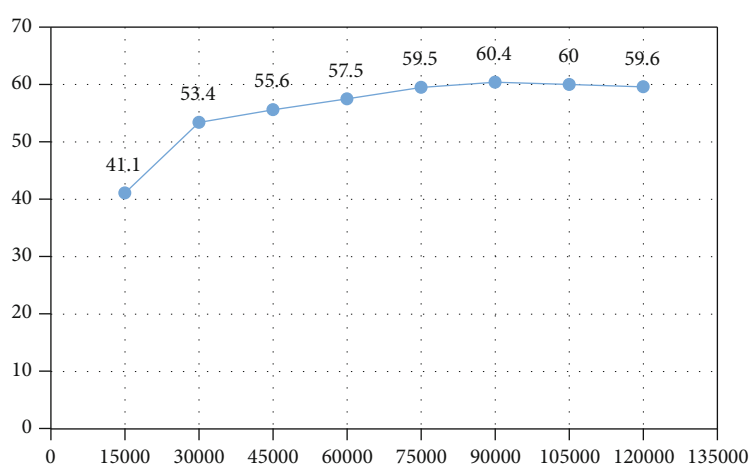


Figure 7: Comparison of panoptic segmentation experiments with different iterations.

Table 1: Comparison of panoptic segmentation of modules.

Model	PQ (%)	SQ (%)	RQ (%)	PQ^{st} (%)	PQ^{th} (%)
Original	57.2	79.9	70.2	63.7	48.2
Model+2-way pyramid network	59.6	81.5	71.7	64.1	51.8
Model+attention	58.7	80.7	71.0	64.0	49.9
Model+2-way pyramid network+attention	60.4	82.4	72.6	64.5	53.2

Table 2: Comparison between our model and mainstream panoptic segmentation networks.

Model	PQ (%)	SQ (%)	RQ (%)	PQ^{st} (%)	PQ^{th} (%)
Deeperlab [21]	56.5	—	—	—	—
Panoptic-FPN [5]	58.1	—	—	62.5	52.0
Panoptic-DeepLab +Res50 [7]	58.0	80.2	70.7	64.3	48.5
Ours	60.4	82.4	72.6	64.5	53.2

maximum pooling and average pooling of the input feature map, respectively, and then puts it into the shared Multilayer Perceptron (MLP):

$$M_C(F) = \text{Sigmoid}(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))). \tag{1}$$

The output features of these two types perform the summation of the element-wise level first, and then, the final channel attention features are obtained through the operation of the sigmoid activation function. The spatial attention module takes the feature map output by the channel attention module as the feature map input by this module:

$$M_S(F) = \text{Sigmoid}\left(f^{7\times7}([\text{AvgPool}(F)\,;\text{MaxPool}(F)])\right). \tag{2}$$

First, do the same operations of maximum pooling and average pooling to obtain two types of outputs and splice them together. Then, the dimension is reduced by convolution, and finally, the spatial attention feature is obtained by the operation of sigmoid activation function:

$$F_1 = M_C(F) \otimes M_S(F) \otimes F. \tag{3}$$

$M_C(F)$ is the output of CAM, while $M_S(F)$ is the output of SAM. Our method uses F_1 to get the output obtained by the cross-multiplication of CAM and SAM.

3.4. Image Segmentation Part. Figure 6 shows the structure of the image segmentation part, and the details of the picture are described below.

Semantic segmentation: this part adopts a method similar to DeepLabv3+, and after each upsampling operation, the 5×5 separable convolution will be used to improve the acquisition of context information. After the 1×1 kernel is finally used, weighted bootstrapped cross-entropy loss is also used in the semantic segmentation branch.

Instance segmentation: the part of the instance segmentation branch is similar to semantic segmentation, while the instance segmentation branch is divided into two parts. One is the instance center prediction, and the other is the instance center regression. F_1 loss is used in the instance center regression to minimize the distance between the predicted heating map and the ground truth heating map. The Mean Square Error (MSE) is the loss of instance segmentation regression.

Fusion: in the subfusion part, there are three kinds of losses, which are from instance center prediction, instance center regression, and semantic segmentation prediction. These three loss functions will be obtained in the form of accumulation:

$$L = \lambda_{\text{ICP}} L_{\text{ICP}} + \lambda_{\text{ICR}} L_{\text{ICR}} + \lambda_S L_S, \tag{4}$$

where λ is the set superparameter, which will be adjusted according to the change of iteration time in training. L is the total loss of our model. ICP represents the instance center prediction, ICR represents the instance center regression, and S represents the semantic segmentation prediction.

4. Results and Discussion

4.1. Experiments. This section introduces the evaluation results of the 2PN model. The first part introduces the applied datasets and corresponding implementation details. The second part discusses the results and their comparison with baseline solutions.

4.2. Datasets and System Settings. Cityscapes: Cityscapes [30] is known as the urban street scene dataset. The images of the dataset are mainly from the street scenes provided by German companies. The dataset consists of 20000 weak annotation frames and 5000 high-quality annotation frames. The Cityscapes dataset has 19 categories, including 2975 pictures to form the training set, 500 pictures to form the Val set, and 1525 pictures to form the test set. The Cityscapes dataset focuses on street scenes with high image quality and fine annotation, which plays an important role in the understanding of street scenes. According to the development direction of panoptic segmentation in the future and in order to reduce the training time, the dataset selected in this experiment is Cityscapes, which focuses on street scenes and has a smaller scale than the Mapillary Vistas dataset [31].

Settings: we choose ResNet-50 [8] as our model's backbone. Our experiments use the same parameter setting as

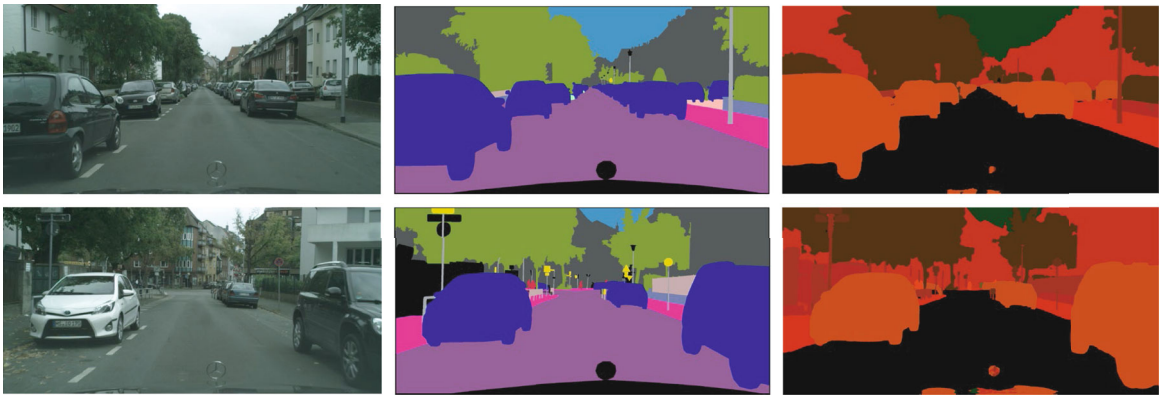


Figure 8: Our visualization results on Cityscapes. We show image, ground truth, prediction from left to right.

Panoptic-DeepLab [7]. Experiments are trained on one NVidia GeForce RTX 3090 with 24 GB video memory.

Our panoptic segmentation model is evaluated by panoptic quality (PQ), which is obtained by multiplying segmentation quality (SQ) and recognition quality (RQ) [1]. PQ^{st} and PQ^{th} represent the panoptic segmentation results of "stuff" and "thing."

5. Results

We first explore the impact of the number of iterations on the accuracy of the network. Figure 7 shows the comparison of panoptic segmentation experiments with different iterations.

As Figure 7 shows, when the number of iterations is 90000, the performance of the network reaches the maximum. Accuracy of the panoptic segmentation network will be reduced due to overfitting.

5.1. Ablation Studies. We analyze the impact of the two-way pyramid network and attention module on the accuracy of panoptic segmentation.

In Table 1, our model gets 3.2% better than the original model in Cityscapes. The model with the 2-way pyramid network works 0.9% better than the model with the attention module.

Due to the advantages of small object feature extraction, the two-way pyramid network and attention module have a great improvement in the "thing" part and less impact on the "stuff" part. The difference is that the two-way pyramid network affects more network layers and has two-way features. Therefore, the two-way pyramid network greatly improves the model. The attention module also improves the panoptic segmentation network, but its improvement range is obviously less than that of the two-way pyramid network, and the main improvement part is also in the "thing" part.

Comparison: compared with the above three models in Table 2, this model has some advantages. First, compared with the baseline, the PQ value in the thing part has obvious advantages. The model in this chapter enhances the feature extraction method and strengthens the extraction of small objects and the acquisition of context information. Meanwhile, compared with Panoptic-FPN, due to the use of empty feature pyramid pooling, it has been significantly improved in semantics and instances.

Finally, Figure 8 shows the results of the proposed model. Our prediction model works well in most areas. It can be observed that things on the roadside are detected, while the things and stuffs can also be correctly distinguished. However, the prediction in the black part of ground truth needs to be improved, where the detailed detection of background is still not clear.

6. Conclusion

This paper proposes a novel framework for panoptic segmentation towards surveillance in an open environment. The proposed network includes a light-weighted two-way pyramid network for better feature extraction, and an attention module is adapted to adjust the importance of extracted features. In this way, our model can obtain the context information features of the image. The attention module also optimizes the space and channel of small-scale features. Finally, the experimental tests show that the methods proposed in this chapter are effective and workable.

Data Availability

The author's e-mail is 630211995@qq.com, and the code of ResNet-50 with 2PN is available.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9404–9413, Long Beach California, 2019.

[2] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125, Honolulu, 2017.

[3] U. Bonde, P. F. Alcantarilla, and S. Leutenegger, "Towards bounding-box free panoptic segmentation," in *DAGM German Conference on Pattern Recognition*, vol. 12544 of Lecture Notes in Computer Science, , pp. 316–330, Springer, Cham, 2021.

[4] H. Wang, R. Luo, M. Maire, and G. Shakhnarovich, "Pixel consensus voting for panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9464–9473, Long Beach California, 2020.

[5] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6399–6408, Long Beach California, 2019.

[6] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[7] B. Cheng, M. D. Collins, Y. Zhu et al., "Panoptic-DeepLab: a simple, strong, and fast baseline for bottom-up panoptic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12475–12485, 2020.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, 2016.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, Boston, 2015.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*, pp. 234–241, Springer, 2015.

[11] H. S. Zhao, J. P. Shi, X. J. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881–2890, Honolulu, 2017.

[12] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semanticimage segmentation," in *Proceedings of the European Conference on ComputerVision (ECCV)*, pp. 801–818, Munich, 2018.

[13] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *European Conference on Computer Vision*, pp. 297–312, Springer, Cham, 2014.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, 2014.

[15] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Boston, 2015.

[16] S. Ren, K. He, R. Girshick, and R.-C. N. N. Faster, *Towards real-time object detection with region proposal networks*, 2015.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *Proceedings of the IEEE International Conference On Computer Vision*, 2017, pp. 2961–2969, Long Beach California, 2017.

[18] A. Fathi, Z. Wojna, V. Rathod et al., "Semantic instance segmentation via deep metric learning," https://arxiv.org/abs/1703.10277.

[19] D. Neven, B. D. Brabandere, M. Proesmans, and L. V. Gool, "Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8837–8845, Long Beach California, 2019.

[20] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid, "Blitznet: a real-time deep network for scene understanding," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4154–4162, Venice, 2017.

[21] T. J. Yang, M. D. Collins, Y. Zhu et al., "Deeperlab," https://arxiv.org/abs/1902.05093.

[22] D. De Geus, P. Meletis, and G. Dubbelman, "Fast panoptic segmentation network," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1742–1749, 2020.

[23] D. De Geus, M. Panagiotis, and G. Dubbelman, "Panoptic segmentation with a joint semantic and instance segmentation network," https://arxiv.org/abs/1809.02110.

[24] J. Li, A. Raventos, A. Bhargava, T. Tagawa, and A. Gaidon, "Learning to fuse things and stuff," https://arxiv.org/abs/1812.01192.

[25] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, Salt Lake City, 2019.

[26] K. Kwon and B. Shin, "3D segmentation for high-resolution image datasets using a commercial editing tool in the IoT environment," *Journal of Information Processing Systems*, vol. 13, no. 5, pp. 1126–1134, 2017.

[27] S. Cheng, Z. Cai, and J. Li, "Curve query processing in wireless sensor networks," *IEEE Transactions on Vehicular Technology.*, vol. 64, no. 11, pp. 5198–5209, 2015.

[28] Z. He, Z. Cai, S. Cheng, and X. Wang, "Approximate aggregation for tracking quantiles and range countings in wireless sensor networks," *Theoretical Computer Science*, vol. 607, pp. 381–390, 2015.

[29] J. Li, S. Cheng, H. Gao, and Z. Cai, "Approximate physical world reconstruction algorithms in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3099–3110, 2014.

[30] M. Cordts, M. Omran, S. Ramos et al., "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, Las Vegas, 2016.

[31] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The Mapillary Vistas Dataset for semantic understanding of street scenes," in *IEEE International Conference on Computer Vision*, pp. 4990–4999, Venice, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3501329, 11 pages
https://doi.org/10.1155/2022/3501329

Research Article

A New Heuristic Computation Offloading Method Based on Cache-Assisted Model

Junhua Wu,[1] **Cang Fan**,[1] **Guangshun Li**,[1] **Zhuqing Xu**,[1] **Zhenyu Jin**,[1] **and Yuanwang Zheng**[2]

[1]*School of Computer Science, Qufu Normal University, Rizhao 276826, China*
[2]*Shandong Huatong Used Car Information Technology Limited Company, Jining 272000, China*

Correspondence should be addressed to Guangshun Li; guangshunli@qfnu.edu.cn

Received 21 January 2022; Revised 24 February 2022; Accepted 1 March 2022; Published 25 March 2022

Academic Editor: Yan Huo

Mobile edge computing (MEC) solves the high latency problem of cloud computing by offloading tasks to edge servers. Due to limited resources, it is necessary to improve the efficiency of computation offloading. However, there is a lot of redundant data transmission between MEC servers and users in the existing methods. Additional data transmission increases the task processing delay. To reduce the total delay, a new cache-assisted computation offloading strategy is proposed. In response to a large number of similar requests from users, a new cache management mechanism is designed. This mechanism can select reusable calculation results more accurately in the cache space through an approximate matching method and improve the cache hit ratio. Then, aiming at the problem of offloading efficiency, the delay optimization problem is transformed into an optimal path problem, a cost function is defined to determine the optimal offloading position, and an improved path planning method is used to plan the optimal offloading path. The simulation results indicate that the proposed scheme can improve the cache hit ratio and reduce the total processing delay of tasks compared with other standard schemes.

1. Introduction

With the development of Internet, increase in the number of network devices generates huge data traffic [1]. It is expected that in the next few years, the load on cloud data centers will be under tremendous pressure. Mobile edge computing (MEC) [2] as a new computing paradigm has been proposed to deal with the problem. MEC servers are deployed at the base stations (BSs) in the MEC system, which can execute the delay sensitive applications in close proximity to the end-users [3]. The edge servers can deploy the computation and storage resources to nearby IoT devices and offer data processing services [4]. Unlike traditional mobile cloud computing (MCC) [5], MEC is able to extend the cloud computing power and the services to the edge of network for two reasons: on the one hand, MEC ensures that data processing relies primarily on local devices rather than cloud servers; on the other hand, it usually does not need to establish a relationship with a remote cloud server, and it can meet the requirements of local users [6, 7]. The edge systems support fine-grained access to different dimensions of data [8]. In addition, mobile devices are faced with the problems such as the energy consumption of battery, the limited resources and the computing capacity in terms of the local processing [9]. Therefore, computation offloading has emerged, which can optimize the transmission delay of the task and reduce the user's computing burden [10]. However, if the users can offload all application tasks to MEC server, the server is likely to be overloaded. While studying the problem of computation offloading, the service cache is also an important topic for MEC [11]. The service cache prestores the database or library related to the application and allows the corresponding task to be offloaded. Due to the limited resources in edge servers, the caching decisions must be made carefully to maximize system performance [12]. However, how to use cache to reduce the service delay while maximize storage utilization is still a key issue in the edge network. But the heterogeneity of edge network and the

uneven distribution of users make it difficult for the system to balance cache and offloading. To deal with this problem, this paper proposes a cache-assisted offloading method and a target server matching strategy based on the cost of task to determine the appropriate target server to meet user's needs.

In brief, the main contributions of this paper are threefold:

(1) This paper considers the MEC server cooperative cache system. To reduce the access delay and the computation overhead, we design a new cache management strategy based on dynamic data approximate matching. Through an approximate matching algorithm based on the sample distance, a data set similar to the input data is selected from the collaborative cache space. By obtaining the corresponding calculation result for reuse, the cache hit ratio can be improved

(2) In order to improve the offloading efficiency, this paper proposes a new computation offloading method. According to the time sensitivity and communication cost, the optimal target server can be estimated; then the computation offloading problem can be transformed into an optimal path planning problem; finally, the optimal offloading path can be planned

(3) Evaluating the effectiveness of HCAM through specific simulation experiments

The rest of the paper is organized as follows. Section 2 summarizes the most related work. Section 3 introduces the system model in detail. In Section 4, we describe the cache-assisted offloading strategy. Section 5 introduces the efficiency evaluation. The simulation results are reported in Section 6. Finally, the conclusion and the future work are discussed in Section 7.

2. Related Works

In recent years, some augmented reality tasks have higher requirements for real-time performance when processing data. So, the traffic of mobile data continues to grow. It is not difficult to find that data requested by users is highly repetitive, which will lead to a large amount of redundant data transmission. In recent years, the caching problem has attracted the attention of researchers as a method to solve the delay problem [13, 14]. Cache is a new strategy to improve the performance and the service quality of mobile edge networks. It includes offloading tasks to the mobile edge cloud and storing computation results in the local storage located at the edge of network. This technology avoids redundant and repetitive processing of the same task, thereby simplifying the offloading process and improving the utilization of network resources [15, 16]. As a new method to alleviate the unprecedented network traffic, mobile edge caching has been widely used in the wired internet, and it has proved that it can reduce delay and energy consumption [17]. To date, a lot of research works have focused on optimizing caching methods to solve the delay and energy consumption problems in computation offloading. In [18], the author considered the horizontal cooperation between mobile edge nodes for joint caching and proposes a new transformation method to solve the problem of edge caching and improve cache hit rate of the network. In [19], authors designed a heterogeneous collaborative edge cache framework by jointly optimizing node selection and cache replacement in mobile networks. The joint optimization problem is expressed as a Markov Decision Process (MDP), and Deep Q Network (DQN) is used to solve the problem, which alleviates the offloading traffic load. In [20], the problem of edge cache optimization in fog radio access networks (F-RANs) was studied, and a distributed edge cache scheme was proposed, which reduced the delay of service and the traffic load. In [21], authors combined user's context behavior to optimize the cache and modeled the problem of maximizing the click-through rate of the content as a knapsack problem. In the MEC paradigm, a heuristic intelligent caching algorithm was proposed, which had the better cache hit rate and the stability and the lower overhead. In [22], authors studied the problem of vehicle edge caching in the actual vehicle scenes. In order to obtain the higher hit ratio, the service process was modeled as a joint process of vehicle movement and parking through the approximation theory, and a method based on the practical vehicle edge cache solution realizes the trade-off between hit ratio and interrupt request ratio. In [23], the computation offloading method of cached data was studied, and a new cache-aware computation offloading strategy was proposed. The goal was to minimize the equivalent weighted response time of all tasks with the constraint of computational power and caching capacity. In [24], the authors designed the underlying structure of cache causality and task's dependency model and designed an alternate minimization technique to reduce the complexity to alternately update the cache placement and the offloading decisions. In [25], the authors considered a complexed scenario, in which multiple moving MDs are sharing multiple heterogeneous MEC servers, and a problem named as minimum energy consumption problem in deadline-aware MEC system is formulated.

Some research works have concentrated on introducing the concept of edge caching in different systems, proposing the new frameworks or models to solve the optimization problem during offloading. In [26], a cooperative offloading and buffering model was designed, an optimization problem containing two independent problems was constructed, and a resource management algorithm was developed to guide a BS to jointly schedule the calculation of offloading and allocate the data buffers. The total delay of system communication can be minimized through the optimal offloading and caching decisions. In [27], authors proposed a collaborative edge caching scheme, defined the joint optimization problem as a Dual-Time-Scale Markov Decision Process (DTS-MDP), and proposed a framework based on Deep Deterministic Policy Gradient (DDPG). In [28], in view of the high link load of edge cache and the small storage space

of the server, a cloud-edge collaborative cache model based on the greedy algorithm was proposed. In [29], the problem of edge caching in the optical fiber computing networks was analyzed, and a capacity-aware edge caching framework was proposed. The problem of average download time minimization is described as a multiclass processor queuing process, and an algorithm based on the Alternating Direction Multiplier Method (ADMM) was proposed. In [30], a new intelligent edge is defined, which combines a heterogeneous IoT architecture with edge computing, caching, and communication. In [31], an offloading framework that enables task caching was proposed in edge computing to jointly optimize the response delay and the energy consumption of roadside units. In [32], in order to minimize the total delay consumption of tasks, the authors jointly considered computation offloading, content caching, and resource allocation as an integrated model, designed an asymmetric search tree, and improved the branch and bound method to obtain a set of accurate decision-making and resource allocation strategies. By summarizing the research of computation offloading method with cached data in MEC, we can conclude that the combination of edge caching and computation offloading has made progress in meeting user's requirements and improving user's experience.

In summary, most of the existing works do not take into account the influence of cache management and also not have full investigation of the collaboration of MEC servers. Thus, when the MEC environment changes dramatically, the burst request volume can bring sudden increased computation load to MEC servers, and the edge network links in certain regions will also become congested, leading to a significant impact on the efficiency of computation offloading. Accordingly, we take full use of the characteristics of edge cache to propose a computation offloading method based on cache-assisted, which can improve the cache hit ratio and the offloading efficiency.

3. Network Model

3.1. System Architecture. Computation offloading is a proven and successful example that can be used to enable resource-intensive applications on mobile devices. Efficient data sharing extends the collaboration capabilities of edge system [33]. For emerging mobile collaboration applications, when multiple users are at the same distance, offloaded tasks can be copied. Researchers urgently need to design a collaborative offloading scheme and cache popular calculation results that may be reused by other mobile users. In multi-acess mobile edge computing, tasks offloaded from the users are usually associated with the specific services, and these services need to be cached in MEC nodes to perform tasks. Deciding which services to cache and which tasks to perform on each MEC node with limited resources is critical to maximizing the efficiency of offloading [34]. In this section, considering an optimized regional collaborative cache system architecture. Table 1 presents the key notations of optimization model and corresponding descriptions.

As shown in Figure 1, considering a distributed multi-user MEC system consisting of multiple MEC servers connected via backhaul links, each of which can provide computation and storage power to meet the delay-sensitive requirements of tasks. This article assumes that only one task is generated per user. In this system, let $N = \{1, 2, \cdots, n\}$ denotes all users, and let $R_i = \{s_i, c_i, t_i\}$ denotes a random generating task i. s_i represents the size of tasks; c_i denotes the amount of computation resource needed to execute the application task, quantified in CPU cycles; t_i represents the time required to perform the task. In this system model, each BS is equipped with a MEC server to handle offloading requests. According to their own needs, the users can choose to perform tasks locally or offloaded to the edge servers. Assuming that each task occupies only one virtual machine, the user-generated request determines whether the virtual machine is occupied or not based on the offloading decisions. In this paper, the optimized cache management model and the offloading strategy are designed to reduce the user's request delay, which are introduced in the following sections.

Table 1: Symbol definition list.

Notions	Description
N	Number of users
M	Number of servers
s_i	Size of the task i
t_i	Processing time of task i
c_i	Computing resource required by task i
l_i	Computation offloading decision i
f_0	Computing capacity of the MEC server
T_i	The processing time of task i
T_n	The total processing time

3.2. Problem Formulation. Latency is an efficiency manifestation of system executing user's requests and a direct evaluation criterion of user experience. In this paper, the delay is composed of four parts: the communication time between MEC servers and users MT_{l_i}; the calculating time for servers to execute tasks CT_{l_i}; the waiting time for other tasks WT_{l_i}; and the time for BS forwarding to target MEC server BT_{l_i}.

Defining the offloading decision variables $L = \{l_1, l_2 \cdots, l_n\}$, a binary variable is used to represent the task executing locally or offloaded to the edge server:

$$l_i = \begin{cases} 1, & \text{the server caching resources,} \\ 0, & \text{other.} \end{cases} \tag{1}$$

Assuming that the channel adopts microwave link and the communication mode is full duplex, the calculation formula of the communication rate τ_i between the MEC server and the user is as follows:

$$\tau_i = w \log_2 \left(1 + \frac{P_i |h_i|}{I_i + n_0} \right), \tag{2}$$

where w is the channel bandwidth, P_i is the data rate sent by

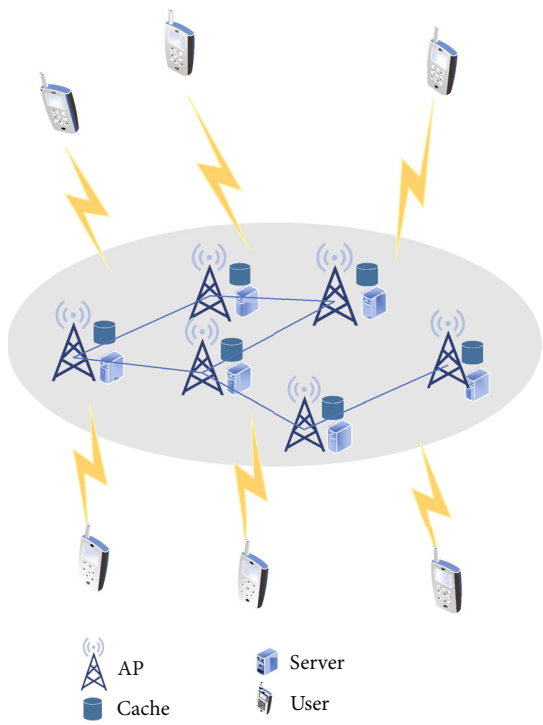


Figure 1: Cache-assisted MEC offloading architecture.

the user, h_i is the channel gain, I_i is the channel interference, and n_0 is Gaussian white noise. Therefore, the calculation formula of MT_{l_i} is:

$$MT_{l_i} = \frac{s_i}{\tau_i}. \tag{3}$$

The calculation formula of CT_{l_i}:

$$CT_{l_i} = \frac{c_i}{f_0}. \tag{4}$$

Because the MEC server accepts user requests on a first-come, first-served basis, the formula for WT_{l_i} is as follows:

$$WT_{l_i} = \sum_{i=1}^{k} \left(MT_{l_i} + CT_{l_i}\right). \tag{5}$$

k is the number of transmitted tasks before the task R_i. When $MT_{l_i} \geq t_i$, the server chooses to forward task i. The calculation formula of BT_{l_i} is as follows:

$$BT_{l_i} = \frac{s_i}{\varphi_i}. \tag{6}$$

φ_i is the forwarding transmission rate of BS. To sum up, the sum delay of task T_i is defined as the sum of MT_{l_i}, CT_{l_i}, WT_{l_i}, and BT_{l_i}. The calculation formula of T_i is as follows:

$$T_i = MT_{l_i} + CT_{l_i} + WT_{l_i} + BT_{l_i}. \tag{7}$$

The calculation formula of T_n is as follows:

$$T_n = \sum_{i=1}^{N} T_i. \tag{8}$$

This problem is equivalent to assigning task to the resource node in different region, minimizing the total processing time of all tasks. When the limited cache capacity of edge server is relaxed, it can be transformed into a classic transmission problem [35]. The optimization problem is as follows:

$$\min\left(T_n\right). \tag{9}$$

4. Optimal Offloading Solution

4.1. Content Update Method. In the MEC distributed architecture, a scenario of dynamic probabilistic cache is designed according to the time-varying content, and it can adapt to the time-varying content popularity without knowing the popularity. In the environment of the server collaboration, due to the different needs of users and the different requests cluster into the area of radius r, different regions are connected by optical fiber, and the collaboration area of MEC server can realize sharing of content. In the case of limited cache capacity, the MEC server must take the initiative to cache. The BS can parse part of the content request and place the cached content without returning the obtained result through the backhaul link, which relieves the pressure on the communication link. However, the popularity of the content changes with time, and the dynamic probability cache can adapt to the time-varying instantaneous content popularity and improve the cache hit rate of instantaneous content. In this article, the probability p_i of user is randomly requesting task i, $i \in (1, n)$ obeys Zipf's law, and therefore, p_i is calculated as follows:

$$p_i = \frac{1/i^{\eta}}{\sum_{i=1}^{n} 1/i^{\eta}}. \tag{10}$$

η is the value of the Zipf's distribution exponent.

4.2. Optimal Cache Management Strategy. In order to further improve the cache hit ratio, this paper adopts an optimized cache management strategy on the basis of the above content update method. Assuming that the area near each BS is divided according to empirical values, it is ensured that the number of edge servers in each area is approximately the same. Collaboration between servers can integrate cache at the edge of network. In a collaborative environment, the requested content can be transferred from one MEC server to another MEC server. As the computation capacity of edge servers are limited, repeated calculation of the same request will consume computing resources and increase the waiting

delay of end users [36]. The above process will face two challenges: on the one hand, since there are almost no two identical images and voices in the scene of image recognition and speech recognition, only the most similar data can be found instead of the completely identical data, so the traditional cache selection strategy based on the accurate matching is no longer applicable [37]; on the other hand, users generate a large amount of data every day, and it takes a lot of time to search for the same or similar data in the massive data, and the search becomes more difficult due to the increase of data dimensions. To address such problems, we propose a new cache management strategy based on the dynamic data approximate matching as given below.

Among these spatial index data structure construction methods, Baton-tree [38] is the most effective, and the complexity of other methods is affected by the dimension of data. When doing an approximate data look up with Baton-tree, it can get a similar data set of the input data, and then, the general approaches are to go through the similar data set and find the closest data set to input data and return the result but the search accuracy of that method is low. In order to improve the search accuracy, KNN [39] algorithm can be used to filter the data in the similar data set.

4.2.1. Matching Method Based on the Distance Threshold of Cache Data. The existing KNN search algorithm generally ignores the influence of distance on the accuracy of the algorithm and believes that approximate data has the same distance weight [40]. In fact, the distance between data in the set and the input data determines the similarity between the data and the input data. In this paper, a matching algorithm based on distance is proposed to search the data in the similar data set acquired by Baton-tree algorithm more accurately, so as to effectively improve the accuracy of data selection.

When defining the weight of each data, the matching method based on distance threshold takes into account the Euclidean distance between each similar data and the input data. Specifically, the farther the approximate data is from the Euclidean distance of the input data, the smaller the weight. Defining the Euclidean distance as $\mathrm{dis}(\mathrm{data}_0, \overline{\mathrm{data}_i})$, the formula is as follows:

$$\mathrm{dis}\left(\mathrm{data}_0, \overline{\mathrm{data}_i}\right) = \sqrt{(x_0 - \overline{x_i}) + (y_0 - \overline{y_i})}, \tag{11}$$

where data_0 denotes the input data R_0 and $\overline{\mathrm{data}_i}$ denotes approximate data of the input data R_0. Let (x_0, y_0) represents the coordinate of data_0, and let $(\overline{x_i}, \overline{y_i})$ represents the coordinate of $\overline{\mathrm{data}_i}$. Given the input data data_0 and the approximate data set $\mathrm{dataset}_j$, where $\forall \overline{\mathrm{data}_i} \in \mathrm{dataset}_j$, $\forall i \in j$. θ_i is used to indicate the weight value of the approximate data $\overline{\mathrm{data}_i}$, it can be calculated using the following:

$$\theta_i = \frac{j}{\left(\mathrm{data}_0, \overline{\mathrm{data}_i}\right)}. \tag{12}$$

θ_0 is the weight threshold.

In this paper, the discriminant function between data_0 and $\overline{\mathrm{data}_i}$ can be expressed as $\boldsymbol{p}_i$; it can be calculated using the following:

$$\boldsymbol{p}_i = \sum_{i=1}^{j} \theta_i = \sum_{i=1}^{j} \frac{j}{\mathrm{dis}\left(\mathrm{data}_0, \overline{\mathrm{data}_i}\right)}. \tag{13}$$

Let $|\boldsymbol{p}_i|$ denotes the coordinate axis vector modulo $\boldsymbol{p}_i$, vector $\boldsymbol{P}$ can be expressed as $\boldsymbol{P} = \sum_{i=1}^{j} \boldsymbol{p}_i$. Therefore, let λ_i defines the similarity between input data and data in the cache space; it can be indicated with cosine between $\boldsymbol{p}_i$ and $\boldsymbol{P}$; the formula is calculated as follows:

$$\lambda_i = \frac{\langle \boldsymbol{p}_i, \boldsymbol{P} \rangle}{|\boldsymbol{p}_i||\boldsymbol{P}|}. \tag{14}$$

λ_0 is the similarity threshold. For input data, data_i is similar to the data set obtained by the Baton-tree algorithm with different distances:

$$\mathrm{dataset}_j = \left\{(\mathrm{data}_1, \lambda_1, \theta_1), (\mathrm{data}_2, \lambda_2, \theta_2), \cdots, \left(\mathrm{data}_j, \lambda_j, \theta_j\right)\right\}. \tag{15}$$

The paper takes the maximum value $\lambda_{\max}$ among the j cosine values in $\mathrm{dataset}_j$, and the corresponding data value is denoted as $\mathrm{data}_{\max}$. If $\lambda_{\max} > \lambda_0$ and $\theta_i < \theta_0$, return θ_i and $\lambda_{\max}$ corresponding data_i as the approximate match of the input data data_0; otherwise, return Null, the query fails. As shown in Figure 2, it describes a cache management mechanism based on approximate matching.

4.3. Problem Transformation. According to the Dijkstra theoretical method [41], the problem of finding appropriate edge cache nodes can be transformed into the problem of shortest path planning. This paper assumes that the transmission rate between connected MEC servers, denoted as v , are all equal. Let $v_{m_2}^{m_1}$ be the transmission rate of the shortest route between the m_1 and m_2. If the server m_2 is connected with another sever m_3, there is a relationship among v, $v_{m_3}^{m_1}$, and $v_{m_2}^{m_1}$:

$$\frac{1}{v_{m_3}^{m_1}} = \frac{1}{v_{m_2}^{m_1}} + \frac{1}{v}. \tag{16}$$

It can be deduced from (16):

$$v_{m_3}^{m_1} = \frac{v_{m_2}^{m_1} \cdot v}{v_{m_2}^{m_1} + v}. \tag{17}$$

According to (16) and (17), it is obvious that the value of $v_{m_3}^{m_1}$ and $v_{m_2}^{m_1}$ proved that the shortest path means the least number of channels.

4.4. Offloading Location Confirmation. Some researchers have proposed different cost estimation methods of task execution. The most common methods are based on task time

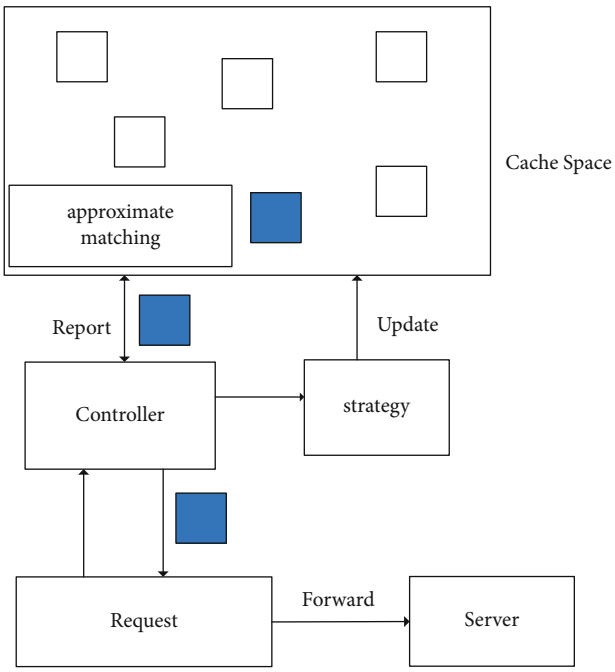


Figure 2: Cache-assisted MEC offloading mechanism.

sensitivity [42]. However, they did not consider the computing resource usage of the MEC server. Therefore, this paper proposes a new method for estimating the cost of task execution; the formula is as follows:

$$C_m = \frac{1}{Q}\sum_{i=1}^{m}(x_m - X). \tag{18}$$

It can obtain the congestion degree of the communication link of servers through (18), where Q is the number of divided areas, x_m is the number of nodes in the area m, and X is the average of the number of nodes in all areas. Combined with formula (8), the execution cost function of task is:

$$V_{i,m} = \alpha T_i + \beta C_m. \tag{19}$$

α and β are the weights given to the two objectives, respectively, with $\alpha + \beta = 1$. After the estimated execution time of the edge server and the congestion level of the communication link are returned to the terminal, the terminal device decides whether to perform a computation offloading and the computation offloading location. This article uses the following steps to confirm offloading position. It is described in Algorithm 1.

Avoiding resource conflicts: after the MEC server executes the search algorithm, it will find a server that meets the user's needs. Before assigning to users, because each user has different task requests and server processing time is also different, it will lead to some resource conflicts: when the server is idle, multiple users compete for cache and computing resources at the same time. This article considers that the server receives the user's resource request and sets the dynamic priority according to the execution time and order of the user request. Among them, the dynamic priority refers to obtaining the initial priority when the user applies for resources. The users constantly modify the priority level when using resources. In this way, conflicts in resource usage are avoided.

4.5. Offloading Path Identification. In respective areas, there are complex routes between edge cache nodes, and the least costly path needs to be found on the premise of determining the appropriate target server. This paper designs a path planning algorithm based on the cost of task. By using problem information to guide the search, the cost of the system search is reduced and the throughput is improved. It is shown in Algorithm 2.

5. Efficiency Evaluation

Figure 3 describes the performance comparison between the impoved distance search algorithm (IDSA) proposed in this paper and the distance search algorithm (DSA). By increasing the number of edge nodes, the total delay changes of two algorithms are compared. It can be seen that when the number of edge node is less than 4, the performance of two algorithms is close. Due to increase in the number of requested users, waiting, transmission, and calculation delay will all increase, resulting in the different degrees of increase in the delay of the two algorithms. However, it can be seen from the figure that when the number of servers is greater than 6, the performance gap between IDSA and DSA gradually increases, mainly because the algorithm proposed in this paper can quickly plan the offloading path and reduce the total delay of users. Therefore, the algorithm proposed in this paper is significantly better than DSA in terms of delay performance.

6. Simulation Experiment

In this section, we will evaluate the performance of the proposed scheme through simulation. The cache scheme is compared with the following four schemes: (1) random cache (RC) [43]: randomly caches popular content; (2) greedy cache (GC) [44]: only cache popular content in this area; (3) fair cache (FC) [45]: each collaboration area proportionally caches popular content; (4) collaborative edge cache offloading (CECO) [46]: only collaborative caching between edge servers; (5) heuristic cache-assisted method (HCAM): cache-assisted offloading method based on approximate matching proposed in this paper.

Experimental simulation parameters are shown in Table 2.

Figure 4 shows the relationship between the number of tasks and the total delay of under the same task processing method and different cache schemes. When the number of tasks is between 100 and 200, the system can process user requests in time, and the performance of each scheme is very close. When the number of tasks is greater than 200, the local and edge nodes cannot process all tasks within the time required by the user, and task access and waiting delays

Input: T_n; C_m; $A = \varnothing$
Output: Optimal offloading location E_m
for $i = 1$ to N **do**
  for $m = 1$ to $M - 1$ **do**
    Calculate the cost $V_{i,m}$ of the server m matching task i by formula (1)–(9);
    $V_{i,m} \longrightarrow A$;
  Select minimum $V_{i,m}$ corresponding m;
  return E_m

Algorithm 1: OLC Algorithm

Input: Request server E_o; cost function $G_1(m)$, $G_2(m)$; actual cost from starting point to candidate point $g_1(m)$, $g_2(m)$; estimated cost from candidate point to target point $h_1(m)$, $h_2(m)$; search step *step*
Output: Optimal path set p
$p = \varnothing$;
Obtain E_m through Algorithm 1, $E_o \longrightarrow p$, $E_m \longrightarrow p$;
If set p is empty **then**
  Return false.
Else
  While m is searched forward and backward and marked as Min and $g_1(\text{Min}) + g_2(\text{Min})$ is smallest **do**
  If(E_0, m) is null **then**
    Search for node $m + 1$;
  Else
    Calculate the cost of successor node m, $G_1(m) = g_1(m) + g_2(m)$;
    Select minimum $G_1(m)$, $E_0 \longleftarrow m$, $E_m \longrightarrow p$, Min $= m$, $step = step + 1$;
  If$(\text{Min}, m + 1)$ is null **then**
    Search for the node $m + 2$;
  Else
    Calculate the cost of successor node $m + 1$, $G_1(m + 1) = g_1(m + 1) + g_2(m + 1)$;
    Select minimum $G_1(m + 1)$, Min $\longleftarrow m + 1$, $E_{m+1} \longrightarrow p$, Min $= m + 1$, $step = step + 1$;
  If$(\text{Min}, m + 2)$ is null **then**
    Search for the node $m + 2$;
  Else
    Calculate the cost of successor node $m + 2$, $G_1(m + 2) = g_1(m + 2) + g_2(m + 2)$;
    Select minimum $G_1(m + 2)$, Min $\longleftarrow m + 2$, $E_{m+2} \longrightarrow p$, Min $= m + 2$, $step = step + 1$;
    Search backward from node E_s, search step is *step*;
    Return 7;
Return p;

Algorithm 2: OPP Algorithm.

increase, which in turn causes the total delay to increase with the increase of tasks. Compared with the greedy cache scheme, fair cache scheme, and random cache scheme, HCAM has a gap in the total delay of tasks as the number of tasks increases. When the number of tasks is 400, the gap is maximum. The task delay of HCAM scheme is 0.053 s, and the task delay of GC is 0.27 s. Although the performance of HCAM and CECO is relatively close, as the tasks increase, CECO has always been above HCAM. It can be seen from the figure that HCAM finally controls the task delay below 0.1 s; its performance is better than the other four schemes. This is mainly because the scheme adopts the principle of approximate matching to improve the cache hit rate when processing user requests at the edge and can reduce the transmission of the backhaul link, thereby reducing the user's waiting delay.

In Figure 5, when the number of user tasks is small, the four methods all show better optimization effect. When the number of tasks is about 100, because the user's request can be processed locally in time, it reduces task transmission and calculation time, so the local method performs better than offloading, CECO, and HCAM. It can be seen that when the number of tasks is between 100 and 200, the performance of HCAM and CECO is close to local, and three methods are better than the offloading. However, due to limited resources, as the number of user tasks increases, the total delay of the four schemes is increasing. When the number of tasks is greater than 200, the delay performance of four methods begins to show a gap. Finally, when the number of tasks reaches 500, HCAM is significantly better than the other three methods, and the performance gap is maximized. It can be seen that when there are many computing

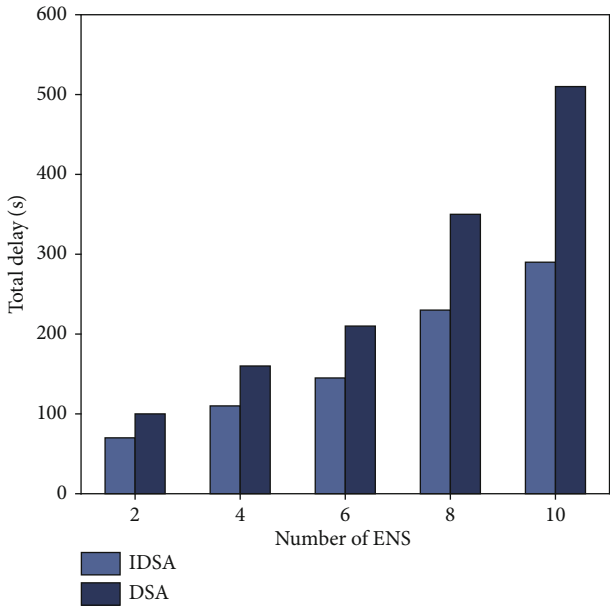


Figure 3: Delay comparison.

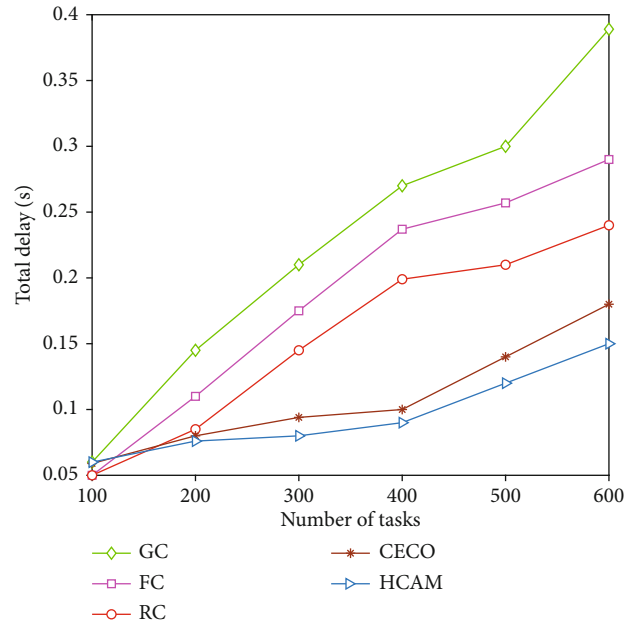


Figure 4: Delay comparison of different cache schemes.

Table 2: Simulation parameters.

Parameters	Value
Number of users	100
Number of edge nodes	10
Number of BS	1
Computing capacity of users	1-2 GHz
Computing capacity of MEC servers	6 GHz
The input data size of tasks	500-800 KB
Background noise power	-100 dBm
Channel bandwidth	1.5 MHZ
Cache size	150 GBs

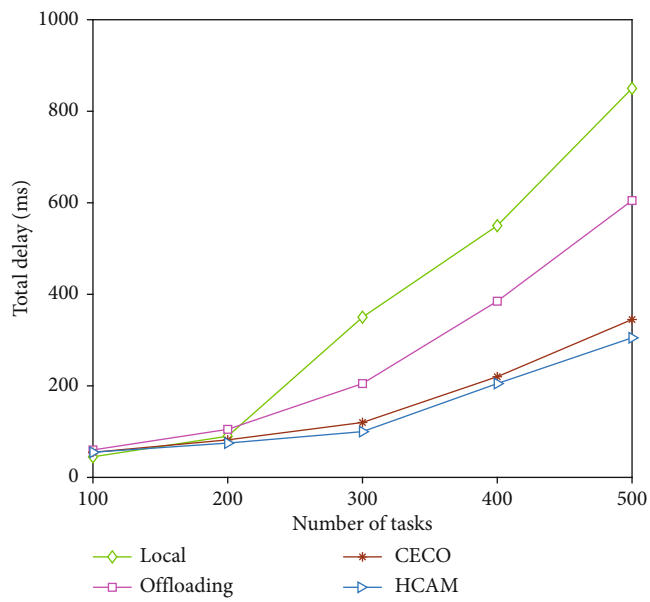


Figure 5: Delay comparison of different task processing schemes.

tasks, both HCAM and CECO use cache resources to reduce the transmission delay of the task; the performance is better than local and offloading. But when the cache is not hit, HCAM reduces the total delay by approximately 11.5% by forwarding tasks to the appropriate MEC server for processing in time which is compared with CECO.

Compared with GC, RC, FC, and CECO to verify the effectiveness of the proposed method, Figure 6 describes the comparison of the four methods on the cache hit ratio, and the cache hit rate is used as one of the performance criteria for evaluating the method proposed in this article. In the case of limited cache space, the higher the cache hit ratio, the lower the overall task processing delay. It can be seen from the figure that when the task time is relatively small, as the cache space increases, these four methods can all increase the cache hit ratio. As the number of tasks increases, the performance of the method proposed in this paper is better than other methods. The main reason is that HCAM optimizes the management and allocation of cache space through a new cache management mechanism. Compared with CECO, it adopts the cache approximate matching principle on the basis of edge collaborative cache, which improves the cache hit ratio.

Figure 7 shows the impact of cache size on the average system delay variation. Since five schemes adopt different cache management and allocation strategies, it can be seen from the figure that as the cache increases, the performance of the five methods differs in performance. As cache space increases, the cache of hot content will also increase, which increases the cache hit ratio. When the user generates a

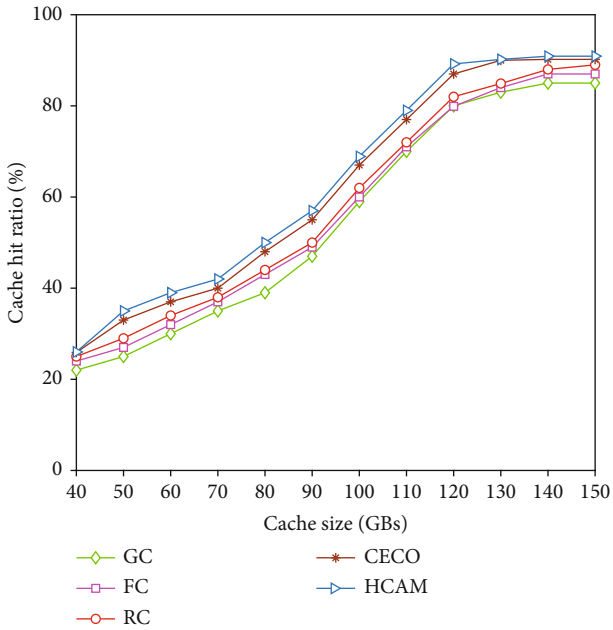


Figure 6: Comparison of cache hit ratio.

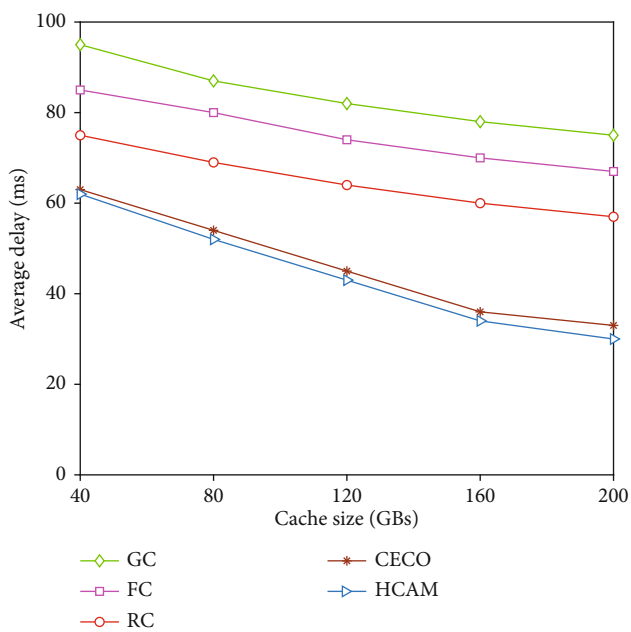


Figure 7: Comparison of average delay.

request, the edge server can directly send the cached content to the user. Users no longer need to wait for tasks to be offloaded to the target server, reducing transmission and calculation delays. When this article adopts an optimized cache management mechanism and cooperative cache model, even if the content requested by the user is not cached, it can be processed at the edge system as far as possible. Compared with the four methods, the average processing delay of the system is reduced to a certain extent. When the cache size is set to 40 GBs, the average delay of HCAM is the smallest. As the cache space increases, the GC method only satisfies the requests of a few users, and the system latency exhibits the greatest. Although FC has a higher cache hit ratio and better performance than GC, the average delay in the system is very close. The RC method further improves the cache hit ratio, which is better than the FC and GC methods. Although both HCAM and CECO use cooperative caching to reduce the average delay performance close to each other, HCAM uses the principle of approximate matching to increase the cache hit ratio, thereby reducing user access latency. However, when the cache space is 200 GBs, the performance of the HCAM method is optimal, which is about 1%, 24%, 36%, and 42% higher than the performance of CECO, RC, FC, and GC.

7. Conclusion and Future Work

In this paper, we focus on a computation offloading strategy. To reduce the processing delay, this paper design a new cache management strategy based on dynamic data approximate matching. Then, a new cache-assisted offloading mechanism for edge server is proposed. To improve the efficiency of offloading, this paper transforms the problem of offloading location confirmation into an optimal path planning problem, a heuristic algorithm based on task cost has been introduced to confirm the optimal server. The simulation results show that our scheme can reduce the total delay compared to GC, FC, RC, and CECO.

In the future, we will optimize the cache strategy. Besides, we will further study the computation offloading method under the job-related situation. In addition, we will explore algorithms suitable for task priority.

Data Availability

The (data type) data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China grants 61771289 and 61832012, Major Basic Research of Natural Science Foundation of Shandong Province with grants ZR2019ZD10; Key Research and Development Program of Shandong Province with grants 2019GGX101050.

References

[1] G. Zhang, S. Zhang, W. Zhang, Z. Shen, and L. Wang, "Joint service caching, computation offloading and resource allocation in Mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 99, pp. 1–1, 2021.

[2] L. Guangshun, S. Jianrong, W. Junhua, and W. Jiping, "Method of resource estimation based on QoS in edge

computing," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–9, 2018.

[3] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-based big data storage systems in cloud computing: perspectives and challenges," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 75–87, 2017.

[4] Z. Cai and T. Shi, "Distributed query processing in the edge assisted IoT data monitoring system," *IEEE Internet of Things Journal*, vol. 99, pp. 1–1, 2020.

[5] A. Hekmati, P. Teymoori, T. D. Todd, D. Zhao, and G. Karakostas, "Optimal mobile computation offloading with hard deadline constraints," *IEEE Transactions on Mobile Computing*, vol. 99, pp. 1–1, 2019.

[6] T. Zhao, S. Zhou, L. Song, Z. Jiang, X. Guo, and Z. Niu, "Energy-optimal and delay-bounded computation offloading in mobile edge computing with heterogeneous clouds," *China Communications*, vol. 17, no. 5, pp. 191–210, 2020.

[7] S. K. Datta and C. B. Onnet, "An edge computing architecture integrating virtual IoT devices," *Consumer Electronics*, pp. 1–3, 2017.

[8] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science & Engineering*, pp. 1–1, 2018.

[9] P. Corcoran and S. K. Datta, "Mobile-edge computing and the internet of things for consumers: extending cloud computing and services to the edge of the network," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 73-74, 2016.

[10] F. Zhou and R. Q. Hu, "Computation efficiency maximization in wireless-powered mobile edge computing networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3170–3184, 2020.

[11] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, 2017.

[12] H. Feng, S. Guo, L. Yang, and Y. Yang, "Collaborative data caching and computation offloading for multi-service mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 99, pp. 1–1, 2021.

[13] G. Li, J. Wang, J. Wu, and J. Song, "Data processing delay optimization in mobile edge computing," *Wireless Communications and Mobile Computing*, vol. 2018, no. 1, pp. 1–9, 2018.

[14] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[15] Z. Cai and Z. He, "Trading private range counting over big IoT data," *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019.

[16] N. D. Pietro and E. C. Strinati, "Proactive computation caching policies for 5G-and-beyond mobile edge cloud networks," *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018.

[17] Y. Liu, D. Zheng, X. Xia, and B. Zhang, "Data caching optimization in the edge computing environment," *Environment*, 2020.

[18] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "A novel mobile edge network architecture with joint caching-delivering and horizontal cooperation," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 19–31, 2018.

[19] X. Wang, R. Li, C. Wang, X. Li, and V. C. M. Leung, "Attention-weighted federated deep reinforcement learning for device-to-device assisted heterogeneous collaborative edge caching," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 154–169, 2021.

[20] Y. Jiang, Y. Hu, M. Bennis, F. C. Zheng, and X. You, "A mean field game-based distributed edge caching in fog radio access networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1567–1580, 2020.

[21] Y. Zeng, J. Xie, H. Jiang, G. Huang, and J. Li, "Smart caching based on user behavior for mobile edge computing," *Information Sciences*, vol. 503, pp. 444–468, 2019.

[22] Y. Zhang, C. Li, T. H. Luan, C. Yuen, and W. Wu, "Towards hit-interruption tradeoff in vehicular edge caching: algorithm and analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–13, 2021.

[23] H. Wei, H. Luo, Y. Sun, and M. S. Obaidat, "Cache-aware computation offloading in IoT systems," *IEEE Systems Journal*, vol. 14, no. 1, pp. 61–72, 2020.

[24] S. Bi, L. Huang, and Y. Zhang, "Joint optimization of service caching placement and computation offloading in mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 99, pp. 1–1, 2020.

[25] T. Zhu, T. Shi, J. Li, Z. Cai, and X. Zhou, "Task scheduling in deadline-aware mobile edge computing systems," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[26] W. Fan, Y. Liu, B. Tang, F. Wu, and H. Zhang, "TerminalBooster: collaborative computation offloading and data caching via smart base stations," *IEEE Wireless Communications Letters*, vol. 5, no. 6, pp. 612–615, 2016.

[27] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 99, pp. 1–1, 2020.

[28] H. Tang, C. Li, Y. Zhang, and Y. Luo, "Optimal multilevel media stream caching in cloud-edge environment," *The Journal of Supercomputing*, vol. 10, pp. 1–20, 2021.

[29] Q. Li, Y. Zhang, Y. Li, Y. Xiao, and X. Ge, "Capacity-aware edge caching in fog computing networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9244–9248, 2020.

[30] Y. Hao, Y. Miao, L. Hu, M. S. Hossain, G. Muhammad, and S. U. Amin, "Smart-edge-CoCaCo: AI-enabled smart edge with joint computation, caching, and communication in heterogeneous IoT," *Network, IEEE*, vol. 33, no. 2, pp. 58–64, 2019.

[31] C. Tang, C. Zhu, X. Wei, Q. Li, and J. J. P. C. Rodrigues, "Task caching in vehicular edge computing," *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021.

[32] J. Zhang, X. Hu, Z. Ning et al., "Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[33] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 99, pp. 1–1, 2020.

[34] J. Gao, S. Zhang, L. Zhao, and X. Shen, "The design of dynamic probabilistic caching with time-varying content popularity," *IEEE Transactions on Mobile Computing*, vol. 99, pp. 1–1, 2020.

[35] X. Meng, W. Wang, Y. Wang, V. Lau, and Z. Zhang, "Delay-optimal computation offloading for computation-constrained mobile edge networks," *2018 IEEE Global Communications Conference (GLOBECOM)*, 2019.

[36] N. Yousefian, J. Hansen, and P. C. Loizou, "A hybrid coherence model for noise reduction in reverberant environments," *IEEE Signal Processing Letters*, vol. 22, no. 3, pp. 279–282, 2015.

[37] S. L. Woodward, W. Zhang, B. G. Bathula et al., "Asymmetric optical connections for improved network efficiency," *J Opt Commun Netw*, vol. 5, no. 11, pp. 1195–1201, 2013.

[38] S. Surati, D. C. Jinwala, and S. Garg, "A survey of simulators for P2P overlay networks with a case study of the P2P tree overlay using an event-driven simulator," *Engineering Science and Technology, an International Journal*, vol. 20, no. 2, pp. 705–720, 2017.

[39] P. Dani and A. Thomas, "Bowditch's JSJ tree and the quasi-isometry classification of certain Coxeter groups," *Journal of Topology*, vol. 10, no. 4, pp. 1066–1106, 2017.

[40] Y. Li and B. Cheng, *An Improved K-Nearest Neighbor Algorithm and Its Application to High Resolution Remote Sensing Image Classification*, IEEE, 2009.

[41] O. A. Gbadamosi and D. R. Aremu, "Design of a modified Dijkstra's algorithm for finding alternate routes for shortest-path problems with huge costs," *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*, 2020.

[42] X. Q. Pham, T. D. Nguyen, V. D. Nguyen, and E. N. Huh, "Joint service caching and task offloading in multi-access edge computing: a QoE-based utility optimization approach," *IEEE Communications Letters*, vol. 99, pp. 1–1, 2020.

[43] W. B. Chu, L. F. Wang, Z. J. Jiang, and C. C. Chang, "Protecting user privacy in a multi-path information-centric network using multiple random-caches," *Journal of Computer Science and Technology*, vol. 32, no. 3, pp. 585–598, 2017.

[44] S. Ghandeharizadeh and S. Shayandeh, "Greedy cache management techniques for mobile devices," *IEEE International Conference on Data Engineering Workshop*, 2007.

[45] M. Kunjir, B. Fain, K. Munagala, and S. Babu, *ROBUS: Fair Cache Allocation for Multi-Tenant Data-Parallel Workloads*, Computer Science, 2015.

[46] Z. Qin, S. Leng, J. Zhou, and S. Mao, "Collaborative edge computing and caching in vehicular networks," *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1290129, 14 pages
https://doi.org/10.1155/2022/1290129

Review Article

Current Status and Security Trend of OSINT

Yong-Woon Hwang,[1] **Im-Yeong Lee**,[1] **Hwankuk Kim**,[2] **Hyejung Lee**,[3] **and Donghyun Kim**[4]

[1]*Department of Software Convergence, Soonchunhyang University, Asan 31538, Republic of Korea*
[2]*Department of Information Security Engineering, Sangmyung University, Cheonan 31066, Republic of Korea*
[3]*Department of Innovation and Convergence, Hoseo University, Cheonan 31066, Republic of Korea*
[4]*Department of Computer Science, Georgia State University, Atlanta 30303, GA, USA*

Correspondence should be addressed to Im-Yeong Lee; imylee@sch.ac.kr

Received 17 November 2021; Revised 4 January 2022; Accepted 26 January 2022; Published 18 February 2022

Academic Editor: Yan Huo

Recently, users have used open-source intelligence (OSINT) to gather and obtain information regarding the data of interest. The advantage of using data gathered by OSINT is that security threats arising in cyberspace can be addressed. However, if a user uses data collected by OSINT for malicious purposes, information regarding the target of an attack can be gathered, which may lead to various cybercrimes, such as hacking, malware, and a denial-of-service attack. Therefore, from a cybersecurity point of view, it is important to positively use the data gathered by OSINT in a positive manner. If exploited in a negative manner, it is important to prepare countermeasures that can minimize the damage caused by cybercrimes. In this paper, the current status and security trends of OSINT will be explained. Specifically, we present security threats and cybercrimes that may occur if data gathered by OSINT are exploited by malicious users. Furthermore, to solve this problem, we propose security requirements that can be applied to the OSINT environment. The proposed security requirements are necessary for securely gathering and storing data in the OSINT environment and for securely accessing and using the data collected by OSINT. The goal of the proposed security requirements is to minimize the damage when cybercrimes occur in the OSINT environment.

1. Introduction

Recent developments regarding the Internet of Things (IoT) and big data have caused the amount of data to increase boundlessly and accelerate the advancement of open-source intelligence (OSINT). The paths of information collection, including IoT, are becoming increasingly diverse, and the data are analyzed based on big data. Thus, deriving intelligence is becoming more important. Here, intelligence is translated as mental intelligence, confidentiality, and information. In the military and spy worlds, it is referred to as espionage. Every country worldwide gathers information regarding other countries. The amount of information collected by countries is called Intelligent Surveillance and Reconnaissance (ISR). Information collection methods like ISR are of three types, as Table 1 shows: OSINT, human intelligence (HUMINT), and technical intelligence (TECHINT) [1].

OSINT is the most basic method of collecting information, which is a form of collecting data through open sources (internet, broadcasting, papers, etc.) and processing them [2]. As open information is used, there are advantages, e.g., the information is collected in real-time, and the data are accessed easily and collected at a low cost. However, the importance of the information is lower than that of other information collection methods.

HUMINT indicates that humans extract or steal information. Simply, it refers to spies or secret agents. It has the advantage of obtaining high-quality information like first-class confidential information, however, there always exists a risk of betrayal and double espionage because people are involved [2].

Table 1: Information collection method.

OSINT	Gather intelligence using open information, data, and software. (i) Open information: information available in everyday life, such as internet, broadcasting, papers, and journals.
HUMINT	Intelligence gathered by humans through activities (e.g., spying, undercover operation). (i) White agent: can collect open information but espionage is not permitted. (ii) Black agent: steals confidential information (first-class confidential information, high-quality information) in secret.
TECHINT	Technology and information assets are used to gather enemy intelligence. (i) IMINT: UAV, reconnaissance planes, satellites, etc., are used to gather information. (ii) SIGINT: signals such as radio waves and radar signals are analyzed to gather information. (iii) MASINT: devices other than IMINT and SIGINT are used to gather information.

Recently, TECHINT emerged as an information collection method that uses technology and information assets to gather enemy intelligence. Here, the technology and information assets refer to devices that have the latest technologies for collecting information, such as imagery intelligence (IMINT) and signals intelligence (SIGINT). Its disadvantages are that the costs are high, and the reliability of the acquired information is low when a problem occurs in signals and radio waves.

Each of the three information collection methods has advantages and disadvantages, depending on the environment. In this paper, we examine and explain (See Section 2) OSINT, which is the basis for information-gathering methods. Currently, all users use OSINT technology when searching for data online. On this basis, users obtain information about the data they are looking for. However, from the perspective of cybersecurity, the use of data gathered by OSINT is a double-edged sword.

(i) On the positive side, data gathered by OSINT can be used as a means of resolving cybersecurity threats, which can track down cybercriminals or prevent cyberattacks before occurring.

(ii) On the negative side, data gathered by OSINT becomes the basis for attackers to create cybersecurity threats. In other words, an attacker can set a target to attack based on data, and after gathering related information, they can engage in various cybercrimes, such as hacking, malware, and denial-of-service (DoS) attacks [3].

To solve this problem, it is important to establish basic security requirements in the stages when data are collected and stored in the OSINT environment, as well as when users access the data. Currently, because anyone can access OSINT, the problem is that security-related requirements are not taken into consideration.

Therefore, this paper will explain the current status and security trends of OSINT. In particular, we focus on security awareness by mentioning the importance of OSINT from the perspective of cybersecurity and providing additional security requirements to resolve security threats occurring in the OSINT environment. It is expected to address the future problem of cybercrimes occurring when attackers misuse data collected by OSINT, and the goal is to reduce the cybercrime occurrence rate through security technology and minimize the damage in the event of an occurrence.

The security requirements proposed in this paper are basics, which can be applied to the OSINT environment, where data importance is high, rather than all OSINT environments. Here, the data importance refers to the data that are worth providing confidentiality and integrity for, such as security elements, because the value of the data processed by OSINT is high. Data with low importance are data that anyone can easily access and check, and it does not significantly affect cybersecurity threats. Therefore, it is necessary to apply basic security requirements and security technologies to data with high importance.

This paper consists of the following sections: Section 2 describes the background of OSINT. In detail, OSINT definition, structure, advantages, disadvantages, and examples of using OSINT are described. Section 3 mentions the basic requirements in OSINT and explains the security threats and cybercrimes arising when collected data are maliciously used. Section 4 describes the importance of OSINT from the cybersecurity perspective and presents common security requirements that are needed to solve the security threats mentioned in Section 3. Section 5 mentions the future challenges or necessary research in the OSINT field, and Section 6 concludes the paper.

2. Background of OSINT

This section describes the definition and structure of OSINT. It also describes the advantages and disadvantages of OSINT, and the examples of using OSINT.

2.1. Definition and Structure of OSINT. OSINT is a compound word for open source and intelligence. It refers to the overall process in which anyone can collect and analyze information based on open-source information and create useful information. Before discussing OSINT, we define each term as follows:

(i) *Intelligence*: refers to information and espionage. Specifically, it refers to information collected, processed, and reduced to satisfy explicit or understood needs [4].

(ii) *Open-source data (OSD)*: refers to unprocessed general data. Examples include images, photographs, survey data, audio data, metadata, and datasets, which can be obtained from public information.

(iii) *Open-source information (OSINF)*: refers to general data that have been partially filtered based on requirements or certain criteria. Examples include books, articles, and papers written on certain topics,

and they are characterized by some filtering before being processed. The OSINT data are the result of collecting and processing data according to the purpose of the OSINT tool. Therefore, OSINF is an essential prerequisite for OSINT, and investigators/information producers collect information for OSINT.

(iv) *OSINT*: it refers to data processed through open sources. In detail, it refers to data that has been processed through a search and filtering process to satisfy a specific request or standard purpose. The information is directly used in all intelligence contexts, and a large amount of data are summarized, sorted, and output for the OSINT tool.

(v) *Validated OSINT (OSINT-V)*: it is OSINT with a high degree of certainty/veracity. Data must be checked (verified) using a reputable OSINT source or a source that is not of OSINT. Validation is essential because some malicious users (attackers) tamper with OSINT analysis, produce inaccurate OSINT information, and spread it.

Media companies, colleges, journalists, and scholars have been analyzing OSINT data in the private sector, hundreds of years ago before the advent of the internet. In 2001, Wikipedia was established in the U.S., a nonprofit organization and website that collects, analyzes, and discloses OSINT. It is the world's largest private nonprofit OSINT collection, analysis, and open site on the internet. Currently, the importance of OSINT has rapidly emerged as much information overflows because of the development of computers and the internet in the twenty-first century. Since guidelines for OSINT tools are publicly available, companies and users can set up OSINT tools according to their purpose and collect data [5].

One of the most basic and important steps of OSINT is the search and collection of public information, and the collection of public information is primarily related to the search and acquisition of relevant specialized data. It is necessary to find ways to search and use a vast amount of information that has explosively increased nowadays. To find the desired information, it is necessary to design a structured OSINT search process by developing a series of search processes consistently and systematically [2].

To effectively collect data online, it is important to identify relevant websites corresponding to the invisible Web or deep Web and securing and organizing the list of these websites plays a key role. As they are not searched for in regular methods, their web addresses are obtained through offline sources, such as library searches, references in relevant books, expert reports, and interviews. OSINT data are collected by creating and managing separate lists of collected useful websites or online information [2, 6, 7].

Figure 1 illustrates the basic structure of OSINT. The OSINT process consists of collecting, processing, analyzing, and reporting data after identifying the specified data. Each organization has a modified structure of OSINT according to its purpose and requirements because OSINT requirements vary from one organization to another. However, the OSINT process consists of five steps as shown in Figure 1.

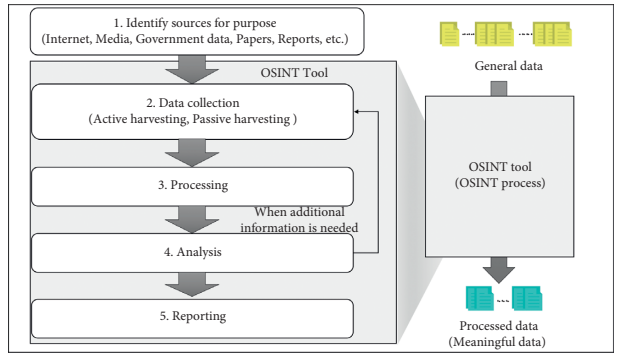


Figure 1: Structure of OSINT.

(i) *Step 1. Identifying the source*: set the information that the investigator (user) wants to obtain among numerous data. One needs to know where and how to get this information.

(ii) *Step 2. Data collection*: it is the stage to bring related data from identifying the source. When bringing in data, the harvesting step is classified into two types, namely active and passive, depending on the data collection method. In the active harvesting step, information is directly collected using a program or script on the target. The active type has a characteristic, which is that logs are left behind because it directly accesses the target. In the passive harvesting step, information is collected using Google, Netcraft, Whois, Recon-NG, Shodan, etc. The passive type has a characteristic, which is that no separate log is left behind because the information is collected using third-party applications.

(iii) *Step 3. Processing*: it is a step to obtain meaningful information by processing and refining the information obtained in Step 2. Since there is a lot of information in Step 2, the task of filtering much information is important in the processing step. Furthermore, it is important to consider the association between information, and the processing step requires a high-difficulty task that needs a lot of experience and perspectives.

(iv) *Step 4. Analysis*: in which the data refined in the processing step are processed according to the investigation purpose. For example, suppose there are evidence data (information) A, B, and C obtained by collecting and processing a variety of information to support an argument. Finally, a conclusion is reached if it is proved by A, B, and C that the argument is correct. If additional information is needed in the analysis step, the data collection and processing steps are continuously repeated to find an association between information to derive meaningful information.

(v) *Step 5. Reporting*: it is a method to summarize the contents up to Step 4 and write it in the form of a report. The reports are distributed and evaluated in various forms, such as evidence and analysis reports, depending on the organization (institution) that uses them. They include all source data that indicate the accuracy of data to give credibility to the data for the argument and evidence. As a result, much of the general data is processed into data that meets the criteria set by the investigator, resulting in meaningful data [2, 8, 9].

2.2. Advantages and Disadvantages of OSINT. As Table 2 shows, OSINT has advantages and disadvantages, depending on how the collected data are used [10, 11]. The advantages of using OSINT are as follows:

(i) *Fast/real-time information collection*: information collected by OSINT is quickly obtained through open sources, and the data are tracked in real-time. To obtain desired data, the user searches data by relying on a variety of OSINF—such as searching the internet data, watching YouTube/TV, and reading books—rather than collecting information from one place. It has the advantage of ensuring prompt data access.

(ii) *Secure acquisition of much data*: the data collected by OSINT secures much data that supplements the gathering of secret intelligence. In HUMINT, the obtained data are few because only a few agents are used, however, in OSINT, there is the advantage that much data can be obtained through open sources. Excellent data (meaningful data) are obtained if a considerable amount of data is processed using OSINT. Furthermore, as OSINF is accessible by anyone, it is legal, and there is the advantage of low risk in terms of a security issue, which means that data are obtained securely.

(iii) *Clarity of sources*: in HUMINT, the credibility of data is questionable because the source of the information that the agent obtains is unclear. In contrast, data collected by OSINT ensures credibility because the clarity of the open sources is guaranteed by a validation process.

(iv) *Convenience and ease of access*: not everyone can access data easily because data access rights are set in such a way that only authorized users can access confidential and high-quality data. In contrast, anyone can easily access information collected by OSINT and use data conveniently according to the user's requirement.

(v) *Low cost*: OSINT has the advantage of obtaining data at a low cost, compared to the cost of training agents in HUMINT and the cost of collecting data using the latest equipment, such as satellites and unmanned aerial vehicles (UAV) in TECHINT.

Disadvantages of using OSINT are as follows:

(i) *The amount of information is too large*: the more information the user has, the harder it is for the user to output reliable data using OSINT. If incorrect information is mixed among the evidence data of several factors supporting an argument, it may reduce the credibility of the data, which may result in false information in the argument data. Currently, because much data is searched for in open sources, it takes time and effort to detect false information and select reliable data.

(ii) *Organizational perception and prejudice of intelligence agencies*: in the organizational culture of intelligence agencies, the value of data collected by OSINT is underestimated, and the importance of data is not considered because anyone can access and use the data.

(iii) *Security issues and technical constraints*: intelligence agencies use internal computer networks because of security issues, which limit the use of OSD using the internet. As a result, analysts at the intelligence agencies exhibit a passive attitude toward the use of OSINT data. Computer security experts are endeavoring to prepare methods of freely using OSD while solving security problems.

(iv) *Cornerstone of cybercrimes when misused*: anyone can access the data collected by OSINT. However, there is the disadvantage that the data collected by OSINT can be the basis of committing cybercrimes because of users with malicious goals. Therefore, research is required on security requirements (measures) and technology that can minimize the damage of cybercrimes, even if users use OSINT's data for malicious purposes.

2.3. Examples of Using OSINT. In recent years, all internet users have been using various OSINT technologies when searching for data online. They collect OSINT data in some form, regardless of whether they are companies, schools, universities, or individuals. The intelligence and investigation agencies of major countries, including the UK and US, have recognized the importance of OSINT early on and are investing systematically and actively building these systems. Related companies are also developing several types of OSINT solutions. In South Korea, a variety of research and education are underway using OSINT [12]. The subsequent paragraphs show the typical examples of OSINT uses [13].

In terms of law enforcement agencies, police use OSINT sources to protect citizens from abuse, sexual violence, identity theft, and other crimes. It is done by monitoring social media channels for keywords and photographs that help prevent crimes before they increase. Law enforcement agencies use OSINT to monitor and track criminal networks in many countries. For example, using OSINT tactics, they collect information about criminals (persons of interest) and

Table 2: Advantages and disadvantages of OSINT.

Advantages	Disadvantages
(i) Fast/real-time information collection (ii) Secure acquisition of large data (iii) Clarity of sources (iv) Convenience and ease of access (v) Low cost	(i) The amount of information is too large (ii) Organizational perception and prejudice of intelligence agencies (iii) Security issues and technical constraints (iv) Cornerstone of cybercrimes when misused

create a complete profile of each criminal. Furthermore, they use OSINT sources to track online counterfeiting and copyright violations and use them as tools for dealing with various cybersecurity threats.

In terms of business corporations, information is power. Businesses use OSINT sources to investigate new markets, monitor competitors' activities, plan marketing activities, and predict anything that may affect the current operation and future growth. In the past, the use of OSINT sources was limited to large corporations with sufficient intelligence budgets, however, nowadays, because of the broad use of the internet, small companies with limited budgets can use OSINT sources and incorporate the obtained information in their business plans.

As opposed to the above advantageous uses, OSINT sources can be used in malicious ways, and terrorist organizations can use OSINT sources to plan attacks. They can gather information about the target (when investigating the target location) before attacking, analyze social media sites to secure more fighters, obtain military information disclosed accidentally by the government (e.g., a method of making explosives), and use various media channels to spread propaganda around the world [13, 14]. Furthermore, data collected by OSINT can become a cornerstone for committing various cybercrimes (See Section 3).

In summary, using OSINT data is important because the results may be a double-edged sword, depending on the aspect of using the data collected by OSINT. For information on overseas, OSINT projects, and open sources, refer to the following papers [5, 15].

3. OSINT General Requirements and Security Threats

3.1. OSINT General Requirements. The requirements of OSINT vary depending on the purpose, organization, and data to be derived from OSINT. Accordingly, the process of OSINT varies. Figure 1 illustrates the basic process of OSINT. The data processed using the OSINT process is stored in the built OSINT database, and it is used and analyzed. In this paper, it is indicated that the data processed by the OSINT tool is stored in the OSINT tool storage. Figure 2 is a schematic diagram of the process of storing and utilizing data collected in OSINT. If the user misuses the stored data, the contents of the security threat are also briefly included. The requirements are composed of the aspect of collecting and storing data and the aspect in which the user accesses and uses the collected data. OSINT data collectors, who collect and store data with OSINT tools, should basically provide data curation, data integrity, and reliability.

(i) *Data curation*: in a museum or art gallery, the term "curator" refers to a person who decides which works to exhibit. In other words, curation is a term that refers to an act of selecting and providing data in the Big Data era, where much data exists. Curation is an essential element for finding the most valuable information by efficiently using a limited "time resource" in the Big Data era. Data curation is provided for the goals of data search, data quality assurance, value addition, reuse, and preservation over time, which includes the creators/recorders and the selection and evaluation of record repositories [16]. In the past, the curation process was carried out using simple information collection. Recently, data are processed more sophisticatedly using data-based deep analysis and machine learning using artificial intelligence (AI) to increase the use-value of data. It is a requirement needed in the OSINT data collectors who collect general data and process it as valuable data.

(ii) *Guarantee of data integrity*: when storing the collected data in the OSINT tool storage, data integrity must be ensured. Here, integrity refers to maintaining the consistency and accuracy of data, and the stored data must not be modified (forgery and alteration) by someone without permission [17, 18]. If anyone accesses and modifies data in an open space, the reliability of data may degrade. In addition, users may accept and spread incorrect information because of the tempered data, laying a cornerstone for cybercrimes like fake news [16]. Therefore, a guarantee of data integrity is an essential requirement in the OSINT process.

(iii) *Guarantee of data reliability*: data reliability was an essential element when users used the collected data [16]. To guarantee data reliability, it was required to validate the data integrity and data sources. Usually, the validation process of sources was performed through the OSINT process, and guaranteeing the sources of the basis for claiming the legitimacy of data and providing integrity for the sources were essential requirements to increase data reliability.

Most OSINT tools did not consider the requirements needed when accessing and using stored data. It means that the collected data can be used by everyone, including general users. Regarding security, one of the biggest problems was that the users of OSINT might accidentally disclose sensitive assets and information on the internet [19]. It was a severe problem because OSINT was used for security purposes. As

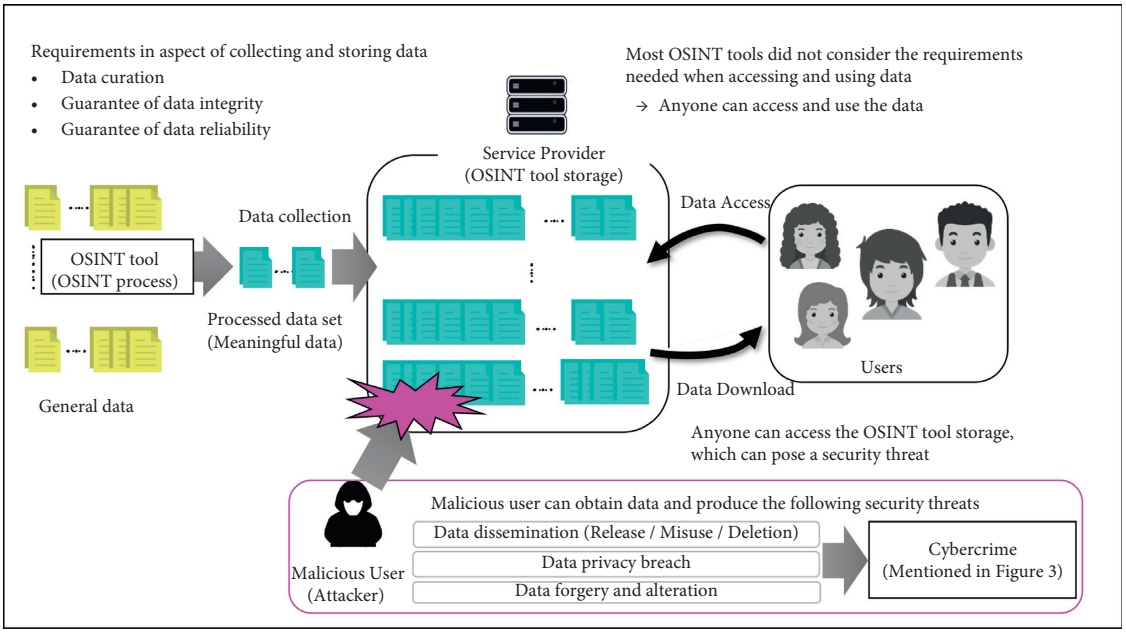


Figure 2: OSINT process and possible security threats.

the source information was disclosed as the word OSINT suggested, anyone could access and use data. Therefore, it was important for users to use data with ethical awareness [16]. Here, ethical use referred to rules that define the allowed actions or proper behaviors. In short, it meant that users should use data for legitimate purposes and not malicious purposes. If users did not have ethical consciousness, it would be the basis for causing various cybercrimes (See Section 3.2).

3.2. Security Threats of OSINT. Figure 2 shows that various security threats, such as data dissemination, data privacy breach, and data forgery and alteration exist in the OSINT environment, which can lead to cybercrime. In Figure 2, the cybercrime mentioned refers to various cybercrimes, such as hacking, data loss, denial of service attack, spreading viruses, and fake news, as well as illegal use in various fields, such as games and financial shopping [19–21]. Figure 3 is a schematic diagram of the contents of cybercrimes that can occur through security threats. Since anyone has access to the data, a malicious user can access the stored data. The acquired data can be disseminated, which can be the basis for hacking financial crimes and virus spread. It can also falsify data, giving users inaccurate information or spreading fake news to create confusion.

Moreover, establishing the reliability of data extracted by OSINT tools is a very difficult problem. It is required to decide who would determine the data reliability and whether the reliability of sources was credible. In the OSINT domain, general trust was often based on perspectives, ideologies, prejudices, beliefs, or product marketing, regardless of value or truth [4]. Public information obtained in secret might be reported early or analyzed immediately and used for a purpose of trusting or disbelieving [2]. Therefore, when general data are processed using OSINT, the evidence for ensuring the data reliability and the guarantee of the data sources must be validated. Thereafter, if the processed data are altered by malicious users, the data reliability would drop, which implied that this needed to be dealt with in advance.

Another issue related to open sources is the concern about privacy breach in the internet age. People use the internet for various purposes, including data collection, analysis, and communication. Ceaselessly expanding social media platforms, such as Facebook and Instagram, are information channels that offer little protection from hackers and malicious attackers [22]. As a result, public trust in privacy was waning, and the problem was that companies (data mining companies) knew information about users, however, the audience knew little about what the data mining companies knew about them [4]. Therefore, security is required for sensitive information, such as personal information. Despite applying general requirements of OSINT mentioned in Section 3.1, various security threats arise, which further contribute to cybercrimes (See Figure 3). The security threats that arise are as follows:

(i) *Data dissemination (release/misuse/deletion)*: if the confidentiality of the collected data is not provided, attackers can obtain data and create various security threats based on them, which might cause the damage of data loss. The collected information of OSINT might vary, depending on the organization and the specialty field, however, if the attacker deleted data, there would be a delay when the users of OSINT data search and check the data they want. Assuming the financial sector, an attacker might use the collected data to obtain users' personal information, financial information, and information about various elements. It would be used as basic data for hacking and might lead to cybercrimes for obtaining sensitive or financial information and monetary gains, such as identity theft and financial

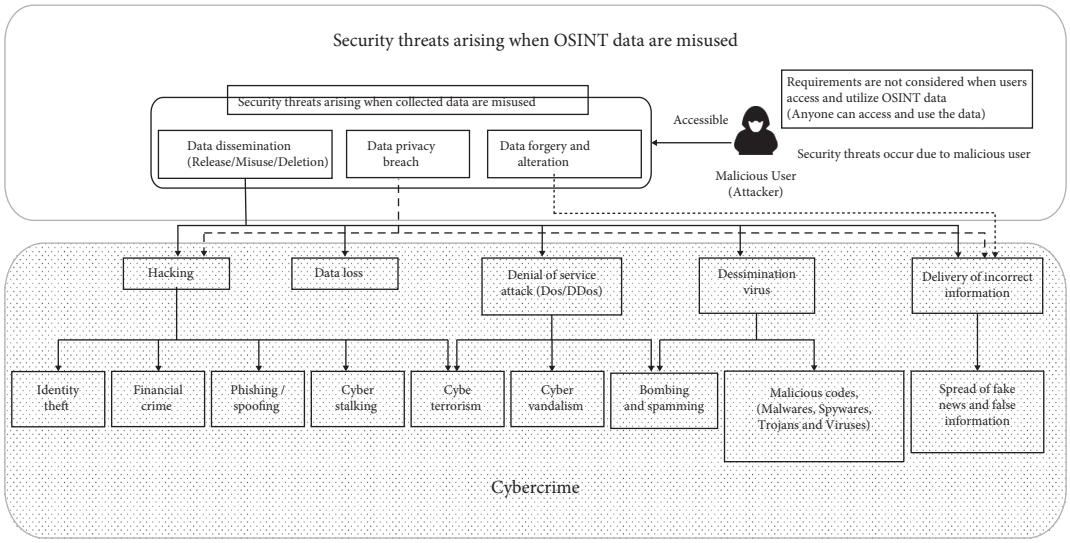


Figure 3: OSINT security threats and cybercrimes.

crimes [21]. Also, because attackers hacked systems based on the obtained OSINT data, it was critical in the national security aspect because not only financial crimes but also, on a larger scale, cyber terrorism activities, such as network overload and DoS attack, could arise [19]. Therefore, if the data collected by OSINT are sensitive and important, it is important to provide confidentiality and the integrity of data, and various requirements, such as user authentication and access control for accessing data, were additionally needed. Furthermore, requirements, such as data backup and recovery, were additionally needed to respond to a loss of data collected by OSINT.

(ii) *Data privacy breach*: if an attacker identified the contents of the collected data, the data for user information (personal information) included in the data could be collected, based on which various security threats could arise [23]. As a typical example, there was a possibility of cybercrimes through the leaking of personal information, which was used as criminal data by the attacker who wanted monetary gains, and cybercrimes, such as identity theft and voice phishing, might occur [24]. The leak of personal information itself caused damages like personal information infringement, however, there was a possibility of additional secondary and tertiary damage. In particular, the attacker set the attack targets based on the personal information of the users and attacked them by spreading viruses or malware. Therefore, additional security measurements were needed, such as anonymity and de-identification, which protect sensitive information, such as the confidential information of users contained in the data collected by OSINT.

(iii) *Data forgery and alteration*: attackers forged or altered the collected data, causing various security threats. If numerically expressed data were altered, the different results of statistical data or figures would be output, and users might misunderstand the information, including the false values in the information. A typical example is fake news, which is rapidly spread to users through social media by manipulating the content and source (evidence data) of data [21]. As a result, the users would be confused about the authenticity of the information, making the collected data less dependable. Therefore, additional security requirements must be in place to ensure the data integrity and guarantee the data source to respond to forgery and alteration of data collected by OSINT.

As such, additional security requirements are needed in OSINT to deal with security threats and cybercrimes (see Section 4). If the importance of the data collected by OSINT was high, security technology, such as user authentication and access control for accessing the collected data, was additionally required. It minimized the security threats that users produced using the data collected by OSINT by making sure in advance that the users would not be able to perform malicious activities, such as data leakage, forgery, and alteration, by accessing data.

4. Role of OSINT from Cybersecurity Perspective

In this section, we explain the importance of OSINT from a cybersecurity perspective. Furthermore, we propose commonly needed security requirements to solve with security threats arising from the misuse of OSINT data.

4.1. Importance of OSINT for Cybersecurity. Regarding cybersecurity, the aspect of using data collected by OSINT could be viewed as a two-edged sword [25]. If the data collected by OSINT were used in the positive aspect, a

considerable amount of data could be obtained compared to secret intelligence data, and on this basis, trends and situations of enemy countries or countries where there were no spies could be examined [2, 3]. Furthermore, if data collected by OSINT were used properly in the security aspect, cybercrimes, such as cyber security threats and cyber terrorism activities, that might occur in cyberspace could be prevented in advance. At present, studies have been continuously conducted to respond to cyberattacks using OSINT [14]. According to a report by the U.S. Office of Homeland Security, the use of data collected by OSINT included general intelligence, advanced warnings, domestic counterterrorism, protection of important infrastructures (including cyberspace), protection against critical terrorism, and emergencies in the domain of important missions [16]. Therefore, the management of data collected by OSINT was crucial because the use of OSINT data was important in terms of cybersecurity. In other words, it would be important for intelligence, security, and public safety agencies in terms of cybersecurity to collect a considerable amount of data from various sources, including criminal records of terrorism incidents and cybersecurity threats, process them into valuable (meaningful) data using OSINT, and securely manage the processed data [20].

If the data collected by OSINT were used in the negative aspect, the attacker could set the target based on profiling. Then, after gathering target information, the attacker commits various cybercrimes, such as SPAM, malware, hacking, DoS attack, phishing, the violation of digital property rights, confidential information infringement, and dissemination of false or confidential information [26, 27]. Cybercriminals could not be easily tracked or caught because they use anonymity and camouflage opportunities through web-based communication to perform malicious activities. Most of these cybercrimes were aimed at user identity theft, stealing sensitive information, and monetary gains. However, the following cases were considered serious cybercrimes in terms of national security: crimes of disrupting legitimate network operations, overloading networks, or denying network services by exploiting loopholes, bugs, improper configuration of software services, or raising false political issues by disseminating incorrect information [28–30].

As OSINT had both positive and negative aspects depending on the data utilization from the cybersecurity perspective, users must make effective use of data collected by OSINT. In other words, the use of OSINT must be limited to legal activities and nonmalicious purposes, and basic security requirements (measures) were additionally needed to minimize the damage, even if the attackers misused the collected information [24].

4.2. Essential Security Requirements When Using OSINT. Various security threats existed in OSINT. As Section 3 mentioned, the basic security threats included data dissemination (release/misuse/deletion), data privacy breach, data forgery, and alteration. To solve these problems, OSINT tools commonly needed additional basic security requirements applied to cloud or IoT environments [18, 31, 32]. Figure 4 is a schematic diagram of the security requirements required when using OSINT. The security requirements in Figure 4 consisted of the security requirements needed in data collection, the storage stage of OSINT, and the security requirements needed when users accessed and used the OSINT data. The details of the requirements are as follows:

4.2.1. Security Requirements Needed When Collecting and Storing OSINT Data

(i) *Data encoding/data encryption*: in general, integrity existed in the data processed by OSINT, however, because they were publicly available information, data confidentiality must be provided, depending on the importance of the data. Confidentiality refers to the prevention of unauthorized access to secure information, and only legitimate users can check the data [18]. Data encoding and encryption technology could be applied for this. The term "data encoding" referred to changing the shape of the information stored in a file to something else according to the purpose and format used when storing or transmitting data. It aimed to increase the data usability in other systems and reduce the space required for storage. The term "data encryption" referred to a method of using an encryption algorithm to conceal the contents of data so that only authorized users could read them, and the purpose was to provide data confidentiality. Encryption was used by mixing symmetric and public-key methods, and lightweight algorithms were often used to reduce the computational amount of encryption [23–34]. However, proxy re-encryption or attribute-based encryption was used if the scope was an environment, where anyone could access and obtain encrypted data in an open space. Proxy re-encryption is an encryption technology that converts the ciphertext without decrypting it so that the proxy can decrypt the ciphertext encrypted with user A's public key with user B's private key [35–37]. Attribute-based encryption is a technology that encrypts and decrypts users' attributes (e.g., affiliation, position, etc.) and an access structure created based on them [38–40]. Both proxy re-encryption and attribute-based encryption are cryptographic techniques, in which users (many unspecified users) accessed and obtained data (ciphertext) when satisfying the policies/conditions of the ciphertext. In other words, the confidentiality of the data is provided, The data can be securely protected because only the users who were authorized to access the data could see the contents of data. Also, if the data were encrypted and stored in the OSINT tool, the ciphertext requested by the user can be easily searched for and transformed without the process of decrypting the ciphertext using searchable encryption [41, 42] and homomorphic encryption [43, 44]. The

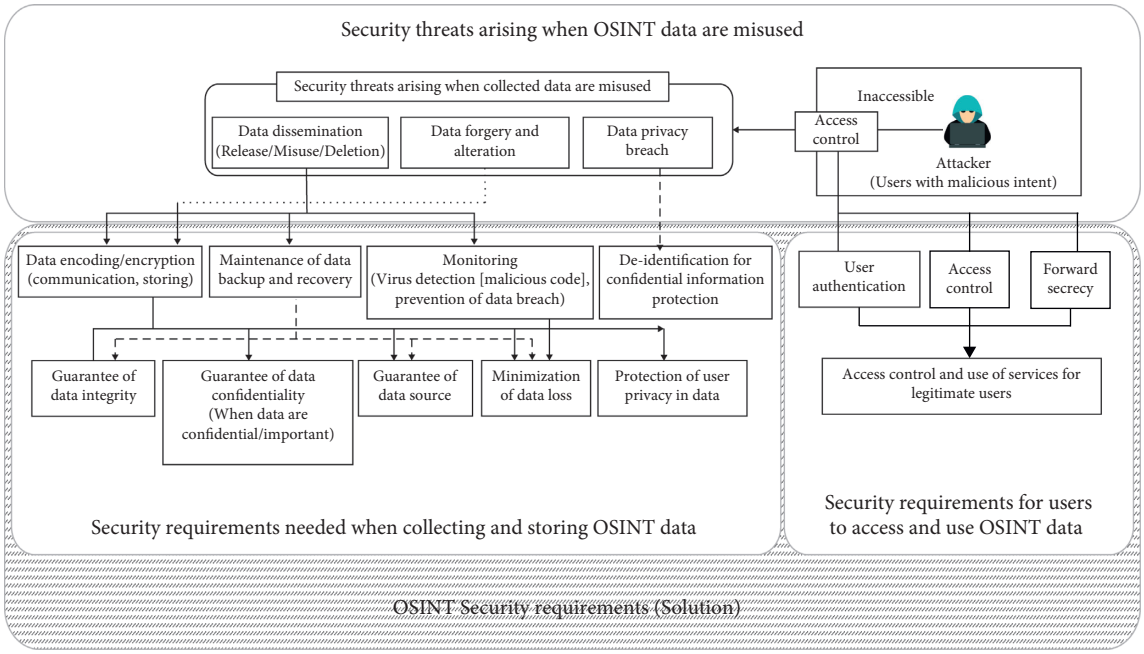


Figure 4: Security requirements needed when using OSINT.

aforementioned encryption techniques were usually used in cloud environments, where the service providers were not trusted. Therefore, it is a basic security requirement to securely store and manage the data collected in the OSINT environment, which is an open space. As such, security elements (data integrity, confidentiality, and guarantee of data source) were provided for the data processed and stored through data encoding/encryption processes. It prevented the loss of data (prevention of data leakage, user privacy breach, etc.) in advance.

(ii) *Maintenance of data backup and recovery*: damage caused by the loss of data because of system crashes, data alteration, and data deletion could not be ignored. Therefore, settings for the backup and recovery of data sets collected by OSINT were important. Data backup and recovery were the processes of backing up data in the case when data loss occurred and configuring the security systems, and in the end, it facilitated the recovery of the lost data [45]. However, considering the backup cost, data recovery cost, and loss cost, it was inefficient to configure the backup and recovery for all data because data collected by the OSINT tool were big data—although they might vary depending on the environment. Therefore, research was required on the operation methods of data backup and recovery to minimize the damage caused by data loss, and this was an additional security requirement.

(iii) *Monitoring*: if OSINT service providers provided additional security elements, such as security audits and security management, the security of the data collected by OSINT would be strengthened. In particular, monitoring should be performed in real-time to detect viruses and malware and prevent users from leaking data [46, 47]. Security audits were activities performed to check whether security activities were conducted appropriately in companies [48]. Security audits were performed according to the policies defined in the OSINT tool and the security activities configured. These were important because security was managed for users and stored data.

(iv) *De-identification of Personal Information*: as data processed by OSINT were open, the users who obtained data could know the contents of personal and sensitive information of users contained in the data [6, 25]. Therefore, if legitimate users obtained data, security technology was required to protect privacy, such as personal and confidential information expressed inside the data. The technology that protected privacy included de-identification, which anonymized personal information. De-identification was a privacy protection technique for reducing the risk of private life infringement by providing statistical anonymity to big data containing sensitive information [49, 50]. Based on this, pseudonymous information and anonymous information can be obtained. Here, pseudonymous information referred to the processing of personal information based on methods, such as partially deleting personal information or replacing some or all of it so that certain users could not recognize it without additional information. Anonymous information referred to the processing of personal credit information so that specific individuals, i.e.,

the subjects of credit information, could not be recognized. The information could not be identified, even if it was combined with additional information, and it was not subject to the Personal Information Protection Act. Therefore, research and security requirements were needed for anonymization of sensitive data, such as de-identification technique, to protect sensitive information contained inside OSINT data. It would minimize cybercrimes, such as privacy breaches and identity theft, caused by user privacy breaches [51, 52].

(v) *Other requirements for guaranteeing data integrity and sources*: the provision of data integrity and guarantee of sources were essential elements because they were related to data reliability for the users of OSINT data. Data integrity and source guarantee are important factors in the stage of collecting and storing data in the OSINT environment. To provide them, signature technology and blockchain technology were typically used. Signature technology referred to a technology that confirmed the signee and showed that the signee signed the data. A variety of signature technologies have been studied and used to increase the reliability of the data sent by the data owner (data sender) [53–55]. Blockchain technology referred to unchanging shared ledgers used for the efficient process of recording transactions and tracking assets in a business network. Among various features, data integrity was provided because transactions for data were recorded, and the recorded transactions were difficult to modify or delete, even for the system administrator [56, 57]. To date, studies have been continuously conducted to ensure data integrity using signature and blockchain technologies in IoT or cloud environments. It can also be used as a way to provide integrity and guarantee sources for data collected and processed by OSINT tools and stored. The use of signature technology or the introduction of blockchain is one of the ways to guarantee integrity and sources. However, the adoption of blockchains should be carefully considered according to the environment since blockchain technology requires high infrastructure.

4.2.2. Security Requirements for Users to Access and Use OSINT Data

(i) *User authentication*: usually, the authentication process was performed to determine who the user was and whether the user had the right when the user tried to access and obtain data on IoT, cloud, or Web. Simply, authentication referred to a process of inputting a user ID and password for accurate verification of the user. In particular, the authentication process was required to access servers, such as a cloud, and it was also required between the users for sending and receiving data [58–61]. OSINT data used in the open space environment was publicly available, and anyone could access and use them. It was considered an advantage and a disadvantage, and as mentioned, authentication technology for users trying to access the OSINT tool was needed, depending on the importance of the data. Authentication technology was typically classified into knowledge-based authentication, ownership-based authentication, biometric authentication, and behavior-based authentication. Usually, it was basic to authenticate a user with a token and certificate issued from the OSINT tool provider after registering the user. Also, one method is to use an identity authentication service using a decentralized identity (DID), which has become an issue recently [62, 63]. DID is a technology in which the user has the authority to control his/her own information, and authentication can be performed with a minimum amount of information compared to the identification method controlled by the existing central system. As a zero-knowledge-proof method, authentication can be performed without disclosing user information [64]. The two methods are the authentication methods that could protect the privacy and sensitive information of users. The method of applying authentication technology might be different, depending on the environment, however, currently, many security threats might exist since any user could access the OSINT tool. They might induce various cybercrimes, as mentioned in Figure 3. Therefore, the malicious use of OSINT data by users should be prevented in advance by at least adopting an authentication process.

(ii) *Access control*: the term access control referred to a function that permitted or denied someone from using something (service). In general, it referred to the user's rights to a service. Regarding information protection, the procedure for access control was conducted to identify the user by the user ID and perform the authentication using the password, token, or signature. Afterward, by granting a security level and a service privilege level according to the Access Control List, authorization/permission would be granted, based on which, the user could use the service [65]. Assuming that the OSINT tool service provider issued a token after user registration, the issued token would contain the rights to access the OSINT tool and the rights and services to use data. In other words, anyone could access the open space and use the data, and the various cybercrimes mentioned in Figure 3 could occur. To deal with this problem, access control technology was required in addition to authentication technology.

(iii) *Forward secrecy*: forward secrecy was important when the OSINT tool was examined from a cloud perspective. Forward secrecy referred to the encrypted communications and sessions recorded

in the past that could not be retrieved [17]. In other words, it was to assume that the user had a token containing the rights for accessing the OSINT tool. If the user's registration period expires or the registration for the OSINT tool is withdrawn, the user's right to access the OSINT tool must expire. Withdrawn user should not be able to access the OSINT tool to obtain the content of the collected data. Moreover, backward secrecy was not considered because when registering in the OSINT tool, the user received the service rights for reading the information in the OSINT tool.

5. Challenges of OSINT

This section describes the research challenges required for future OSINT development. Research should be conducted on the OSINT process for the efficient extraction of the data that the user wants from countless big data in OSINT. Furthermore, additional research is required according to the situation to improve the security of the collected data. As Section 3 mentioned, research for providing general requirements should be provided, and additional research is required to proactively prevent security threats and cybercrimes that might occur if the users use the data of OSINT maliciously (See Section 4). The elements needed for future OSINT development might vary depending on the situation, however, in common, the following challenges should be continuously studied [13, 15]. It is similar to the requirements mentioned earlier but will have a major impact on the evolution of OSINT.

(i) *Efficient and reliable data filtering*: to extract the data that the user wants from OSINT, much data should be collected and effectively filtered [13]. It consumes a huge amount of time and human resources, depending on the amount of data. Organizations or users will utilize automation tools (organizations have their own AI filtering tools) and skill sets to filter data according to purpose. However, the accuracy and reliability of the data extracted when there are software defects in the set of automation tools and techniques are questionable. Therefore, it is important to continuously check the automation tool that is the standard for data filtering, and research on the verification of the extracted data is necessary. It remains a challenge for collectors who collect and filter data from OSINT.

(ii) *Provides data transparency*: the reliability of the collected data was a critical issue in the aspect of the users using OSINT's data. In particular, the verification of sources for claiming the legitimacy of data during the OSINT process increased data reliability significantly. However, in the case of obtaining OSINT data by illegal means, the user might intentionally discard or hide important sources, however, no countermeasure existed in OSINT. It is important to keep a record of the sources on which the data extracted from OSINT in the future is based on credibility. Through this, users need to be provided with data transparency, and research on this still remains a challenge. It also requires the integration and collaboration of many OSINT tools to provide data reliability and transparency [15].

(iii) *Lack of validation of privacy management procedures*: many companies, such as Facebook and Google, collect much data from online users for commercial intelligence. Data that was collected online included not only general data produced by users but also sensitive information, such as names, birthdays, addresses, and passport numbers. Many companies have revealed that they collect and manage data anonymizing to justify data collection, however, it is unknown whether this is being done properly [13]. It is similar to the privacy problem that may arise when collecting data depending on the purpose of the OSINT tool. If anonymous information was mixed in a large amount of data, it would be questionable whether the processed data were reliable and whether the provided anonymization method was properly executed for the data. Therefore, research into the validation of privacy management procedures in OSINT data still remains a challenge. From a legal perspective, OSINT should use data while respecting the data protection policy according to the law [13].

6. Conclusions

In this paper, we explained the current status and security trends of OSINT and proposed security requirements needed in OSINT from a cybersecurity perspective. Specifically, to deal with security threats arising when data collected in the OSINT environment are misused, we proposed basic security requirements according to the steps of collecting and storing data in OSINT and the steps involved when the users accessed the data collected by OSINT. They were similar to the security requirements required in cloud environments or IoT environments. It was crucial to provide data confidentiality and integrity for important data through data encoding/encryption. Anonymization techniques, such as de-identification, were required to protect user privacy in data. Data backup and recovery processes were also needed to minimize data loss, and users accessing the data collected in OSINT should be managed through authentication and access control. Furthermore, based on forward secrecy, users whose registration was canceled should not have access to the data collected by OSINT. In the OSINT environment where these security requirements were satisfied, attackers would not be able to commit cybercrimes easily. The best method for dealing with cybercrimes was to use the data collected in OSINT only in a positive aspect, and it was important for users to have ethical awareness.

In future research, we will propose a secure OSINT model based on the security requirements presented in this

paper. Moreover, we need to conduct a study to improve the security performance of the secure OSINT model. It can be accomplished by testing whether the security threats mentioned in this paper can arise in the secure OSINT model.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) (No. 2021-0-00358, AI·Big data based Cyber Security Orchestration and Automated Response Technology Development)

References

[1] P. Casanovas, "Cyber warfare and organised crime. a regulatory model and meta-model for open source intelligence (OSINT)," *Ethics and Policies for Cyber Operations*, pp. 139–167, 2017.

[2] W. H. Lee, M. W. Yun, and J. S. Park, "Intelligence in the internet Era: understanding OSINT and case analysis," *Korean Security Journal*, vol. 34, pp. 259–278, 2013.

[3] K. Shin, fnm au, J. Yoo et al., "A study on building a cyber at tack database using open source intelligence (OSINT)," *Jouranl of Information and Security*, vol. 19, no. 2, pp. 113–121, 2019.

[4] B. H. Miller, "Open source intelligence (OSINT): an oxymoron?" *International Journal of Intelligence & Counter Intelligence*, vol. 31, no. 4, pp. 702–719, 2018.

[5] M. E. Hayden, *Guide to Open Source Intelligence (OSINT)*, Tow Center for Digital Journalism, Columbia University, New York, NY, USA, pp. 1–61, 2019.

[6] S. Chauhan and N. K. Panda, "Open source intelligence and advanced social media search," *Hacking Web Intelligence Open Source Intelligence and Web Reconnaissance Concepts and Techniques*, Elsevier, Amsterdam, Netherlands, pp. 15–32, 2015.

[7] S. Chauhan and N. K. Panda, "Understanding browsers and beyond," *Hacking Web Intelligence Open Source Intelligence and Web Reconnaissance Concepts and Techniques*, Elsevier, Amsterdam, Netherlands, pp. 33–52, 2015.

[8] M. Danda, "Open source intelligence and cybersecurity," Unpublished Master's Thesis, Webster University, Webster Groves, MO, USA, 2019.

[9] A. Kanta, I. Coisel, and M. Scanlon, "A survey exploring open source Intelligence for smarter password cracking," *Forensic Science International: Digital Investigation*, vol. 35, Article ID 301075, 2020.

[10] W. Chun, "Open source intelligence in the information age," *Journal of National Intelligence Studies*, vol. 1, no. 1, p. 151, 2008.

[11] T. Dokman and T. Ivanjko, "Open source intelligence (OSINT) issues and trends," *The Future of Information Sciences*, pp. 191–196, 2020.

[12] L. Benes, "OSINT, new technologies, education: expanding opportunities and threats, a new paradigm," *Journal of Strategic Security*, vol. 6, no. 3, pp. 22–37, 2013.

[13] N. A. Hassan and R. Hijazi, "The evolution of open source intelligence," *Open Source Intelligence Methods and Tools*, pp. 1–20, Apress, Berkeley, CA, USA, 2018.

[14] D. Wells, "Taking stock of subjective narratives surrounding modern OSINT," *Open Source Intelligence Investigation*, pp. 57–65, 2016.

[15] J. Pastor-Galindo, P. Nespoli, F. Martinez Perez, and G. M. Perez, "The not yet exploited goldmine of OSINT: opportunities, open challenges and future trends," *IEEE Access*, vol. 8, pp. 10282–10304, 2020.

[16] F. Tabatabaei and D. Wells, "OSINT in the context of cybersecurity," in *Open Source Intelligence Investigation: From Strategy to Implementation, B. Akhgar*, P. S. Bayerl and F. Sampson, Eds., Springer, Cham, Switzerland, pp. 213–231, 2016.

[17] F. Alkhudhayr, S. Alfarraj, B. Aljameeli, and S. Elkhdiri, "Information security: a review of information security issues and techniques." a review of information security issues and techniques," in *Proceedings of the 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pp. 1–6, Riyadh, Saudi Arabia, May 2019.

[18] R. Barona and E. M. Anita, "A survey on data breach challenges in cloud computing security: issues and threats," in *Proceedings of the 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pp. 1–8, Kollam, India, April 2017.

[19] A. Yeboah-Ofori and A. Brimicombe, "Cyber intelligence and OSINT: developing mitigation techniques against cybercrime threats on social media," *International Journal of Cyber-Security and Digital Forensics*, vol. 7, no. 1, pp. 87–98, 2018.

[20] H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: from big data to big impact," *MIS Quarterly*, vol. 36, no. 4, pp. 1165–1188, 2012.

[21] W. Kim, O.-R. Jeong, C. Kim, and J. So, "The dark side of the Internet: attacks, costs and responses," *Information Systems*, vol. 36, no. 3, pp. 675–705, 2011.

[22] G. Li, Z. Cai, G. Yin, Z. He, and M. Siddula, "Differentially private recommendation system based on community detection in social network applications," *Security and Communication Networks*, vol. 2018, Article ID 3530123, 2018.

[23] M. Siddula, Y. Li, X. Cheng, Z. Tian, and Z. Cai, "Privacy-enhancing preferential lbs query for mobile social network users," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–13, 2020.

[24] B. J. Koops, J. H. Hoepman, and R. Leenes, "Open-source intelligence and privacy by design," *Computer Law & Security Review*, vol. 29, no. 1, pp. 676–688, 2013.

[25] P. Chen, "Data mining applications in e-government information security," *Procedia Engineering*, vol. 29, pp. 235–240, 2012.

[26] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart internet of things systems: a consideration from a privacy perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[27] W. Sun, Z. Cai, Y. Li, and F. Liu, "Security and privacy in the medical internet of things: a review," *Security and Communication Networks*, vol. 2018, pp. 1–10, 2018.

[28] R. Buch, D. Ganda, P. Kalola, and N. Borad, *World of Cyber Security and Cybercrime*, STM Journals 2017, vol. 4, no. 2, pp. 18–23, 2017.

[29] B. Akhgar, "Osint as an integral part of the national security apparatus," *Open Source Intelligence Investigation*, Springer, Cham, Switzerland, pp. 3–9, 2016.

[30] I. Vacas, I. Medeiros, and N. Neves, "Detecting network threats using OSINT knowledge-based IDS," in *Proceedings of the 2018 14th European Dependable Computing Conference (EDCC)*, pp. 128–135, Lasi, Romania, 2018.

[31] A. Singh and K. Chatterjee, "Cloud security issues and challenges: a survey," *Journal of Network and Computer Applications*, vol. 79, pp. 88–115, 2017.

[32] R. Kumar and R. Goyal, "On cloud security requirements, threats, vulnerabilities and countermeasures: a survey," *Computer Science Review*, vol. 33, pp. 1–48, 2019.

[33] I. Bhardwaj, A. Kumar, and M. Bansal, "A review on lightweight cryptography algorithms for data security and authentication in IoTs," in *Proceedings of the 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*, pp. 504–509, Solan, India, September 2017.

[34] S. Sallam and B. D. Beheshti, "A survey on lightweight cryptographic algorithms," in *Proceedings of the TENCON 2018-2018 IEEE Region 10 Conference*, pp. 1784–1789, Jeju, Korea, October 2018.

[35] C. K. Chu and W. G. Tzeng, "Identity-based proxy re-encryption without random oracles," *International Conference on Information Security*, vol. 4779, pp. 189–202, 2017.

[36] Y. Yang, H. Zhu, H. Lu, J. Weng, Y. Zhang, and K.-K. R. Choo, "Cloud based data sharing with fine-grained proxy re-encryption," *Pervasive and Mobile Computing*, vol. 28, pp. 122–134, 2016.

[37] Z. Qin, H. Xiong, S. Wu, and J. Batamuliza, "A survey of proxy re-encryption for secure data sharing in cloud computing," *IEEE Transactions on Services Computing*, p. 1, 2016.

[38] S. Namasudra, "An improved attribute-based encryption technique towards the data security in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 3, Article ID e4364, 2019.

[39] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 89–98, New York, NY, USA, 2006.

[40] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07)*, pp. 321–334, Berkeley, CA, USA, May 2007.

[41] L. Wu, B. Chen, K.-K. R. Choo, and D. He, "Efficient and secure searchable encryption protocol for cloud-based Internet of Things," *Journal of Parallel and Distributed Computing*, vol. 111, pp. 152–161, 2018.

[42] Y. Wang, J. Wang, and X. Chen, "Secure searchable encryption: a survey," *Journal of Communications and Information Networks*, vol. 1, no. 4, pp. 52–65, 2016.

[43] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 24–43, Nice, France, 2010.

[44] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: theory and implementation," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, 2018.

[45] J. Zhang and H. Li, "Research and implementation of a data backup and recovery system for important business areas," in *Proceedings of the 2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 2, pp. 432–437, Hangzhou, China, 2017.

[46] H. J. Syed, A. Gani, R. W. Ahmad, M. K. Khan, and A. I. A. Ahmed, "Cloud monitoring: a review, taxonomy, and open research issues," *Journal of Network and Computer Applications*, vol. 98, pp. 11–26, 2017.

[47] R. Badhwar, "Introduction to cloud monitoring security controls," in *The CISO's Next Frontier*, pp. 289–296, Springer, Cham, Switzerland, 2021.

[48] S. Majumdar, T. Madi, Y. Jarraya, and M. Pourzandi, "Cloud security auditing: major approaches and existing challenges," in *Proceedings of the International Symposium on Foundations and Practice of Security*, pp. 61–77, Montreal, QC, Canada, November 2018.

[49] M. Kayaalp, "Modes of de-identification," *AMIA Annual Symposium Proceedings*, vol. 1044, 2017.

[50] H. J. Lee, S. H. Cho, J. W. Seong, S. Lee, and W. Lee, "De-identification and privacy issues on bigdata transformation," in *Proceedings of the 2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 514–519, Busan, Korea, February 2020.

[51] H. Li, F. Guo, W. Zhang, J. Wang, and J. Xing, "(a,k)-Anonymous scheme for privacy-preserving data collection in iot-based healthcare services systems," *Journal of Medical Systems*, vol. 42, no. 3, pp. 56–59, 2018.

[52] C. Su, "Big data security and privacy protection," in *Proceedings of the 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, pp. 87–89, Jishou, China, September 2019.

[53] X. Ma, J. Shao, C. Zuo, and R. Meng, "Efficient certificate-based signature and its aggregation," in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 391–408, Melbourne, VIC, Australia, December 2017.

[54] A. Buldas, D. Firsov, R. Laanoja, H. Lakk, and A. Truu, "A new approach to constructing digital signature schemes," *Advances in Information and Computer Security*, pp. 363–373, 2019.

[55] F. Rezaeibagha, Y. Mu, X. Huang, W. Yang, and K. Huang, "Fully secure lightweight certificateless signature scheme for IIoT," *IEEE Access*, vol. 7, pp. 144433–144443, 2019.

[56] W. Gao, W. G. Hatcher, and W. Yu, "A survey of blockchain: techniques, applications, and challenges," in *Proceedings of the 2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–11, Lyon, France, April 2018.

[57] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for IIoT devices," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[58] A. Tewari and B. B. Gupta, "A lightweight mutual authentication protocol based on elliptic curve cryptography for IoT devices," *International Journal of Advanced Intelligence Paradigms*, vol. 9, no. 2-3, pp. 111–121, 2017.

[59] P. K. Panda and S. Chattopadhyay, "A secure mutual authentication protocol for IoT environment," *Journal of Reliable Intelligent Environments*, vol. 6, no. 2, pp. 79–94, 2020.

[60] P. Mohit, R. Amin, A. Karati, G. P. Biswas, and M. K. Khan, "A standard mutual authentication protocol for cloud computing based health care system," *Journal of Medical Systems*, vol. 41, no. 4, p. 50, 2017.

[61] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication scheme for fog computing services," *Future Generation Computer Systems*, vol. 91, pp. 475–492, 2019.

[62] A. Abraham, F. Hörandner, O. Omolola, and S. Ramacher, "Privacy-preserving EID derivation for self-sovereign identity systems," *International Conference on Information and Communications Security*, vol. 11999, pp. 307–323, 2019.

[63] D. van Bokkem, R. Hageman, G. Koning, L. Nguyen, and N. Zarin, "Self-sovereign identity solutions: the necessity of blockchain technology," 2019, https://arxiv.org/abs/1904.12816.

[64] N. V. Kulabukhova, "Zero-knowledge proof in self-sovereign identity," *CEUR Workshop Proceedings*, vol. 2507, pp. 381–385, 2019.

[65] F. Cai, N. Zhu, J. He, P. Mu, W. Li, and Y. Yu, "Survey of access control models and technologies for cloud computing," *Cluster Computing*, vol. 22, no. 3, pp. 6111–6122, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1424078, 9 pages
https://doi.org/10.1155/2022/1424078

Research Article

Trajectory Privacy Preserving for Continuous LBSs in VANET

Zhihong Li,[1,2] Xiaoshuang Xing,[2] Jin Qian,[3] Hui Li,[3] and Gaofei Sun[2]

[1]*School of Computer Science and Technology, Soochow University, Suzhou, China*
[2]*School of Computer Science and Engineering, Changshu Institute of Technology, Suzhou, China*
[3]*School of Computer Science and Technology, Taizhou University, Taizhou, China*

Correspondence should be addressed to Xiaoshuang Xing; xing@cslg.edu.cn

Received 2 December 2021; Revised 18 January 2022; Accepted 28 January 2022; Published 15 February 2022

Academic Editor: Ruinian Li

Location-based services (LBSs) support various applications in vehicular ad hoc network (VANET). However, location/trajectory privacy becomes a serious concern for LBSs. Existing location/trajectory privacy-preserving schemes rarely take the attack model of adversaries into consideration, and the cost for achieving privacy has not been carefully studied. To deal with these problems, this study proposes a collaborative trajectory obfuscation scheme based on analyzing the attack model and designs a privacy-preserving efficiency metric that balances the achieved privacy and the cost. Through simulation, the effects of the density of vehicles using the same LBSs on the performance of our design and an existing scheme are investigated. The performance comparison results validate the effectiveness and efficiency of our scheme.

1. Introduction

Vehicular ad hoc network (VANET) has become an important framework of the intelligent transportation system (ITS) for applications such as navigation, road safety, and entertainment [1]. Location-based service (LBS) usually acts as the foundation that supports these applications. For example, a vehicle that wants to find a nearby supermarket and decide a suitable driving route should submit its identity, location, and service requests to corresponding LBS providers (LBSPs). While providing LBSs, LBSPs collect vehicles' locations. Once the LBSP is untrustworthy or attacked by malicious adversaries, vehicles' identities and locations will be disclosed. By analyzing frequently visited locations, private information of vehicle drivers, such as personal preferences, work locations, home addresses, and health conditions, can be revealed [2, 3]. To deal with these threats, location privacy preserving for LBSs has long been considered as an important research topic in VANETs.

The basic idea for location privacy preserving is to use pseudonyms instead of real identities when submitting LBS requests to eliminate the link between vehicles' identities and locations [4]. However, simple pseudonym replacement can only provide a single-point location privacy preserving [5]. When continuous LBSs are used, frequently visited location and/or trajectories of vehicles can still be revealed if locations are submitted to LBSPs periodically with the same pseudonym. To deal with this problem, pseudonym-changing schemes such as silent period [6, 7] and mix zone [8] are proposed, but their performance is unsatisfactory when facing the correlation attacks and their real-time performance is expected to be improved.

Location/trajectory obfuscation schemes have been designed to solve the aforementioned problems. In location obfuscation schemes, vehicles change their actual location coordinates within a tolerable error range and submit the changed locations to LBSPs. In this way, location privacy is preserved with the cost of decreased service quality since the LBSs are provided based on changed locations [9]. In trajectory obfuscation schemes, fake LBS requests, whose locations are obtained from collaborators, are submitted together with vehicles' actual LBS requests. In this way, adversaries will be misled and failed to trace the trajectories when proper collaborators are selected [10]. Trajectory obfuscation schemes preserve privacy without decreasing the service quality. However, most work has not taken the attack model of adversaries into consideration for designing trajectory obfuscation schemes. Intuitively, trajectory

privacy can be better preserved if we understand how the attackers trace the trajectories. Besides, the scenario when not all vehicles are using the same LBS has not been carefully studied. When vehicle v selects a vehicle not using the same LBS as the collaborator, the collaborator will not be able to cause significant bifurcation of the trajectory. Consequently, an attacker can still trace the trajectory successfully with high probability. Regarding the performance evaluation, various metrics, such as tracking successful ratio [11], location/trajectory entropy [12], and anonymity set size [13], have been designed to measure the achieved location/trajectory privacy. However, the cost for achieving such privacy-preserving performance has rarely been studied.

In this study, we tackle these challenges by making the following contributions:

(1) We propose a novel trajectory privacy-preserving scheme based on understanding the attack model of adversaries. We first analyze how the adversaries predict/trace the vehicles' trajectories using the Kalman filter. Then, vehicles using LBSs predict their future locations with the Kalman filter and select collaborators based on predicted locations. In this way, collaborators that are most capable of misleading the adversaries can be selected and the trajectory privacy-preserving performance can be ensured.

(2) A privacy-preserving efficiency metric is designed to evaluate the trajectory privacy-preserving performance and the cost for achieving such privacy performance.

(3) Unlike the simulation settings of the existing collaborative solutions, we set the density of vehicles using the same LBSs as a variable to better reproduce the real usage scenario. The effects of the density on our design and an existing scheme are investigated.

The rest of the study is organized as follows. Related works are reviewed in Section 2. Section 3 describes the considered system model. The attack model of adversaries is analyzed in Section 4, based on which a collaborative trajectory obfuscation scheme is proposed in Section 5. The performance of our design is compared with some exiting schemes in Section 6, and this study is concluded in Section 7.

2. Related Work

LBS supports a broad range of applications in VANETs. Locations are submitted to the LBS provider together with vehicle identities, which threaten users' location privacy and trajectory privacy. To deal with these threats, pseudonym-changing-based schemes and location/trajectory obfuscation-based schemes have been extensively studied [14].

In pseudonym-changing-based schemes, vehicles use pseudonyms instead of real identities when submitting LBS requests. In this way, links between locations and identities are broken, and location privacy can be preserved. Moreover, a vehicle changes pseudonyms following designed algorithms. Thus, links among locations of a vehicle at different times are broken and trajectory privacy can be preserved.

The silent period is an early proposed pseudonym-changing algorithm [6, 7]. A time period is defined as silent period, within which vehicles do not submit any LBS requests (i.e., keep silent) and after which vehicles submit LBS requests with changed pseudonyms. This method can mislead the attacker when more than one vehicles change their pseudonyms at the end of a silent period. However, due to the silence period, applications with high real-time requirements cannot be satisfied.

Mix zone is considered as a promising pseudonym-changing algorithm [8, 15] where mutually cooperative vehicles concurrently change their pseudonyms in mix zones created by themselves. The effectiveness of the mix zone depends on factors such as geometry, vehicle density, and geographic location in the road network. In addition, most mix zone schemes cannot avoid the continuous query correlation attack, thus limiting their performance in continuous LBS applications.

Due to the limitations mentioned above, pseudonym changing is usually used together with location/trajectory obfuscation. In location obfuscation, a vehicle changes the actual location coordinates within a tolerable error range. Then, the changed location is submitted to the LBSP. A method called CoPrivacy is proposed in [16], where vehicles form k anonymity groups through collaboration and a vehicle replaces its actual location by the regional density center of the anonymous group it belongs to. Reference [17] explores the minimum amounts of obfuscation and anonymization to block attacks on user's location privacy using an information-theoretic approach with the Markov chains. However, location obfuscation preserves location privacy with the cost of decreased service quality since LBSP provides service based on the changed locations [9].

In 2016, [18] proposed a trajectory obfuscation scheme called mutual obfuscating path (MOP) to preserve privacy without decreasing the service quality. For each vehicle v, it selects a collaborator from vehicles that are currently within its communication range. Vehicles whose trajectories are predicted to converge on v's trajectory within a predefined time threshold form the candidate collaborator set and within which the vehicle being nearest to v is selected as the collaborator. Then, v will send two LBS requests with two different pseudonyms to the LBSP. The locations of these two requests will be v's actual location and the collaborator's predicted location, respectively. From the attacker's perspective, the trajectories continue to bifurcate over time, which impedes the attacker from successful trajectory tracing. Reference [10] pointed out that there may be nefarious vehicles in the internal cooperation of MOP. To deal with this problem, it proposed a non-collaborative approach. A vehicle independently decides whether to exploit the fake location of the surrounding vehicles based on the proposed algorithm.

Despite the real time and guaranteed service quality of these two schemes, there are still open challenges to be

solved. First, the attack model has not been investigated in this work. Intuitively, trajectory privacy can be better preserved if we understand how the attackers trace the trajectories. Therefore, in this work, we will design a trajectory obfuscation scheme based on analyzing the attack model. Second, this work does not consider the scenario when not all vehicles are using the same LBS. When vehicle v selects a vehicle not using the same LBS as the collaborator, the collaborator will not be able to cause significant bifurcation of the trajectory. Consequently, an attacker can still trace the trajectory successfully with high probability. Therefore, we will take the scenario when not all vehicles are using the same LBS into consideration and the effects of the density of vehicles using the same LBS on the trajectory obfuscation performance will be investigated.

3. System Model

For easier following, we summarize the notations introduced throughout the next three sections in Table 1.

In the considered system model, there are n vehicles, denoted by $\mathbf{V} = \{v_1, v_2, \ldots, v_n\}$ on the road, and m LBS providers (LBSPs), denoted by $\mathbf{P} = \{p_1, p_2, \ldots, p_m\}$, providing m different LBSs. Each vehicle can access to $p_i \in \mathbf{P}$ for LBS or do not use LBS depending on its demand. Each vehicle is equipped with an onboard unit (OBU). Through the OBU, a vehicle can obtain real-time information about the surrounding environment and communicate with other vehicles or infrastructures in VANET. When a vehicle wants to use the ith $(1 \le i \le m)$ LBS, it sends a request to p_i via Internet service provider (ISP). ISP is a communication agency between OBU and LBSP via which the LBS requests and responses are sent. The considered LBS model is shown in Figure 1.

In our work, we mainly focus on the continuous LBS scenario, where a vehicle periodically sends the LBS request to LBSP during the service time. A topical continuous LBS is navigation. Assume a vehicle v_k uses the ith LBS from time t_0 to time t_{end}. The LBS request sent by v_k at time $t_0 \le t_j \le t_{end}$ can be denoted by $\left\{I_k, Loc_k^j, R_k^i, t_j\right\}$. Here, I_k denotes the identity of v_k, $Loc_k^j = (Lo_k^j, La_k^j)^T$ indicates v_k's location at t_j with Lo_k^j and La_k^j being the longitude and the latitude, R_k^i represents v_k's service request to LBSP p_i, and t_j is the timestamp. Let $T_{int} = t_{j+1} - t_j$ denote the time interval between two continuous LBS requests. Without loss of generality, we let T_{int} be an unit time in this study. To resist the identity-link attacks, vehicles usually use time-changing pseudonyms instead of real identity for LBS request. Therefore, the LBS request can be revised as $\left\{\widetilde{I}_k^j, Loc_k^j, R_k^i, t_j\right\}$ with $\widetilde{I}_k^j$ being the pseudonym of v_k at t_j.

Let $\mathbf{v}_i \in \mathbf{V}$ denote the subset of vehicles using the ith LBS from t_0 to t_{end}. By the end of t_{end}, p_i collects information regarding the pseudonyms and locations of these vehicles at different time points as $CI = \left\{\left\{\widetilde{I}_k^0, Loc_k^0, t_0\right\}, \ldots, \left\{\widetilde{I}_k^{end}, Loc_k^{end}, t_{end}\right\}\right\}$, $v_k \in \mathbf{v}_i$. If p_i is malicious or it is attacked by malicious adversaries, the possible trajectories of vehicles in $\mathbf{v}_i$ can be predicted using the Kalman filter or other methods. Thus, the private trajectory information will be disclosed. To tackle this problem and preserve the trajectory privacy, a collaborative obfuscation method will be designed based on the understanding of the adversary's attack model.

4. Adversary Model When the Kalman Filter Is Used for Trajectory Prediction

To effectively protect the trajectory information from being disclosed, we should understand how the adversaries predict the trajectories. Therefore, we will analyze the attack model of adversaries when the Kalman filter is used for trajectory prediction. Based on CI collected at the LBSP p_i, the adversary forms a state vector $\mathbf{x}^j = (Lo^j, La^j, Vo^j, Va^j)^T$ to denote the state of a LBS request with timestamp t_j. Here, Lo^j and La^j denote the longitude and the latitude obtained from the LBS request, and Vo^j and Va^j denote the velocity in the longitude direction and the latitude direction. Vo^0 and Va^0 are set to be 0. Vo^j and Va^j $(j \ne 0)$ can be calculated as $Vo^j = (1/T_{int})Do^j$ and $Va^j = (1/T_{int})Da^j)$ with Do^j indicating the distance between locations (Lo^j, La^j) and (Lo^{j-1}, La^j) in the longitude direction and Da^j indicating the distance between locations (Lo^j, La^j) and (Lo^{j-1}, La^j) in the latitude direction.

For t_0, the adversary will form $|\mathbf{v}_i|$ state vectors $\mathbf{x}_1^j$, $\mathbf{x}_2^j, \ldots, \mathbf{x}_{|\mathbf{v}_i|}^j$ to denote the initial states of the $|\mathbf{v}_i|$ vehicles that are using p_i's service. For t_j $(j \ne 0)$, the adversary may not be able to identify the state of $v_k \in \mathbf{v}_i$ due to pseudonym changing. Let us take an example for better understanding. Let $\{\{110011, (41.40, 2.17), t_0\}, \{110000, (40.40, 2.31), t_0\}, \{110011, (42.40, 2.17), t_1\}, \{110000, (40.87, 2.40), t_1\}, \{100110, (42.98, 2.39), t_2\}, \{001100, (41.05, 2.37), t_2\}\}$ denote the CI collected by p_i from t_0 to t_2. Two state vectors $\mathbf{x}_1^0 = (41.40, 2.17, 0, 0)^T$ and $\mathbf{x}_2^0 = (40.40, 2.31, 0, 0)^T$ will be formed at t_0. Since the pseudonyms are not changed from t_0 to t_1, the adversary easily identifies $\mathbf{x}_1^1$ and $\mathbf{x}_2^2$ from CI as $\mathbf{x}_1^1 = (42.40, 2.17, Vo_1^1, Va_1^1)^T$ and $\mathbf{x}_2^1 = (40.87, 2.40, Vo_2^1, Va_2^1)^T$. At t_3, the pseudonyms of vehicles are changed. The adversary cannot identify directly from CI whether location $(42.98, 2.39)$ or location $(41.05, 2.37)$ belongs to $\mathbf{x}_1^3$. In this case, the Kalman filter will be applied by the adversary to make a prediction and decide which location belongs to $\mathbf{x}_1^3$.

Vehicle $v_k \in \mathbf{v}_i$ is taken as an example, and we will describe how the adversary predicts v_k's trajectory. Let $\overline{\mathbf{x}}_k^j$ denote the state of v_k at t_j $(j \ne 0)$ estimated using motion model, and it can be given as follows:

$$\overline{\mathbf{x}}_k^j = \mathbf{A}\mathbf{x}_k^{j-1} + \mathbf{B}\mathbf{u}_k^{j-1} + \mathbf{w}, \tag{1}$$

where $\mathbf{x}_k^{j-1}$ is considered by the adversary to be v_k's state at t_{j-1} with probability $p(k, j-1)$ and $p(k, 0)$ is set to be 1. The calculation of $p(k, j-1)$ will be given later in this section. $\mathbf{A}$ is called as the state transition matrix, and $\mathbf{B}$ is called as the input matrix. They are defined as follows:

Table 1: Summary of notations.

Symbol	Meaning
n, m	Number of vehicles on the road, number of LBSPs
$v_k, \mathbf{V}$	The kth vehicle $(1 \le k \le n)$, set of vehicles on the road
$p_i, \mathbf{P}$	The ith LBSP $(1 \le i \le m)$, set of LBSPs
t_j	The jth timestamp of v_k using $p_i\,(0 \le j \le end)$
$I_k, \tilde{I}_k^j$	Identity of v_k, pseudonym of v_k at t_j
Lo_k^j, La_k^j, Loc_k^j	v_k's longitude at t_j, v_k's latitude at t_j, v_k's location at t_j
R_k^i	v_k's service request to p_i
T_{int}	The time interval between two continuous LBS requests
$\mathbf{v}_i$	Subset of vehicles using the ith LBS $(1 \le i \le m)$
CI	Set of v_i's information collected by p_i
$\mathbf{x}^j$	State vector formed by the adversary at t_j
$\mathbf{x}_{\lvert \mathbf{v}_i \rvert}^j$	Initial state of the $\lvert \mathbf{v}_i \rvert$th vehicle that is using p_i's service at t_j
Vo^j, Va^j	Velocity in the longitude direction and the latitude direction at t_j
Do^j	Distance between locations (Lo^j, La^j) and (Lo^{j-1}, La^j) in the longitude direction
Da^j	Distance between locations (Lo^j, La^j) and (Lo^{j-1}, La^j) in the latitude direction
$\overline{\mathbf{x}}_k^j, \mathbf{x}_k^j$	Estimated state of v_k at t_j $(j \ne 0)$, v_k's state considered by the adversary at t_j
$\mathbf{u}_k^j$	v_k's acceleration vector at t_j
$\mathbf{A}, \mathbf{B}, \mathbf{H}$	State transition matrix, input matrix, measurement matrix
Ao_k^j, Aa_k^j	v_k's acceleration in the longitude direction and the latitude direction at t_j
$\mathbf{w}, \mathbf{v}$	Disturbance noise, measurement noise
$\mathbf{Q}, \mathbf{R}$	Covariance of $\mathbf{w}$, covariance of $\mathbf{v}$
$\mathbf{E}_n$	Unit matrix with n rows and n columns
$\mathbf{P}_j, \overline{\mathbf{P}}_j$	Error covariance at t_j, error covariance calculated according to $\mathbf{P}_{j-1}$ at t_j
$\overline{Lo}_k^j, \overline{La}_k^j$	Estimated longitude and latitude of v_k at t_j
$\overline{Vo}_k^j, \overline{Va}_k^j$	Estimated velocity in the longitude direction and the latitude direction of v_k at t_j
G, C	Distance threshold, set of vehicles meeting the condition G
$p(k', j)$	Probability that $x_k'^j$ is considered by the adversary to be $v_{k'}$'s state at t_j
$\mathbf{Z}_{k'}^j$	Measurement value consisting of $Lo_{k'}^j, La_{k'}^j$
$\mathbf{K}$	Kalman gain
s_k^j, S_k^j	v_k's recorded information regarding coordinate, velocity, and acceleration at t_j, set of s_k^j
V_k^j	Set of v_k's neighbors at time t_j
y	Serial number of the minimum distance between $\overline{Loc}_k^{j+1}$ and
v_{ky}	Collaborator selected by v_k

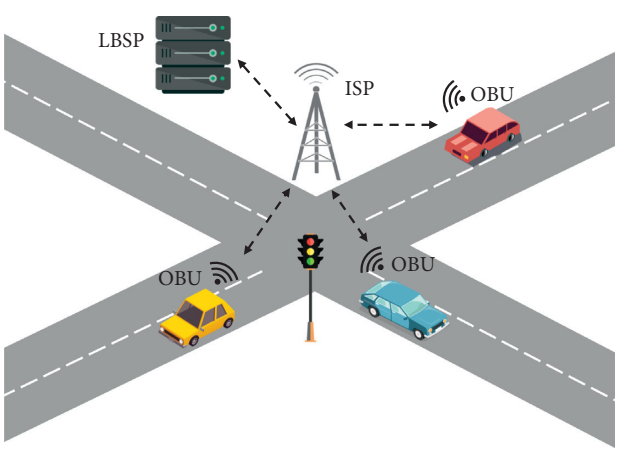


Figure 1: LBS system model in VANET.

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \frac{1}{2} & 0 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 \end{pmatrix}^{\mathrm{T}}. \tag{2}$$

$\mathbf{u}_k^{j-1}$ is called as the acceleration vector and is defined as $\mathbf{u}_k^{j-1} = (Ao_k^{j-1}, Aa_k^{j-1})^T$ with Ao_k^{j-1} and Aa_k^{j-1} denoting v_k's acceleration in the longitude direction and the latitude direction at time t_{j-1}. Note that Ao_k^0 and Aa_k^0 are set to be 0. Therefore, $\mathbf{A}\mathbf{x}_k^{j-1} + \mathbf{B}\mathbf{u}_k^{j-1}$ can be used to estimate the location and velocity of v_k at time t_j based on the location, velocity, and acceleration at t_{j-1}. Considering the error caused by imprecise estimation, a disturbance noise $\mathbf{w}$ is added in (1). Assume that $\mathbf{w}$ is a Gaussian white noise, and the covariance $\mathbf{Q}$ can be given as follows:

$$\mathbf{Q} = q\mathbf{E}_4. \tag{3}$$

Here, $\mathbf{E}_n$ denotes an unit matrix with n rows and n columns. The error covariance $\overline{\mathbf{P}}$ caused by $\mathbf{w}$ can be calculated as follows:

$$\overline{\mathbf{P}}_j = \mathbf{A}\mathbf{P}_{j-1}\mathbf{A}^T + \mathbf{Q}, \tag{4}$$

where $\mathbf{P}_{j-1}$ is the error covariance at time t_{j-1} and $\mathbf{P}_0$ is set to be $\mathbf{E}_4$.

Getting $\overline{\mathbf{x}}_k^j = (\overline{Lo}_k^j, \overline{La}_k^j, \overline{Vo}_k^j, \overline{Va}_k^j)^T$, the adversary will check CI and find a set of vehicles:

$$C = \{v_{k'} | Dis(k') < G\}. \tag{5}$$

Here, $Dis(k')$ indicates the distance between locations $(\overline{Lo}_k^j, \overline{La}_k^j)$ and $(Lo_{k'}^j, La_{k'}^j)$, G is a distance threshold set based on the longest distance traveled by the vehicle in one T_{int} under the speed limit on the road. Here, we set G to a fixed value 15. Then, vehicle $v_{k'}$ will be considered to be vehicle v_k by the adversary at time t_j with probability $p(k', j)$, and $p(k', j)$ can be calculated as follows:

$$p(k', j) = \frac{\sum_{v_{k''} \in \{C/v_{k'}\}} Dis(k'')}{\sum_{v_{k''} \in C} Dis(k'') \times \frac{p(k, j-1)}{|C| - 1}}. \tag{6}$$

Here, $p(k, j-1)$ is the probability that x_k^{j-1} is considered by the adversary to be v_k's state at t_{j-1}. When vehicle $v_{k'}$ is considered by the adversary to be vehicle v_k at time t_j, $\mathbf{Z}_{k'}^j = (Lo_{k'}^j, La_{k'}^j)^T$ will be taken as v_k's location at time t_j. Considering the errors caused by imprecise positioning, the relationship between $\mathbf{Z}_{k'}^j$ and the actual location $\mathbf{H}\mathbf{x}_{k'}^j$ can be described as follows:

$$\mathbf{Z}_{k'}^j = \mathbf{H}\mathbf{x}_{k'}^j + \mathbf{v}. \tag{7}$$

Here, $\mathbf{H}$ is a measurement matrix:

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \tag{8}$$

where $\mathbf{v}$ is the measurement noise caused by imprecise positioning. Assume that $\mathbf{v}$ in (7) is a Gaussian white noise, and the covariances $\mathbf{R}$ can be given as follows:

$$\mathbf{R} = r\mathbf{E}_2, \tag{9}$$

The parameters of the Kalman filter are updated and v_k's state at time t_j is predicted using the following formula:

$$\begin{aligned} \mathbf{K} &= \overline{\mathbf{P}}_j\mathbf{H}^T\left(\mathbf{H}\overline{\mathbf{P}}_j\mathbf{H}^T + \mathbf{R}\right)^{-1}, \\ \mathbf{x}_k^j &= \overline{\mathbf{x}}_k^j + \mathbf{K}\left(\mathbf{Z}_{k'}^j - \mathbf{H}\overline{\mathbf{x}}_k^j\right), \\ \mathbf{P}_j &= (\mathbf{E}_4 - \mathbf{K}\mathbf{H})\overline{\mathbf{P}}_j, \end{aligned} \tag{10}$$

where $\mathbf{K}$ is called as the Kalman gain. It can be found from (10) that $\mathbf{K}$ decreases with $\mathbf{R}$. That is, smaller $\mathbf{R}$ will lead to more trustable estimation using the Kalman filter. Given today's increasingly precise positioning systems, we assume that there is almost no deviation between $\mathbf{Z}_{k'}^j$ and $\mathbf{x}_{k'}^j$. Thus, r is set to be a small value $r = 0.05$. As for the setting of q, we refer to the conclusion in [5] and $q = 0.05$ is set. It should be noted that $\mathbf{x}_k^j$ obtained using (10) denotes v_k's state at time t_j predicted by the adversary when $v_{k'}$ is considered to be v_k. $\mathbf{x}_k^j$ will be substituted into (1) to estimate $\overline{\mathbf{x}}_k^{j+1}$ and correspondingly $p(k, j-1)$ in (6) will be replaced by $p(k', j)$.

5. Collaborative Obfuscation for Trajectory Privacy Preserving

Understanding how the adversary traces the vehicles' trajectories, we will design a collaborative trajectory obfuscation algorithm in this section.

Our main idea is to find collaborators for each vehicle v_k, and the collaborators will help v_k by sending the same LBS request with their pseudonyms and locations. When proper collaborators are selected, the adversary will mistake the collaborators for v_k and thus be misled during trajectory tracing. It should be noted that it is meaningless if a collaborator is driving on the same road with v_k and the adversary is successfully misled by the collaborator since v_k is still predicted to be on that road and the adversary can still get v_k's trajectory. Therefore, this work only selects collaborators at intersections and vehicles that are most capable of misleading the adversary will be selected as collaborators.

While driving, a vehicle v_k keeps recording its locations, velocities, and accelerations for the current time and N most recent historical time points. Assume the current time point is t_j, and the recorded information can be denoted as $S_k^j = \{s_k^j, s_k^{j-1}, \ldots, s_k^{j-N}\}$ with $s_k^j = (Lo_k^j, La_k^j, Vo_k^j, Va_k^j, Ao_k^j, Aa_k^j)$. Here, Lo/La, Vo/Va, and Ao/Aa denote the coordinate, velocity, and acceleration in the longitude/latitude direction, subscript k denotes vehicle v_k, and superscript j denotes time t_j. With the recorded information, v_k will predict its

location in the future time t_{j+1}, denoted by $\overline{Loc}_k^{j+1}$, using the Kalman filter. The prediction process is given in Algorithm 1.

When arriving at an intersection, vehicles predict their future locations at t_{j+1} according to Algorithm 1. Each vehicle adds its predicted location to the beacon message and broadcasts the message to neighbors, which are vehicles within its communication range. In our work, we assume that all vehicles have the same radius of communication range. Therefore, v_k is a neighbor of v_x if vehicle v_x is a neighbor of v_k. For vehicle v_k, let $V_k^j = \{v_{k1}, v_{k2}, \ldots, v_{kY}\}$ denote the set of its neighbors at time t_j. v_k will select $v_{ky} \in V_k^j$ as the collaborator if:

$$y = \arg\min_{1 \le y \le Y} Dis\left(\overline{Loc}_k^{j+1}, \overline{Loc}_{ky}^{j+1}\right), \tag{11}$$

and

$$Dis\left(\overline{Loc}_k^{j+1}, \overline{Loc}_{ky}^{j+1}\right) \le G. \tag{12}$$

Here, G is a distance threshold as described in Section 4. After selecting v_{ky} as the collaborator, v_k sends to v_{ky} a cooperative awareness message containing the LBSP v_k is connected to and the kind of LBS request it is using. Then, v_{ky} will help v_k send fake LBS requests with v_{ky}'s pseudonyms and locations to the LBSP. We assume that all vehicles are willing to collaborate since a selfish vehicle who does not collaborate will threaten its own privacy [18]. Therefore, motivation schemes will not be considered in our work.

As we have mentioned, the scenario where not all vehicles are using the same LBS should be considered when designing algorithms to preserve trajectory privacy. Under this scenario, if the selected collaborator v_{ky} is not using the same LBS as v_k and v_{ky} just sends one fake request to the LBSP, it will be easy for the adversary to identify the fake request. To deal with this problem, the collaborator should keep sending fake LBS requests for a period of time.

In our design, v_{ky}, which is selected as the collaborator at an intersection, is required to keep sending fake LBS requests until arriving at the next intersection. Once the attacker mistakenly tracks v_{ky}, its predicted trajectory may create more divergences at the next intersection. In this way, trajectory privacy of vehicles will be better preserved. The distance between two intersections is generally more than 600 meters in the main urban roads, and the distance between the two gateways of the expressway is even further [19]. According to [20], the average speed of vehicles on weekday in first-tier cities is about $24\,(km/h)$, while it can reach $30 - 45\,(km/h)$ in other major cities. Therefore, we assume that the average speed of vehicles is $10\,(m/s)$ and the time required for a collaborator to keep sending fake requests is $60s$.

6. Performance Evaluation

In this section, we conduct traffic simulation to evaluate the performance of the proposed trajectory privacy-preserving scheme. 219 vehicles drive on an 6 km × 6 km map of Suzhou, China, as shown in Figure 2, using SUMO [21, 22].

Input: $S_k^j = \left\{s_k^j, s_k^{j-1}, \ldots, s_k^{j-N}\right\}$, $\mathbf{P}_{j-N} = \mathbf{E}_4$
Output: $\overline{\mathbf{Loc}}_k^{j+1}$
(1) $\mathbf{x}_k^{j-N} = (Lo_k^{j-N}, La_k^{j-N}, Vo_k^{j-N}, Va_k^{j-N})^T$
(2) **for** $l \leftarrow j - N$ to $j - 1$ **do**
(3) $\mathbf{u}_k^l = (Ao_k^j, Aa_k^j)^T$
(4) $\overline{\mathbf{x}}_k^{l+1} = \mathbf{A}\mathbf{x}_k^l + \mathbf{B}\mathbf{u}_k^l$
(5) $\mathbf{Z}_k^{l+1} = (Lo_k^{l+1}, La_k^{l+1})^T$
(6) $\overline{\mathbf{P}}_{l+1} = \mathbf{A}\mathbf{P}_l\mathbf{A}^T + \mathbf{Q}$
(7) $\mathbf{K} = \overline{\mathbf{P}}_{l+1}\mathbf{H}^T(\mathbf{H}\overline{\mathbf{P}}_{l+1}\mathbf{H}^T + \mathbf{R})^{-1}$
(8) $\mathbf{x}_k^{l+1} = \overline{\mathbf{x}}_{l+1} + \mathbf{K}(\mathbf{Z}_{l+1} - \mathbf{H}\overline{\mathbf{x}}_{l+1})$
(9) $\mathbf{P}_{l+1} = (\mathbf{E}_4 - \mathbf{K}\mathbf{H})\overline{\mathbf{P}}_{l+1}$
(10) **end for**
(11) $\overline{\mathbf{Loc}}_k^{j+1} = \mathbf{H}(\mathbf{A}\mathbf{x}_k^j + \mathbf{B}\mathbf{u}_k^j)$

Algorithm 1: Location prediction process of vehicle v_k.

We consider that a LBSP collects the received LBS requests for 5 minutes. Within these 5 minutes, vehicles using this LBS move randomly on the map following traffic rules. The LBS requests are sent by the vehicles every T_{int}. To investigate the effects of the density of vehicles using the LBS on the performance of our design, four situations, where 25%, 50%, 75%, and 100% vehicles on the road are using the LBS, are set.

As described in Section 5, a vehicle v_k records its locations, velocities, and accelerations of N most recent historical time points. This information of N time points will be used to predict v_k's location at the future time and the collaborator will be selected based on the predicted location. A proper set of N will help v_k make an accurate prediction, thus selecting a capable collaborator, with small memory and computation cost. To set a proper value of N, we try to make location predictions using the moving information of the 219 vehicles. As shown in Figure 3, prediction deviation, the average distance difference between predicted locations and real locations, changes with N. Generally, smaller deviation comes with greater N. This is intuitive since more information leads to more precise prediction. However, fluctuation occurs in Figure 3. This is because that vehicles cannot stay on the same moving pattern on different points of the road. For example, the velocity tends to be stable or change slightly between intersections, while the velocity tends to decrease and the acceleration tends to be stable or change slightly when a vehicle approaches the intersections. These complex road conditions cause fluctuations in the results of our predictions of vehicles' full trajectories. Therefore, if N is not set properly, a vehicle will use the moving pattern between intersections to predict the moving pattern approaching intersections leading to imprecise predictions and increased deviations. According to the results shown in Figure 3, N is set to be 5 in our simulation.

6.1. Performance Metrics. Two metrics are designed to evaluate the trajectory privacy-preserving performance. First, tracking success ratio is designed to represent the possibility that the actual trajectory is v_k's trajectory in the

Figure 2: Map used in simulation.

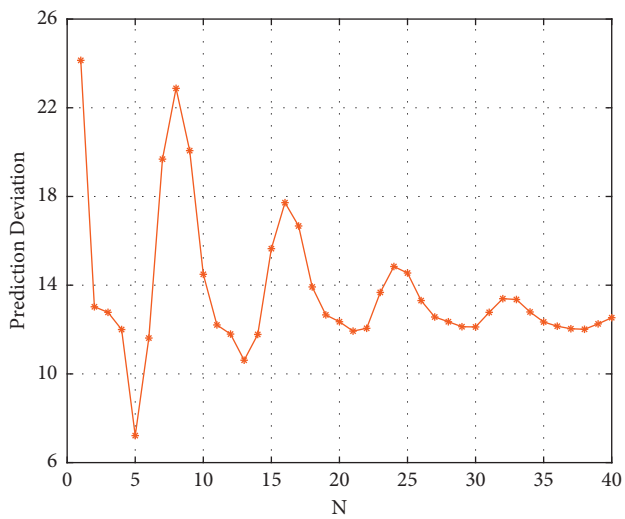


Figure 3: Prediction deviation with different N values.

adversary's eyes. For a time t, the tracking success ratio is defined as follows:

$$SR_t = \prod_{j=0}^{t} p(k, j), \tag{13}$$

where k satisfies that location (Lo_k^j, La_k^j) is on the trajectory.

Then, we propose a metric to investigate the trajectory privacy-preserving efficiency. For collaborative obfuscation, collaborators are required to send fake LBS requests. Therefore, the number of collaborators selected by v_k is considered as the cost to preserve v_k's trajectory privacy. Let NCV_t denote the number of selected collaborators until time t, and the privacy-preserving efficiency E_t is defined as follows:

$$E_t = \frac{1 - SR_t}{NCV_t}. \tag{14}$$

6.2. Performance Analysis and Comparison. We first investigate the average tracking success ratio of each density. As can be seen from Figure 4, tracking success ratio of

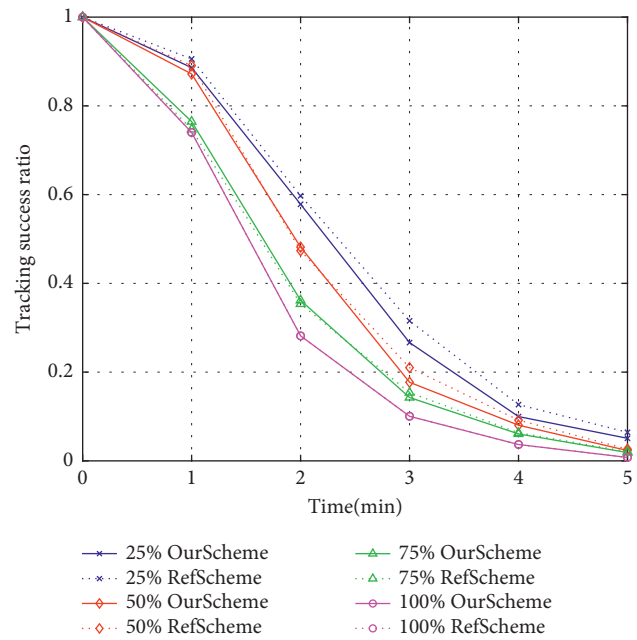


Figure 4: Tracking success ratio *v.s.* The density of vehicles using the LBS.

each density decreases gradually and reduces to almost 0 over time. The tracking success ratio of low density is obviously higher than that of high density. This result confirms the influence of the number of obfuscation locations on privacy preserving. More vehicles using the same LBS will lead to more LBS requests with similar locations submitted at a specific time. These requests can help each other mislead the adversary. That is, the actual LBS requests from other vehicles play the role of fake requests for v_k.

The tracking success ratio of our design is also compared with that of the scheme proposed in [10], referred to RefScheme in the rest of this section. It can be seen from Figure 4 that our scheme holds similar tracking success ratio with RefScheme and our scheme performances slightly worse than RefScheme before 2 minutes under 50% and 75% densities. It is because that vehicles generate fake locations all the time on their way in RefScheme, while our scheme only works at intersections. Less collaborators are selected, and less fake LBS requests are sent leading to a slightly worse performance with less communication cost for collaborator selection and request sending. However, our design outperforms RefScheme when the density of vehicles using the LBS is low. This is because that more confusing trajectories are generated by our scheme. When the density reaches 100%, because the vehicles on the road are using the same LBS and the location updates received by LBSP under these two schemes are the same, the tracking success ratio of the two schemes is the same.

Figure 5 shows the efficiency defined in (14) of two schemes. We can see that the efficiency of both is 0 at first since $SR_0 = 1$. Clearly, our scheme outperforms RefScheme since a similar tracking success ratio can be achieved by our

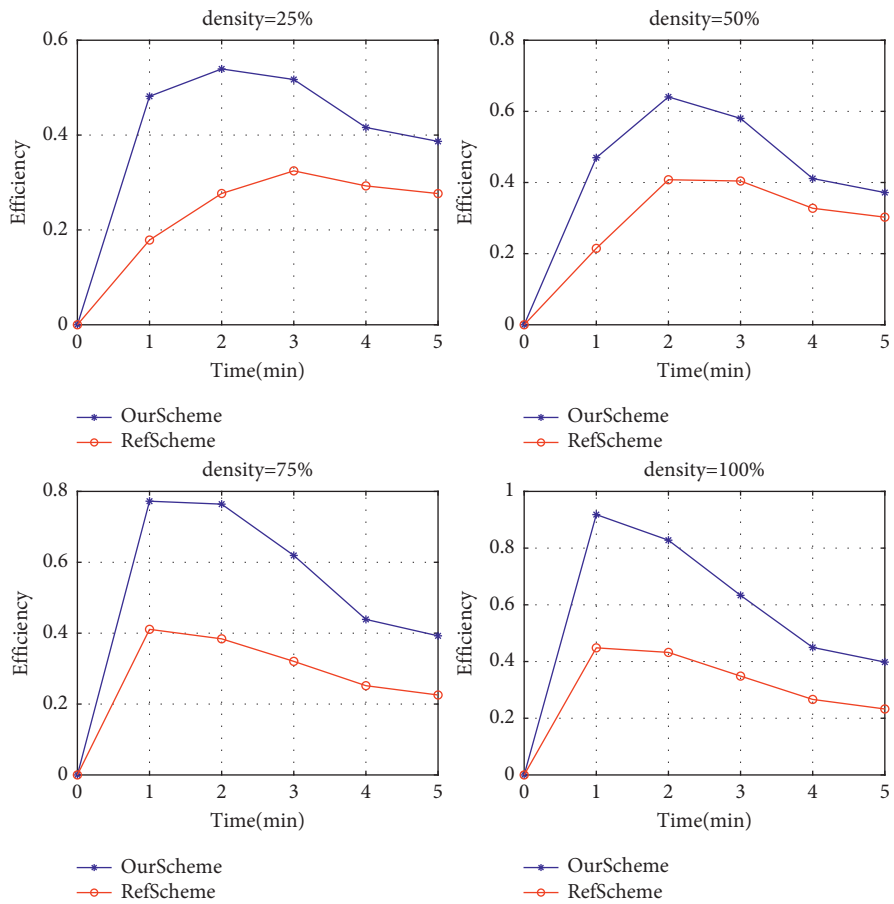


Figure 5: Privacy-preserving efficiency for different densities.

Table 2: Qualitative performance comparison.

Scheme	Service quality	Privacy performance
The proposed scheme	High	High
Silent period [7]	Low	High
Mix zone [15]	High	Low
Location obfuscation [17]	Low	High

scheme with less collaborators. It should be noted that all curves in Figure 5 are not monotonous. This is because that *NCV* does not increase monotonically due to the randomness of vehicle movement.

Moreover, the proposed scheme can be qualitatively compared with other existing methods, such as silent period [7], mix zone [15], and location obfuscation [17] from the perspectives of privacy performance and service quality. The comparison result is shown in Table 2. References [7, 17] provide good privacy protection with compromise on service quality since no service request is sent during the silent period and service is provided based on deviated locations, respectively. The service quality of scheme [15] can be guaranteed, but it is vulnerable to continuous tracking attacks. Our scheme overcomes these drawbacks since fake requests from collaborators protect trajectory privacy, while actual requests ensure service quality.

7. Conclusion

This study proposes a collaborative trajectory obfuscation scheme based on analyzing the attack model of adversaries. Compared with exiting works, our design has high service quality and high privacy strength and can preserve the trajectory privacy with less cost, that is, fewer collaborators. To better reproduce the real usage scenario where not all vehicles are using the same LBS, we introduce the density of vehicles using the same LBS as a variable in our simulation. The results show that low density will increase the risk of trajectory exposure, and therefore, density should be considered when designing trajectory privacy-preserving schemes.

Data Availability

The data that support the findings of this study are derived from the following resources available in the public domain: https://www.openstreetmap.org/and http://sourceforge.net/projects/sumo/.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This work was supported by the Natural Science Foundation of Jiangsu Province (BK20211357), Future Network Scientific Research Fund Project (FNSRFP-2021-YB-40), and Natural Science Foundation of China (grant no. 61802274).

References

[1] J. Liang and W. Wang, "Security and privacy in vehicular ad hoc network and vehicle cloud computing: a survey," *Wireless Communications and Mobile Computing*, vol. 2020, p. 2020.

[2] A. Ullah, X. Yao, S. Shaheen, and H. Ning, "Advances in position based routing towards its enabled fog-oriented vanet: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 828–840, 2019.

[3] S. Khan, I. Sharma, M. Aslam, M. Z. Khan, and S. Khan, "Security challenges of location privacy in VANETs and state-of-the-art solutions: a survey," *Future Internet*, vol. 13, no. 4, p. 96, 2021.

[4] T. Gao and L. Zhao, "Pseudonym schemes based on location privacy protection in vanets: a survey," *Innovative Mobile and Internet Services in Ubiquitous Computing*, vol. 55, pp. 597–605, 2020.

[5] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos, "Privacy in inter-vehicular networks: why simple pseudonym change is not enough," in *Proceedings of the Seventh International Conference on Wireless On-Demand Network Systems and Services (WONS)*, pp. 176–183, Kranjska Gora, Slovenia, February 2010.

[6] S. Krishna, L. Huang, and M. Li, *Caravan: Providing Location Privacy for Vanet*, Washington University Seattle Department of Electrical Engineering, Washington, DC, USA, 2005.

[7] S. Krishna, M. Li, L. Huang, and R. Poovendran, "Amoeba: robust location privacy scheme for vanet," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1569–1589, 2007.

[8] C. Kalaiarasy, N. Sreenath, and A. Amuthan, "Location privacy preservation in vanet using mix zones: a survey," in *Proceedings of the International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–5, Tamilnadu, India, December 2019.

[9] X. Zhang, X. Gui, and Z. Wu, "Survey of privacy protection research for location services," *Journal of Software*, vol. 09, pp. 223–245, 2015.

[10] J. Cui, J. Wen, S. Han, and H. Zhong, "Efficient privacy-preserving scheme for real-time location data in vehicular ad-hoc network," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3491–3498, 2018.

[11] I. Wagner and D. Eckhoff, "Technical privacy metrics: a systematic survey," *ACM Computing Surveys*, vol. 51, no. 3, pp. 1–38, 2018.

[12] H. To, K. Nguyen, and C. Shahabi, "Differentially private publication of location entropy," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 1–10, Redondo Beach, CA, USA, November 2016.

[13] G. P. Corser, H. Fu, and A. Banihani, "Evaluating location privacy in vehicular communications and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2658–2667, 2016.

[14] T. Hassan, T. Nomani, M. Mohsin, and Saira Sattar, "A survey on location privacy techniques deployed in vehicular networks," in *Proceedings of the 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 604–613, Islamabad, Pakistan, January 2019.

[15] R. Lu, X. Lin, H. Tom, X. Liang, and X. Shen, "Pseudonym changing at social spots: an effective strategy for location privacy in vanets," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 86–96, 2011.

[16] Yi Huang, H. Zheng, and X.-F. Meng, "Coprivacy: a collaborative location privacy-preserving method without cloaking region," *Jisuanji Xuebao*, vol. 34, no. 10, pp. 1976–1985, 2011.

[17] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Limits of location privacy under anonymization and obfuscation," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pp. 764–768, Victoria, Australia, July 2017.

[18] J. Lim, H. Yu, K. Kim, M. Kim, and S.-B. Lee, "Preserving location privacy of connected vehicles with highly accurate location updates," *IEEE Communications Letters*, vol. 21, no. 3, pp. 540–543, 2016.

[19] Ministry of Housing and PRC Urban-Rural Development, *CJJ129 Specification for Design of Urban Expressway*, China Architecture and Building Press, Beijing, China, 2009.

[20] Amap, *Urban Transportation Report*", Beijing Traffic Information Center, Beijing, China, 2020.

[21] Steve Coast. "Open street map". [Online]. Available: https://www.openstreetmap.org/.

[22] Eclipse Foundation. "Simulation of urban mobility". [Online]. Available: http://sourceforge.net/projects/sumo/.