

Wireless Communications and Mobile Computing

Security and Privacy Challenges for Internet-of-Things and Fog Computing 2021

Lead Guest Editor: Ximeng Liu

Guest Editors: Lei Chen and Hui Zhu





Security and Privacy Challenges for Internet-of-Things and Fog Computing 2021

Wireless Communications and Mobile Computing

**Security and Privacy Challenges for
Internet-of-Things and Fog Computing
2021**

Lead Guest Editor: Ximeng Liu

Guest Editors: Lei Chen and Hui Zhu






Copyright © 2022 Hindawi Limited. All rights reserved.

This is a special issue published in “Wireless Communications and Mobile Computing.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Chief Editor

































Zhipeng Cai , USA

Associate Editors

Ke Guan , China
Jaime Lloret , Spain
Maode Ma , Singapore

Academic Editors

Muhammad Inam Abbasi, Malaysia
Ghufran Ahmed , Pakistan
Hamza Mohammed Ridha Al-Khafaji ,
Iraq
Abdullah Alamoodi , Malaysia
Marica Amadeo, Italy
Sandhya Aneja, USA
Mohd Dilshad Ansari, India
Eva Antonino-Daviu , Spain
Mehmet Emin Aydin, United Kingdom
Parameshchhari B. D. , India
Kalapaveen Bagadi , India
Ashish Bagwari , India
Dr. Abdul Basit , Pakistan
Alessandro Bazzi , Italy
Zdenek Becvar , Czech Republic
Nabil Benamar , Morocco
Olivier Berder, France
Petros S. Bithas, Greece
Dario Bruneo , Italy
Jun Cai, Canada
Xuesong Cai, Denmark
Gerardo Canfora , Italy
Rolando Carrasco, United Kingdom
Vicente Casares-Giner , Spain
Brijesh Chaurasia, India
Lin Chen , France
Xianfu Chen , Finland
Hui Cheng , United Kingdom
Hsin-Hung Cho, Taiwan
Ernestina Cianca , Italy
Marta Cimitile , Italy
Riccardo Colella , Italy
Mario Collotta , Italy
Massimo Condoluci , Sweden
Antonino Crivello , Italy
Antonio De Domenico , France
Floriano De Rango , Italy




Antonio De la Oliva , Spain
Margot Deruyck, Belgium
Liang Dong , USA
Praveen Kumar Donta, Austria
Zhuojun Duan, USA
Mohammed El-Hajjar , United Kingdom
Oscar Esparza , Spain
Maria Fazio , Italy
Mauro Femminella , Italy
Manuel Fernandez-Veiga , Spain
Gianluigi Ferrari , Italy
Luca Foschini , Italy
Alexandros G. Fragkiadakis , Greece
Ivan Ganchev , Bulgaria
Óscar García, Spain
Manuel García Sánchez , Spain
L. J. García Villalba , Spain
Miguel Garcia-Pineda , Spain
Piedad Garrido , Spain
Michele Girolami, Italy
Mariusz Glabowski , Poland
Carles Gomez , Spain
Antonio Guerrieri , Italy
Barbara Guidi , Italy
Rami Hamdi, Qatar
Tao Han, USA
Sherief Hashima , Egypt
Mahmoud Hassaballah , Egypt
Yejun He , China
Yixin He, China
Andrej Hrovat , Slovenia
Chunqiang Hu , China
Xuexian Hu , China
Zhenghua Huang , China
Xiaohong Jiang , Japan
Vicente Julian , Spain
Rajesh Kaluri , India
Dimitrios Katsaros, Greece
Muhammad Asghar Khan, Pakistan
Rahim Khan , Pakistan
Ahmed Khattab, Egypt
Hasan Ali Khattak, Pakistan
Mario Kolberg , United Kingdom
Meet Kumari, India
Wen-Cheng Lai , Taiwan

Jose M. Lanza-Gutierrez, Spain
Pavlos I. Lazaridis , United Kingdom
Kim-Hung Le , Vietnam
Tuan Anh Le , United Kingdom
Xianfu Lei, China
Jianfeng Li , China
Xiangxue Li , China
Yaguang Lin , China
Zhi Lin , China
Liu Liu , China
Mingqian Liu , China
Zhi Liu, Japan
Miguel López-Benítez , United Kingdom
Chuanwen Luo , China
Lu Lv, China
Basem M. ElHalawany , Egypt
Imadeldin Mahgoub , USA
Rajesh Manoharan , India
Davide Mattera , Italy
Michael McGuire , Canada
Weizhi Meng , Denmark
Klaus Moessner , United Kingdom
Simone Morosi , Italy
Amrit Mukherjee, Czech Republic
Shahid Mumtaz , Portugal
Giovanni Nardini , Italy
Tuan M. Nguyen , Vietnam
Petros Nicolaitidis , Greece
Rajendran Parthiban , Malaysia
Giovanni Pau , Italy
Matteo Petracca , Italy
Marco Picone , Italy
Daniele Pinchera , Italy
Giuseppe Piro , Italy
Javier Prieto , Spain
Umair Rafique, Finland
Maheswar Rajagopal , India
Sujan Rajbhandari , United Kingdom
Rajib Rana, Australia
Luca Reggiani , Italy
Daniel G. Reina , Spain
Bo Rong , Canada
Mangal Sain , Republic of Korea
Praneet Saurabh , India

Hans Schotten, Germany
Patrick Seeling , USA
Muhammad Shafiq , China
Zaffar Ahmed Shaikh , Pakistan
Vishal Sharma , United Kingdom
Kaize Shi , Australia
Chakchai So-In, Thailand
Enrique Stevens-Navarro , Mexico
Sangeetha Subbaraj , India
Tien-Wen Sung, Taiwan
Suhua Tang , Japan
Pan Tang , China
Pierre-Martin Tardif , Canada
Sreenath Reddy Thummaluru, India
Tran Trung Duy , Vietnam
Fan-Hsun Tseng, Taiwan
S Velliangiri , India
Quoc-Tuan Vien , United Kingdom
Enrico M. Vitucci , Italy
Shaohua Wan , China
Dawei Wang, China
Huaqun Wang , China
Pengfei Wang , China
Dapeng Wu , China
Huaming Wu , China
Ding Xu , China
YAN YAO , China
Jie Yang, USA
Long Yang , China
Qiang Ye , Canada
Changyan Yi , China
Ya-Ju Yu , Taiwan
Marat V. Yuldashev , Finland
Sherali Zeadally, USA
Hong-Hai Zhang, USA
Jiliang Zhang, China
Lei Zhang, Spain
Wence Zhang , China
Yushu Zhang, China
Kechen Zheng, China
Fuhui Zhou , USA
Meiling Zhu, United Kingdom
Zhengyu Zhu , China

Contents

Improved Cloud Auditing Protocol and Its Application for Pandemic Data Management

Xu An Wang , Zhengge Yi , Xiaoyuan Yang, Jindan Zhang, Yun Xie, Manman Zhang, and Guixin Wu 

Research Article (10 pages), Article ID 5127499, Volume 2022 (2022)

Public Key Encryption with Authorized Equality Test on Outsourced Ciphertexts for Cloud-Assisted IoT in Dual Server Model

Meng Zhao, Yong Ding , Shijie Tang, Hai Liang , and Huiyong Wang


Research Article (10 pages), Article ID 4462134, Volume 2022 (2022)

Security Guarantee for Vehicular Message Transmission Based on Dynamic Social Attributes

Lishui Chen , Jing Wang , Xing Chen , and Yifu Zhang 


Research Article (11 pages), Article ID 8302527, Volume 2021 (2021)

Decentralized Certificate Management for Network Function Virtualisation (NFV) Implementation in Telecommunication Networks

Junzhi Yan , Na Li, Bo Yang, Min Li, Li Su, and Shen He

Research Article (10 pages), Article ID 6985492, Volume 2021 (2021)

Securing Open Banking with Model-View-Controller Architecture and OWASP

Deina Kellezi, Christian Boegelund, and Weizhi Meng 



Research Article (13 pages), Article ID 8028073, Volume 2021 (2021)

Efficient Authentication for Internet of Things Devices in Information Management Systems

Xiaofeng Wu , Fangyuan Ren , Yiming Li , Zhenwei Chen , and Xiaoling Tao 

Research Article (14 pages), Article ID 9921036, Volume 2021 (2021)

Differential Privacy Location Protection Method Based on the Markov Model

Hongtao Li , Yue Wang , Feng Guo, Jie Wang, Bo Wang, and Chuankun Wu

Research Article (12 pages), Article ID 4696455, Volume 2021 (2021)

Research Article

Improved Cloud Auditing Protocol and Its Application for Pandemic Data Management

Xu An Wang ^{1,2}, **Zhengge Yi** ¹, **Xiaoyuan Yang**¹, **Jindan Zhang**³, **Yun Xie**⁴,
Manman Zhang⁴ and **Guixin Wu** ⁵

¹Engineering University of PAP, Xi'an, China

²State Key Laboratory of Public Big Data, Guizhou University, Guiyang, China

³Xianyang Vocational Technical College, Xianyang, China

⁴Nanjing University of Posts and Telecommunications, Nanjing, China

⁵Chongqing University Cancer Hospital, Chongqing, China

Correspondence should be addressed to Xu An Wang; 1261510059@qq.com and Guixin Wu; wuguixin123456@126.com

Received 11 May 2021; Revised 23 November 2021; Accepted 13 January 2022; Published 26 February 2022

Academic Editor: Ximeng Liu

Copyright © 2022 Xu An Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Data integrity verification mechanisms play an important role in cloud environments. Recently, a lightweight identity-based cloud storage audit scheme has been proposed; this paper points out security vulnerabilities of their OffTagGen algorithm. That is, the attackers such as malicious cloud servers can forge the tags, which can destroy data integrity. By improving the construction of OffTagGen algorithm, an improved security cloud auditing protocol is proposed in this work to better protect user's privacy. The analysis shows that the new protocol is effective and resistant to attacks.

1. Introduction

In the last two years, the COVID-19 pandemic has become a major disaster in the world. As COVID-19 has a certain fatality rate and spreads very fast, prevention and control of this virus have become a top priority worldwide. However, compared to the prevention and control regarding SARS in 2003, information related technique is being widely used in all aspects regarding prevention and control of this COVID-19 pandemic. Hence, extensive collection, processing, and investigation of personal data has become an important part of the anti-pandemic work. Given the huge amount of data collected, it is necessary to store these data in the cloud to reduce the storage burden.

As a new storage paradigm, cloud storage collects different storage devices to provide users with massive data storage. Hospitals and patients can easily access data by connecting to the cloud anytime, anywhere, and through any networked device whenever needed.

The infrastructures supporting cloud storage are distributed and virtual. This brings some threats to users' data security, such as network virus propagation, unauthorized access, denial of service attacks, information leakage, data loss, as well as network infrastructure that could damage data integrity during data transmission, etc. For pandemic prevention and control, it is clear that the loss of data, such as the patient's recent whereabouts, will certainly cause enormous trouble and could even lead to further virus spread and may even favour the situation of successive waves of the new coronavirus.

Due to the threat of internal or external attacks, data stored in the cloud are easily damaged. In addition, the cloud service provider (CSP) may not notify the user of this event in consideration of its own reputation. The user has a mechanism to detect data corruption only after accessing the data [1–3]. Therefore, in order to improve the reputation of cloud storage and let users know the integrity of the hosted data in a timely manner, a mechanism is needed to verify the

data integrity in the cloud. Hence, an integrity verification mechanism is very important in the cloud environment.

1.1. Related Works. Traditional methods need to download the entire data from the cloud when verifying data integrity, which brings unacceptable communication and computing costs and greatly consumes users' resources. In order to satisfy the user's remote checking of data integrity, the cloud data remote integrity check solution should need not to download the complete data in the cloud storage environment. Thus, the following solutions have been proposed.

Ateniese et al. [4] first proposed a provable data possession (PDP) scheme, an effective technology to audit the cloud storage. In the PDP protocol, data are encoded as blocks, and the user processes the block data to generate a verifiable authenticator, and then it outsources the data blocks and authenticators to the cloud. A public verifier with sufficient resources is also called a third-party auditor (TPA) and is trusted by users to check the data integrity. TPA creates a challenge to the server by randomly selecting a small group of block indexes. The server returns a proof that proves the integrity of the challenged blocks. TPA can effectively verify the proof without downloading data block. PDP has laid the foundation for the design of cloud storage audit schemes. In recent years, many researchers have conducted extensive and in-depth explorations around PDP [5–7].

In order to obtain better efficiency and performance, several improved PDP protocols have been proposed [8, 9]. The previously proposed schemes mostly use traditional public key cryptography, so a trusted certificate authority (CA) is required to issue a certificate to bind certain user identities and their public keys. Heavy certificate management, including certificate generation, distribution, and revocation, requires a lot of computing and storage resources. As the number of users increases, certificate management becomes extremely difficult. In addition, the verifier must retrieve the certificate from the CA and then check the validity of the public key certificate, which also brings heavy calculation and communication costs to the verifier. Therefore, the certificate-based PDP protocol is very inefficient when used in actual situations.

In order to overcome this problem, researchers considered applying identity-based cryptography to the PDP protocol and therefore proposed many ID-PDP solutions. Wang et al. [10] first introduced the concept of identity-based PDP (ID-PDP), which uses user names or emails instead of public keys. Then, ID-PDP is further extended to the multicloud storage environment [11] to check the integrity of remote data. In order to improve performance, Wang et al. [12] added a proxy server to the remote data integrity check scheme. The proxy server processes data instead of users. In the scheme, incentive and unconditional anonymous ID-PDP was first proposed to protect and encourage criminal whistleblowers. Yu et al. [13] used RSA signature technology to design an ID-based integrity cloud data check protocol. The protocol supports variable size file blocks and public verification. In order to further improve

security, Yu et al. [14] combined the key homomorphic encryption technology in the cryptographic cloud audit system and proposed an improved scheme with perfect data privacy protection capabilities. The ID-based privacy-preserving integrity verification of shared data over untrusted cloud scheme is proposed, which can support users to update the data in cloud and protect users' privacy in untrusted cloud servers. Li et al. [15] proposed identity-based privacy-preserving remote data integrity checking for cloud storage scheme, which uses homomorphic verifiable tags to reduce the computational complexity and uses random integer addition to mask the original data to protect the verifier from obtaining any knowledge about the data during the integrity checking process.

1.2. Contribution. Currently, using cloud storage audit protocols is regarded as an important cloud service. However, the existing audit protocols have certain shortcomings. On the one hand, most of them rely on expensive public key infrastructure (PKI), so certificate management/verification is very complicated. On the other hand, most cloud users have limited resources. Nowadays, ID-based cloud audit protocols have attracted the attention of researchers, but most of them require users with limited resources to perform expensive operations.

Recently, Rabaninejad et al. [16] proposed a lightweight identity-based provable data ownership cloud storage audit scheme, which supports privacy and traceability of user identities. They also proposed an online/offline ID-based PDP scheme [17]. However, we discovered that there are security flaws in the digital signature (OffTagGen) of their scheme. Attackers, such as malicious cloud servers, can destroy the privacy of user's identity privacy and damage data privacy and integrity. In order to get a more secure protocol, based on the scheme presented in [16], we propose an improved one and discuss its application in pandemic data management. The main contributions of this paper are summarized as follows:

- (1) We firstly point out the insecurity of Rabaninejad et al.'s lightweight identity-based provable data ownership cloud storage audit scheme. We give two attacks to show that data tags can be easily forged.
- (2) We provide an improved secure cloud audit protocol that protects user privacy. This new protocol is effective yet resistant to attacks.
- (3) Finally, we show how our scheme can be applied to the pandemic data management.

1.3. Organization. The rest of this article is organized as follows. In Section 2, we describe the system framework. In Section 3, we review the cloud audit scheme proposed by Rabaninejad et al. [17]. In Sections 4 and 5, we introduce the attack. In Section 6, we provide an improved privacy protection cloud audit protocol and conduct a rough analysis of its security. In Section 7, we apply our scheme to pandemic data management. Finally, in Section 8, we conclude the work and point out some directions for future work.

2. System Framework

In this section, we first describe the system model. Then, we give the goals of the design. After that, we introduce some necessary definitions. Finally, we show the security model.

2.1. System Model. The system model includes four entities, as shown in Figure 1, which involves the key generation center (KGC), the users, the third-party auditor (TPA), and the cloud server. The functions of each entity are summarized as follows:

- (1) *KGC.* Based on the user's identity, KGC generates its private key.
- (2) *The Users.* When the users want to store data files remotely in the cloud, they first divide the file into several blocks, use their own private key to generate a label or tag on each data block, and then outsource (block, label) to the cloud server.
- (3) *TPA.* TPA performs public audits delegated by the users.
- (4) *Cloud Server.* The cloud server stores user-managed data and generates a proof verified by TPA.

As shown in Figure 1, the workflow of the four parties can be described as follows:

- (1) The users generate the offline tags and store them locally.
- (2) The users send their identity information to KGC.
- (3) KGC uses the master key and the users' identity information to generate the users' private key and returns it to them.
- (4) When users need to store data files in the cloud server, they generate online tags by using some lightweight computations based on offline tags.
- (5) Users outsource the (block, online tag) pair to the cloud server.
- (6) The user sends an audit request to TPA with some audit information attached.
- (7) TPA sends the challenge message to the cloud server.
- (8) The cloud server generates a proof based on the challenge message and sends it to TPA.
- (9) TPA sends the results of the audit as the auditing report to the users.

2.2. Design Goal. The design goals are roughly as follows.

- (i) *Correctness.* If TPA honestly follows the agreement, it can correctly audit the integrity of outsourced data.

- (ii) *Soundness.* If an untrusted cloud server has not completely stored the outsourcing data, it cannot pass the audit.
- (iii) *Public Audit.* TPA can replace the user to remotely audit the integrity of the data.
- (iv) *Scalability.* TPA can simultaneously support the effective verification of multiple audit requirements.
- (v) *Lightweight.* TPA provides low-cost label generation algorithms for users with limited computing resources.

2.3. Definition. The ID-PDP scheme includes the following algorithms:

- (i) *Setup*(1^k) \rightarrow ($param, msk$). KGC executes this algorithm. The input is the security parameter 1^k , and the output is the master key msk and the public parameter $param$.
- (ii) *Extract*($ID, param, msk$) $\rightarrow k_{ID}$. KGC receives the inputs, including msk , and identity ID and then outputs the secret key k_{ID} .
- (iii) *TagGen*($param, k_{ID}, F$) $\rightarrow \sigma$. The data owner with the key k_{ID} executes this algorithm, inputs the parameters and the data file, and first splits F into n blocks $F = (m_1, \dots, m_n)$. Next, the data owner generates a corresponding label σ_i for each block m_i and outsources the label $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$ and the data blocks $F = (m_1, \dots, m_n)$ to the cloud together.
- (iv) *Challenge*($param, F_{name}; ID$) $\rightarrow chal$. TPA outputs a challenge on behalf of the data owner whose identity is ID to challenge the integrity of the F_{name} file. The parameters, the name F_{name} of the data file F , and the identity ID of the data owner are taken as input.
- (v) *ProofGen*($param, ID, chal, F, \sigma$) $\rightarrow proof$. The cloud server executes *ProofGen*, inputs $param, chal$, owner's identity ID , and the file F with label σ , and outputs a certificate proving the integrity of the challenge block.
- (vi) *ProofVerify*($param, F_{name}, R, ID, chal, proof$) $\rightarrow \{0, 1\}$. TPA executes *ProofVerify* to verify the proof of challenge reported by the cloud server. The input are $param, F_{name}$, the identity ID of the data owner, and the pair ($chal, proof$). If the verification is passed, 1 is output, otherwise 0.

2.4. Security Model. For maintaining their own reputation, cloud servers are generally unwilling to disclose data loss/damage to the verifier, so they are not completely credible in

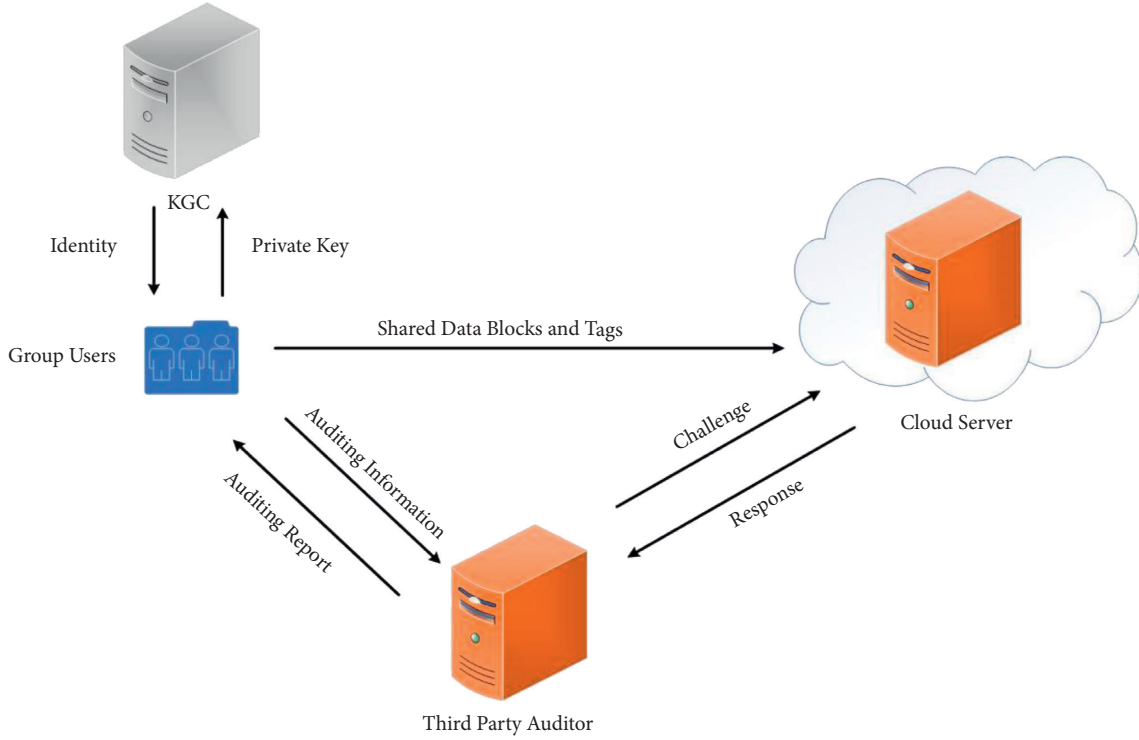


FIGURE 1: System model.

the PDP scheme. Here, we focus on the security model of auditing soundness of the cloud storage auditing protocol. The security model is described as follows: a game between an adversary server A and a challenger B .

- (i) *Setup*. The algorithm *Setup* is run by challenger B , and the master key msk and public parameters $param$ are obtained. Then, challenger B forwards the parameters to the server A but keeps msk secret.
- (ii) *Query*. The adversary server A adaptively makes the following queries to the challenger B :
 - (a) *Hash Query*. A performs a hash function query, and B uses the hash value as a response.
 - (b) *Extract Query*. A can query any user's key through it. B obtains the key by running the *Extract* algorithm and sends the obtained result to A .
 - (c) *Tag Query*. A queries the tags on the input pair (ID, m) . B responds to the query by running the *TagGen* algorithm and feeds the results back to A .
- (iii) *Challenge*. Challenger B runs the *Challenge* algorithm on the file F of the user with ID , and B sends a challenge to A . Note that ID has never been queried from the *Extract* oracle, and all blocks of the file F have been queried from the *tag* oracle.
- (iv) *ProofGen*. Adversary A executes the algorithm and computes a *proof* based on the received challenge.

- (v) *ProofVerify*. If the proof is verified, A wins the game. Also, part of the proof represented by μ in the protocol is not equal to the aggregate value coming from the challenger B based on the *ProofGen* algorithm.

3. Review of Rabaninejad's Scheme

In this section, we will review the specific scheme of Rabaninejad et al. [17]. First, we review the concept of a bilinear map. G_1 and G_2 denote a cyclic additive group and a cyclic multiplicative group of the same prime order q , where g is the generator of G_1 . The bilinear map $e: G_1 \times G_1 \rightarrow G_2$ is a function with the following properties:

- (i) *Bilinearity*. $e(aP, bQ) = e(P, Q)^{ab}$, for all $P, Q \in G_1$ and $a, b \in Z_q$.
- (ii) *Non-De generacy*. $e(P, P) \neq 1$, where 1 denotes the identity element of G_2 .
- (iii) *Computability*. There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

Rabaninejad et al. [17] proposed an online/offline ID-based PDP scheme, which consists of the following algorithms. In addition to the definition and notation in the bilinear map, two hash functions $H: \{0, 1\}^* \rightarrow G_1$ and $h: \{0, 1\}^* \rightarrow Z_q$ are used in the scheme.

- (1) *Setup*. The KGC chooses a random value $\alpha \in Z_q$ as the master secret key msk and sets the master public key as $mpk = g^\alpha$. So, the system public parameters are $param = (e, q, G_1, G_2, g, mpk, h, H)$.

- (2) *Extract*. The KGC uses $param = (e, q, G_1, G_2, g, mpk, h, H)$ and $mpk = \alpha$ to generate the secret key $k_{ID} = H(ID)^\alpha$ for user ID .
- (3) *OffTagGen*. The user with identity ID chooses a secret random value $x \in Z_q$ as the trapdoor key and sets $\gamma = mpk^x$. Then, it generates an offline tag σ_i^{off} for $i \in [1, B]$ by choosing two random value (m_i^t, r_i^t) from Z_q as follows:

$$\sigma_i^{off} = (k_{ID}^x)^{m_i^t} k_{ID}^{r_i^t} = k_{ID}^{xm_i^t + r_i^t}. \quad (1)$$

At last, the offline tags $\{(m_i^t, r_i^t, \sigma_i^{off})\}_{i \in [1, B]}$ are locally stored.

- (4) *OnTagGen*. The user with identity ID owns the file F with F_{name} . First, it divides F into n blocks as $F = (m_1, \dots, m_n)$, where $m_i \in Z_q$. Then, it generates the online tag (r_i, σ_i) on block m_i based on an unused offline tag $(m_i^t, r_i^t, \sigma_i^{off})$ and uses trapdoor key as follows:

$$\begin{aligned} r_i &= r_i^t + x(m_i^t - m_i) \bmod q, \\ \sigma_i &= \sigma_i^{off}. \end{aligned} \quad (2)$$

Finally, the online $tag\{(r_i, \sigma_i)\}_{i \in [1, n]}$ together with the data blocks $F = (m_1, \dots, m_n)$ is outsourced to the cloud server. The data owner also creates an MHT on the ordered hash value $\{h(r_i)\}_{i \in [1, n]}$ with the root node $root$ and generates $\sigma_{root} = I DS(root)$. At the same time, the pair $(root, \sigma_{root})$ together with (γ, F_{name}) , $IDS(\gamma || F_{name})$ is sent to the server and the TPA. Here, IDS is a secure ID-based signature. When the server receives the tags $\{(r_i, \sigma_i)\}_{i \in [1, n]}$ and the data blocks $F = (m_1, \dots, m_n)$, it first checks whether $Verify(m_i, r_i, \sigma_i, I D, mpk, \gamma) = 1$ passes for all $i \in [1, n]$. If so, it stores $\{(m_i, r_i, \sigma_i)\}_{i \in [1, n]}$ in its storage; otherwise, it outputs \perp . Furthermore, if σ_{root} and $I DS(\gamma || F_{name})$ are valid signatures, the server and the TPA save the values' root, γ for the file name F_{name} .

- (5) *Challenge*. In order to challenge the integrity of the file F_{name} which is owned by the user with identity ID , the TPA runs the following process:
- Choose a random subset $J \subset [1, n]$ as the block indices to be challenged in the auditing process, and for each $j \in J$, choose a random value $y_j \in Z_q$.
 - Send the challenge $chal = (F_{name}, \{(j, y_j)\}_{j \in J})$ to the server.
- (6) *ProofGen*. The server generates an auditing proof according to the received challenge $chal = (F_{name}, \{(j, y_j)\}_{j \in J})$, through the following procedure:
- Computes a combination of the challenged blocks as $\mu^t = \sum_{j \in J} y_j m_j$ and sets $\mu = H(ID)^\mu$.
 - Aggregates the tags as $\sigma = \sum_{j \in J} \sigma_j^{y_j}$.

- (c) Sends back $(\mu, \sigma, \{r_j, \Delta_j\}_{j \in J})$ as the auditing proof to the TPA. Here, r_j is the first term in tag of block m_j and Δ_j is the corresponding AAI in MHT.

- (7) *ProofVerify*. When TPA receives the proof $(\mu, \sigma, \{r_j, \Delta_j\}_{j \in J})$, it first computes $root'$ from $\{h(r_j) | j \in J\}$ and the corresponding $AAI\{\Delta_j | j \in J\}$. If $root' = root$, it then computes $R = \sum_{j \in J} y_j r_j$ and checks equation (3). which indicates that the cloud storage is good or not

$$e(\sigma, g) \stackrel{?}{=} e(\mu, \gamma) \cdot e(H(ID), mpk)^R. \quad (3)$$

4. Attack I on the OffTagGen Algorithm

The attack I is as follows:

- (1) The adversary (which can be the malicious cloud server) can obtain many block-signature pairs, such as $(M_1, \sigma_1), (M_2, \sigma_2), \dots, (M_n, \sigma_n)$, and the following holds:

$$\begin{aligned} \sigma_1 &= k_{ID}^{xm_1 + r_1} = k_{ID}^{xm_1 + r_1}, \\ \sigma_2 &= k_{ID}^{xm_2 + r_2} = k_{ID}^{xm_2 + r_2}, \\ &\dots \\ \sigma_n &= k_{ID}^{xm_n + r_n} = k_{ID}^{xm_n + r_n}. \end{aligned} \quad (4)$$

- (2) Let $k_{ID}^x = A, k_{ID} = B$, and r_1, r_2, \dots, r_n be all known to the adversary; thus, the above equations can be rewritten as follows:

$$\begin{aligned} \sigma_1 &= k_{ID}^{xm_1 + r_1} = A^{m_1} B^{r_1}, \\ \sigma_2 &= k_{ID}^{xm_2 + r_2} = A^{m_2} B^{r_2}, \\ &\dots \\ \sigma_n &= k_{ID}^{xm_n + r_n} = A^{m_n} B^{r_n}. \end{aligned} \quad (5)$$

- (3) With these equations, the adversary can compute A and B . We must point out that actually two equations are enough to compute A and B . Concretely, the adversary first computes

$$\begin{aligned} \sigma_1^{m_2} &= A^{m_1 m_2} B^{r_1 m_2}, \\ \sigma_2^{m_1} &= A^{m_2 m_1} B^{r_2 m_1}, \\ \sigma_1^{r_2} &= A^{m_1 r_2} B^{r_1 r_2}, \\ \sigma_2^{r_1} &= A^{m_2 r_1} B^{r_2 r_1}, \end{aligned} \quad (6)$$

and then computes

$$\begin{aligned} \frac{\sigma_1^{m_2}}{\sigma_2^{m_1}} &= \frac{A^{m_1 m_2} B^{r_1 m_2}}{A^{m_2 m_1} B^{r_2 m_1}} = \frac{B^{r_1 m_2}}{B^{r_2 m_1}} = B^{r_1 m_2 - r_2 m_1}, \\ \frac{\sigma_1^{r_2}}{\sigma_2^{r_1}} &= \frac{A^{m_1 r_2} B^{r_1 r_2}}{A^{m_2 r_1} B^{r_2 r_1}} = \frac{A^{m_1 r_2}}{A^{m_2 r_1}} = A^{m_1 r_2 - m_2 r_1}. \end{aligned} \quad (7)$$

(4) Let $C = \sigma_1^{m_2}/\sigma_2^{m_1}$ and $D = \sigma_1^{r_2}/\sigma_2^{r_1}$:

$$C = \frac{\sigma_1^{m_2}}{\sigma_2^{m_1}} = B^{r_1 m_2 - r_2 m_1},$$

$$D = \frac{\sigma_1^{r_2}}{\sigma_2^{r_1}} = A^{m_1 r_2 - m_2 r_1}. \quad (8)$$

For the exponential prime modular, q is publicly known to all; thus, the adversary can compute A and B as follows:

$$C^{1/r_1 m_2 - r_2 m_1} = B,$$

$$D^{1/m_1 r_2 - m_2 r_1} = A. \quad (9)$$

(5) With A and B , the adversary can forge any offline and online tags; concretely, the offline tags and online tags are generated as follows:

(a) *OffTagGen*. The adversary forges offline tags σ_i^{off} for $i \in [1, B]$, and by choosing random values (\bar{m}_i, \bar{r}_i) from Z_q , it computes

$$\sigma_i^{off} = A^{\bar{m}_i} B^{\bar{r}_i} = (k_{ID}^x)^{\bar{m}_i} k_{ID}^{\bar{r}_i} = k_{ID}^{x\bar{m}_i + \bar{r}_i}. \quad (10)$$

Because the adversary knows A and B , it can compute σ_i^{off} . At last, the adversary locally stores the offline tags $\{(\bar{m}_i, \bar{r}_i, \sigma_i^{off})\}_{i \in [1, B]}$.

(b) *OnTagGen*. For the file F with F_{name} where $= (m_1, \dots, m_n)$, assume the adversary wants to modify $F = (m_1, \dots, m_n)$ to be $F = (\bar{m}_1, \bar{m}_2, \dots, \bar{m}_n)$ and then forge the tags. It generates the online tag $(\bar{r}_i, \bar{\sigma}_i)$ on block \bar{m}_i as follows:

$$\bar{r}_i = r_i \bmod q,$$

$$\bar{\sigma}_i = A^{\bar{m}_i} B^{r_i} = (k_{ID}^x)^{\bar{m}_i} k_{ID}^{r_i} = k_{ID}^{x\bar{m}_i + r_i}. \quad (11)$$

Finally, the online tags $\{(\bar{r}_i, \bar{\sigma}_i)\}_{i \in [1, n]}$ together with the data blocks $F = (\bar{m}_1, \bar{m}_2, \dots, \bar{m}_n)$ are outsourced to the cloud server. At the same time, the original pair $(root, \sigma_{root})$ together with (γ, F_{name}) , $IDS(\gamma || F_{name})$ is also sent to the server and the TPA. Here, IDS is a secure ID-based signature.

We can check that the forged tag $\bar{\sigma}_i$ a valid one because the below equation holds:

$$e(\bar{\sigma}_i, g) = e\left(k_{ID}^{x\bar{m}_i + r_i}, g\right)$$

$$= e\left((H_{ID})^{\alpha x \bar{m}_i} \cdot (H_{ID})^{\alpha r_i}, g\right) \quad (12)$$

$$= e\left((H_{ID})^{\alpha x \bar{m}_i}, \gamma\right) \cdot e\left((H_{ID})^{r_i}, mpk\right).$$

Thus, OffTagGen algorithm is not secure. Even with two block-signature pairs, anyone can first modify the contents of the blocks and then forge the offline and online tags correspondingly.

5. Attack II on the Cloud Auditing Protocol

In our attack II, we show that the adversary (which can be the malicious cloud server) can forge proof while it can even delete all the outsourced data blocks. Concretely, the attack is the following:

- (1) The first four steps of the attack are the same as attack I. The malicious cloud server can get $= k_{ID}^x$, $B = k_{ID}$ after these steps. The below steps follow the framework of the cloud auditing protocol
- (2) When the cloud server receives the tags $(r_i, \sigma_i)_{i \in [1, n]}$ and the data blocks $F = (m_1, m_2, \dots, m_n)$ outsourced by the data owner, it first checks whether $Verify(m_i, r_i, \sigma_i, ID, mpk, \gamma) = 1$ passes for all $i \in [1, n]$. If so, it stores $\{(m_i, r_i, \sigma_i)\}_{i \in [1, n]}$ in its storage; otherwise, it outputs \perp . Furthermore, if σ_{root} and $IDS(\gamma || F_{name})$ are valid signatures, the server and the TPA save the values' root, γ for the file name F_{name} .
- (3) *Challenge*. In order to challenge the integrity of the file F_{name} which is owned by the user with identity ID, the TPA runs the following process:
 - (a) Choose a random c -element subset $J \subset [1, n]$ as the block indices to be challenged in the auditing process, and for each $j \in J$, choose a random value $y_j \in Z_q$.
 - (b) Send the challenge $chal = (F_{name}, \{(j, y_j)\}_{j \in J})$ to the server.
- (4) *ProofGen*. Here we show that the malicious cloud server can even delete all the outsourced data blocks but still has the ability to return the correct auditing proof to the TPA. Note here that the malicious cloud server still stores all the tags $\{(r_i, \sigma_i)\}_{i \in [1, n]}$. Concretely, the server generates an auditing proof according to the received challenge $chal = (F_{name}, \{(j, y_j)\}_{j \in J})$, through the following procedure:
 - (a) First, it randomly chooses $\hat{m}_j \in Z_q$ ($j \in J$) and computes a combination of the challenged blocks as $\mu' = \sum_{j \in J} y_j \hat{m}_j$ and sets $\mu = H(ID)^{\mu'}$.
 - (b) For any $\hat{m}_j \in Z_q$ ($j \in J$), the malicious cloud server computes the forged tags as $\hat{\sigma}_j = A^{\hat{m}_j} B^{r_j} = (k_{ID}^x)^{\hat{m}_j} k_{ID}^{r_j} = k_{ID}^{x\hat{m}_j + r_j}$ and aggregates the tags as $\sigma = \prod_{j \in J} \hat{\sigma}_j^{y_j}$.
 - (c) It sends back $(\mu, \sigma, \{r_j, \Delta_j\}_{j \in J})$ as the auditing proof to the TPA. Here, r_j is the first term in the original tags of corresponding to the deleted block m_j and Δ_j is the corresponding AAI in MHT.
- (5) *ProofVerify*. When TPA receives the proof $(\mu, \sigma, \{r_j, \Delta_j\}_{j \in J})$, it first computes $root'$ from $\{h(r_j)_{j \in J}\}$ and the corresponding AAI $\{\Delta_j\}_{j \in J}$. If $root = root'$, it then computes $R = \sum_{j \in J} y_j r_j$ and

checks equation (13). If the equation holds, the TPA outputs 1, which means that the verification passes; otherwise, it outputs 0.

$$e(\sigma, g) \stackrel{?}{=} e(\mu, \gamma) \cdot e(H(ID), mpk)^R. \quad (13)$$

Here, we can verify that the forged proof, is a valid one because the below equation holds:

$$\begin{aligned} e(\sigma, g) &= e\left(\prod_{j \in J} \hat{\sigma}_j^{y_j}, g\right) \\ &= e\left(\prod_{j \in J} \left((k_{ID}^x)^{\hat{m}_j}\right)^{y_j}, g\right) e\left(\left(k_{ID}^{r_j}\right)^{y_j}, g\right) \\ &= e\left(\prod_{j \in J} \left((H_{ID}^{x\alpha})^{\hat{m}_j}\right)^{y_j}, g\right) e\left(\left((H_{ID}^\alpha)^{r_j}\right)^{y_j}, g\right) \\ &= e\left(\prod_{j \in J} \left((H_{ID})^{\hat{m}_j}\right)^{y_j}, g^{x\alpha}\right) e\left(\left(H_{ID}^{r_j}\right)^{y_j}, g^\alpha\right) \quad (14) \\ &= e\left(\prod_{j \in J} (H_{ID})^{y_j \hat{m}_j}, g^{x\alpha}\right) e\left(H_{ID}^{r_j y_j}, g^\alpha\right) \\ &= e\left(\prod_{j \in J} (H_{ID})^{y_j \hat{m}_j}, g^{x\alpha}\right) e\left(H_{ID}, g^\alpha\right)^{\sum_{j \in J} r_j y_j} \\ &= e\left((H_{ID})^{\sum_{j \in J} y_j \hat{m}_j}, g^{x\alpha}\right) e\left(H_{ID}, g^\alpha\right)^{\sum_{j \in J} r_j y_j} \\ &= e(\mu, \gamma) \cdot e(H(ID), mpk)^R. \end{aligned}$$

6. Our Improved Cloud Auditing Protocol

- (1) *Setup*. The KGC chooses a random value $\alpha \in Z_q$ as the master secret key msk and sets the master public key as $mpk = g^\alpha$. So, the system public parameters are $param = (e, q, G_1, G_2, g, mpk, h, H)$.
- (2) *Extract*. The KGC uses $param = (e, q, G_1, G_2, g, mpk, h, H)$ and $mpk = \alpha$ to generate the secret key $k_{ID} = H(ID)^\alpha$ for user ID.
- (3) *Offline TagGen*. The user with identity ID owns the file F with F_{name} , chooses a secret random value $x \in Z_q$ as the trapdoor key, and sets $\gamma = mpk^x$, $\gamma' = g^x$. Then, it generates an offline tag θ_i^{off} for $i \in [1, B]$, by choosing two random values (m'_i, r'_i) from Z_q as follows:

$$\begin{aligned} \sigma_i^{off} &= (k_{ID}^x)^{m'_i} k_{ID}^{r'_i} H(F_{name} || i)^x \\ &= (k_{ID})^{m_i x + r'_i} H(F_{name} || i)^x. \end{aligned} \quad (15)$$

At last, it locally stores the offline tags, $\{(m'_i, r'_i, \sigma_i^{off})\}_{i \in [1, B]}$.

- (4) *OnTagGen*. First, it divides F into n blocks as $F = (m_1, \dots, m_n)$, where $m_i \in Z_q$. Then, it generates the online tag (r_i, σ_i) on block m_i based on an unused offline tag $(m'_i, r'_i, \sigma_i^{off})$ and uses trapdoor key as follows:

$$r_i = r'_i + x(m'_i - m_i) \bmod q,$$

$$\sigma_i = \sigma_i^{off}.$$

Finally, the online tags $\{(r_i, \sigma_i)\}_{i \in [1, n]}$ together with the data blocks $F = (m_1, \dots, m_n)$ are outsourced to the cloud server. The data owner also creates an MHT on the ordered hash value $\{h(r_i)\}_{i \in [1, n]}$ with the root node $root$ and generates $\sigma_{root} = IDS(root)$. At the same time, the pair $(root, \sigma_{root})$ together with $(\gamma, F_{name}), IDS(\gamma || F_{name})$ is sent to the server and the TPA. Here, IDS is a secure ID-based signature.

When the server receives the tags $\{(r_i, \sigma_i)\}_{i \in [1, n]}$ and the data blocks $F = (m_1, \dots, m_n)$, it first checks whether $Verify(\gamma, m_i, r_i, \sigma_i, ID, mpk, \gamma) = 1$ passes for all $i \in [1, n]$. If so, it stores $\{(m_i, r_i, \sigma_i)\}_{i \in [1, n]}$ in its storage; otherwise, it outputs \perp . Furthermore, if σ_{root} and $IDS(\gamma, F_{name})$ are valid signatures, the server and the TPA save the values' $root, \gamma$ for the file name F_{name} .

- (5) *Challenge*. In order to challenge the integrity of the file F_{name} which is owned by the user with identity ID, the TPA runs the following process:
 - (a) Choose a random c -element subset $J \subset [1, n]$ as the block indices to be challenged in the auditing process, and for each $j \in J$, choose a random value $y_j \in Z_q$.
 - (b) Send the challenge $chal = (F_{name}, \{(j, y_j)\}_{j \in J})$ to the server.
- (6) *ProofGen*. The server generates an auditing proof according to the received challenge $chal = (F_{name}, \{(j, y_j)\}_{j \in J})$, through the following procedure:
 - (a) Computes a combination of the challenged blocks as $\mu^t = \sum_{j \in J} y_j m_j$ and sets $\mu = H(ID)^{\mu^t}$.
 - (b) Aggregates the tags as $\sigma = \sum_{j \in J} \sigma_j^{y_j}$.
 - (c) Sends back $(\mu, \sigma, \{r_j, \Delta_j\}_{j \in J})$ as the auditing proof to the TPA. Here, r_j is the first term in tag of block m_j and Δ_j is the corresponding AAI in MHT.
- (7) *ProofVerify*. When TPA receives the proof $(\mu, \sigma, \{r_j, \Delta_j\}_{j \in J})$, it first computes $root'$ from $\{h(r_j)_{j \in J}\}$ and the corresponding AAI $\{\Delta_j\}_{j \in J}$. If $root' = root$, it then computes $R = \sum_{j \in J} y_j r_j$ and checks equation (17).

$$e(\sigma, g) = e\left(\sum_{j \in J} (H(F_{name} || i))^{y_j}, \gamma'\right) e(\mu, \gamma) e(H(ID), mpk)^R. \quad (17)$$

7. Security Analysis

In this section, we first prove the correctness of our improved scheme. Then, we prove that the audit proof in our

proposed scheme cannot be forged, which proves that our proposed scheme can resist attacks I and II.

- (1) *Correctness*. The correctness of verification equation (17) is proved below:

$$\begin{aligned}
e(\sigma, g) &= e\left(\sum_{j \in J} \left((k_{ID}^x)^{m_{i'}} k_{ID}^{r'_j} H(F_{name} || i)^x \right)^{y_j}, g\right) \\
&= e\left(\sum_{j \in J} H((F_{name} || i)^x)^{y_j}, g\right) \cdot e\left(\sum_{j \in J} \left((k_{ID}^x)^{m_{i'}} \right)^{y_j}, g\right) \cdot e\left(\sum_{j \in J} \left(k_{ID}^{r'_j} \right)^{y_j}, g\right) \\
&= e\left(\sum_{j \in J} (H(F_{name} || i))^{y_j}, \gamma'\right) \cdot e\left(\sum_{j \in J} \left(H^{m_{i'}} \right)^{y_j}, g^{x\alpha}\right) \cdot e\left(\sum_{j \in J} \left(k_{ID}^{r'_j} \right)^{y_j}, g^x\right) \\
&= e\left(\sum_{j \in J} (H(F_{name} || i))^{y_j}, \gamma'\right) \cdot e(\mu, \gamma) \cdot e(H(ID), mpk)^R.
\end{aligned} \tag{18}$$

- (2) *Soundness*. In our improved scheme, a malicious CSP cannot forge a correct audit proof by using our attacks.

Proof

- (i) *Setup*. The challenger B chooses a random value $\alpha \in \mathbb{Z}_q$ as the master secret key msk and sets the master public key as $mpk = g^\alpha$. Then, challenger B forwards the parameters to the server A but keeps msk secret.
- (ii) *Query*. The adversary server A adaptively makes the following queries to the challenger B :
- (a) *Hash Query*. A queries hash value based on ID_i , and B chooses random $y_i \in \mathbb{Z}_q$ and then outputs $H_i = g^{y_i}$ as a response.
- (b) *Extract Query*. A can query any user's key through its ID . By running the *Extract* algorithm, B generates $K_{ID_i} = (g^{y_i})^\alpha$ and sends the obtained result to A .
- (c) *Tag Query*. A queries the tags on the input pair (PID, m) . B responds to the query by running the *TagGen* algorithm. B chooses random $r_i, r_j \in \mathbb{Z}_q$ and generates $\sigma_{ij} = ((g^x)^{r_i y_i \alpha})^{m_j} \cdot (g^{y_i \alpha})^{r_j} \cdot H(F_{name} || i)^x$. Finally, B outputs (r_j, σ_{ij}) and $(\gamma_i, IDS(\gamma_i))$ and sends them to A .
- (iii) *Challenge*. Challenger B runs the *Challenge* algorithm on the file F of the user with PID and B sends a challenge $chal^* = (F_{name}^*, \{(j, y_j)\}_{j \in J^*})$ to A .
- (iv) *ProofGen*. Adversary A executes the algorithm and computes a *Proof* $P^* = (\mu^*, \sigma, \{r_j^*, \Delta_j^*\}_{j \in J^*})$ based on the received challenge $chal^*$.
- (v) *ProofVerify*. If the proof is verified, A wins the game. Also, part of the proof represented by μ^* in the protocol is not equal to the aggregate

value μ coming from the challenger B based on the *ProofVerify* algorithm.

In the original scheme, the adversary A can compute $\mu = H(ID_i)^{\mu'}$; therefore, he can set $\mu = \mu^*$.

However, in our improved scheme, the authentication tag is calculated as equation (15):

$$\sigma_i^{off} = (k_{ID})^{m_i' x + r_i'} H(F_{name} || i)^x. \tag{19}$$

Before the malicious adversary forges an off tag, he needs to know the value of $H(F_{name} || i)^x$. However, for $i \in [1, n]$, the value of $H(F_{name} || i)^x$ is different. Even if he can get the hash value of the data $H(F_{name} || i)$, calculating $H(F_{name} || i)^x$ is as hard as solving the DL problem in G_1 , which is computationally infeasible. If p^* passes the verification, we can get $(xr_i^*)\mu' + R = (xr_i^*)\mu^* + R^*$, where $\mu \neq \mu^*$. B outputs $((\mu/\mu^*)^{\Delta R^{-1}})^{r_i^* y_i^*}$ which is a solution to the InvCDH problem and is not feasible. Therefore, the malicious CSP cannot forge a correct audit proof to pass the proof verify of TPA.

- (3) *Data Privacy against TPA*. While providing the integrity audit service to the user, TPA cannot obtain any information about the content of the user's data from the information provided by the user or from the auditing process.

Proof. On the one hand, TPA received information from user before performing the auditing work are $(root, \sigma_{root})$ and (γ, F_{name}) . The user data cannot be accessed from root due to the one-way nature of the hash function. Meanwhile, $\gamma = mpk^x$, and TPA cannot get user's data from it.

On the other hand, in the auditing process, the TPA gains $\mu = H(ID_i)^{\mu'}$. However, given $H(ID_i) \in G_1$ and $\mu \in G_1$, computing $\mu' = \sum_{j \in J} y_j m_j$ is solution to the DL problem.

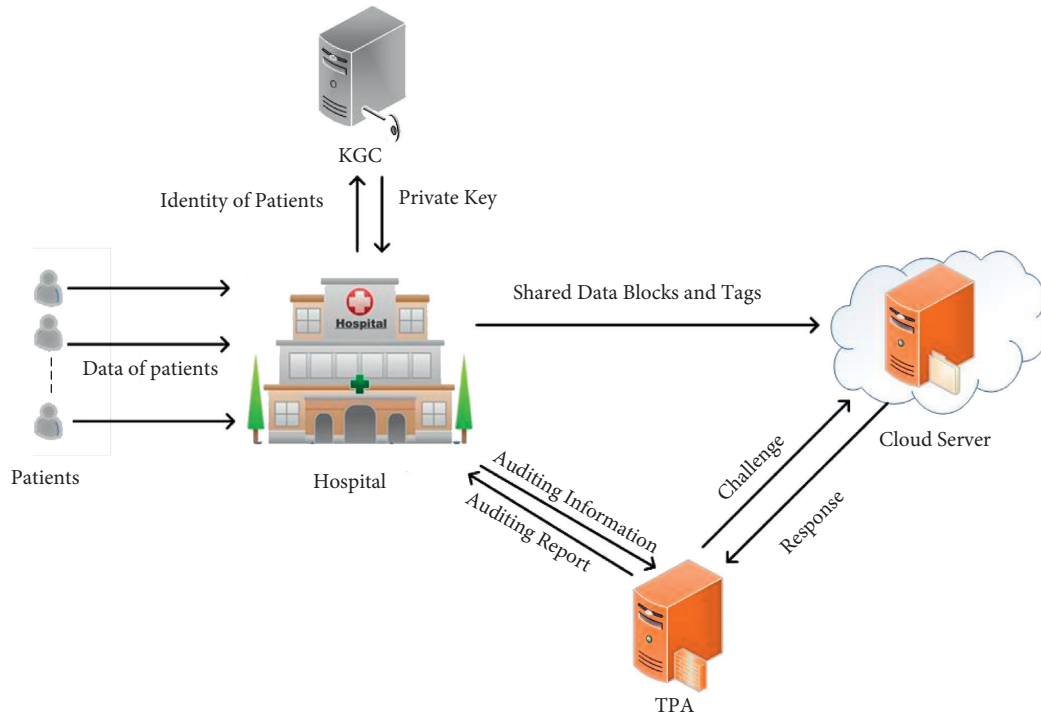


FIGURE 2: Pandemic data management in a hospital-based application.

Therefore, our improved scheme can preserve the user's data privacy.

8. Application of Our Scheme

As an application, we consider the hospital's data management in cloud setting as an example to demonstrate the effectiveness of our scheme to the actual pandemic data management. During the prevention and control of the epidemic, many medical data of patients need to be recorded, including the patients' nucleic acid testing results, the doctor's diagnosis, and if diagnosed as COVID-19, the recent whereabouts of the patients. In fact, the amount of these data is very huge; if the hospital stores them locally, it will consume a lot of storage resources, so it is better to outsource these huge data to cloud servers for storage. However, the cloud server is not completely reliable. Many important data may be lost due to various unexpected accidents. This will cause the treatment of many patients to be delayed due to the loss of data. Furthermore, the loss of some key data of diagnosed patients may lead to inadequate control of the epidemic, which may lead to the spread of the epidemic. If there is no data integrity audit mechanism, we cannot know whether the data are completely stored. So, an integrity audit mechanism is used to ensure the integrity of cloud data. At the same time, the public key is usually used to generate the authentication information of patient data. Due to the large number of patients and the continuous increase in the number of patients, the key management is a big problem. Our scheme uses identity-based way to generate public and private keys and directly uses the user's identity information to generate public keys, which effectively avoids the difficulty of key management.

Figure 2 illustrates the system model. It includes five entities. The five entities are patients, hospital (which we termed as HSP), KGC, CSP, and TPA. Because the KGC generates the private keys for the patients, it is necessary for the hospital to select a trusted key generation server as the KGC. As the cloud server needs to store a large amount of medical data, servers with strong storage capacity are selected as the cloud storage servers. Since the TPA requires a lot of audit work, a server with powerful computing power is used as the TPA. There are three steps in this model, and they are key generation, data upload, and integrity verification. The concrete implementation is as follows:

Step 1 (key generation): first, the KGC in the HSP sets the parameters and calculates the master key in the Setup stage. When the patient comes to the hospital, the hospital generates the patient's ID based on the patient's identity information and sends the ID to KGC. Then, the KGC computes the identity-based private key for the patient according to the Extract algorithm.

Step 2 (data upload): after the hospital collected the patients' data, these data need to be uploaded to the HSP's storage server. Firstly, the hospital computes the corresponding tag for the patient based on the corresponding collected data. Then, it uploads the data blocks and the corresponding tags to the HSP's storage server. According to the policy, the tags and auxiliary information based on patient's identity are transmitted to TPA, ensuring that the TPA can implement the auditing for the data stored in the cloud server.

Step 3 (integrity verification): in order to guarantee the integrity of the data, HSP needs to check it regularly. First of all, the hospital or patients make a request for

integrity verification to TPA, which in turn uses the challenge-response auditing protocol to verify the integrity of the data stored in the cloud, as requested. If the verification is successful, the patient's data are considered to be good stored in the cloud server. Otherwise, the CSP does not store the patient's data well, and other patient's data blocks may also be lost. At this time, other important data need to be checked also, and if the data are lost, remedial measures such as backup and recovery of the lost data are needed in time.

9. Conclusions

In this paper, we review a lightweight ID-based verifiable data ownership cloud storage audit scheme proposed by Rabaninejad et al. [17]. Then, we point out the security vulnerabilities in the OffTagGen and OnTagGen part of the scheme and further demonstrate the insecurity of the original protocol by showing the attack. In order to protect the integrity of users' data, an improved secure cloud audit protocol is proposed. The security analysis shows that the new protocol is secure.

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 study was supported by the National Natural Science Foundation of China under grant nos. U1636114, 62102452, and 62172436, Open Project from Guizhou Provincial Key Laboratory of Public Big Data under grant no. 2019BDKFJJ008, Engineering University of PAP's Funding for Scientific Research Innovation Team under grant no. KYTD201805, and Engineering University of PAP's Funding for Key Researcher under grant no. KYGG202011.

References

- [1] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.
- [2] M. Khorshed, A. B. M. Ali, and S. Wasimi, "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing," *Future Generation Computer Systems*, vol. 28, pp. 833–851, 2012.
- [3] J. Gudeme, S. Pasupuleti, and R. Kandukuri, "Review of remote data integrity auditing schemes in cloud computing: taxonomy, analysis, and open issues," *International Journal of Cloud Computing*, vol. 8, p. 20, 2019.
- [4] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," pp. 598–609, 2007.
- [5] M. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," *IACR Cryptology ePrint Archive*, vol. 2008, p. 186, 2008.
- [6] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 90–107, Melbourne, VIC, Australia, December 2008.
- [7] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: multiple-replica provable data possession," in *Proceedings of the 28th International Conference on Distributed Computing Systems*, Beijing, China, June 2008.
- [8] Z. Hao, S. Zhong, and N. Yu, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, pp. 1432–1437, 2011.
- [9] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceedings of the 2010 Proceedings IEEE INFOCOM*, pp. 525–533, San Diego, CA, USA, March 2010.
- [10] H. Wang, Q. Wu, B. Qin, and J. Domingo-Ferrer, "Identity-based remote data possession checking in public clouds," *Information Security*, vol. 8, pp. 114–121, 2014.
- [11] H. Wang, "Identity-based distributed provable data possession in multicloud storage," *IEEE Transactions on Services Computing*, vol. 8, pp. 328–340, 2015.
- [12] H. Wang, D. He, and S. Tang, "Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1165–1176, 2016.
- [13] Y. Yu, M. Au, G. Ateniese et al., "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 767–778, 2017.
- [14] Y. Yu, L. Xue, M. H. Au et al., "Cloud data integrity checking with an identity-based auditing mechanism from RSA," *Future Generation Computer Systems*, vol. 62, 2016.
- [15] J. Li, H. Yan, and Y. Zhang, "Identity-based privacy preserving remote data integrity checking for cloud storage," *IEEE Systems Journal*, vol. 15, no. 1, pp. 577–585, 2020.
- [16] R. Rabaninejad, S. M. Sedaghat, M. Ahmadian Attari, and M. R. Aref, "An ID-based privacy-preserving integrity verification of shared data over untrusted cloud," *Computer Society of Iran*, vol. 2020, pp. 1–6, 2020.
- [17] R. Rabaninejad, M. R. Asaar, M. A. Attari, and M. R. Aref, "An identity-based online/offline secure cloud storage auditing scheme," *Cluster Computing*, vol. 23, pp. 1455–1468, 2020.

Research Article

Public Key Encryption with Authorized Equality Test on Outsourced Ciphertexts for Cloud-Assisted IoT in Dual Server Model

Meng Zhao,¹ Yong Ding ,^{1,2} Shijie Tang,³ Hai Liang ,¹ and Huiyong Wang⁴

¹Guangxi Key Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China

²Cyberspace Security Research Center, Pengcheng Laboratory, Shenzhen, China

³School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin, China

⁴School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin, China

Correspondence should be addressed to Yong Ding; stone_dingy@126.com

Received 24 August 2021; Revised 6 October 2021; Accepted 15 October 2021; Published 20 January 2022

Academic Editor: Ximeng Liu

Copyright © 2022 Meng Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In cloud computing, the outsourced data face many privacy and security threats. To allow the cloud server to perform comparison, search, and classification on outsourced ciphertexts while simultaneously providing privacy guarantee, the encryption method that supports the ciphertext equality test is considered as a promising way. Users are able to authorize the cloud server to conduct the ciphertext equality test, so that two ciphertexts can be determined whether they encrypt the same message without being decrypted. In this process, users do not need to retrieve, decrypt, and then perform comparison on data; thus, the computing and communication efficiency can be greatly improved, and the privacy of user data can be guaranteed at the cloud server side. However, existing encryption schemes supporting authorized ciphertext equality test in the single server model cannot resist the keyword guessing attacks, and the solutions in the dual server model do not provide simultaneous authorization on two servers. To address these issues, this paper proposes a public key encryption scheme supporting authorized equality test on ciphertexts in the dual server model (PKE-AUT), where the primary server and secondary server must get the authorization from users before performing a sequential equality test on ciphertexts. Security and performance analysis demonstrate that the proposed PKE-AUT scheme not only guarantees the privacy of user data and authorization but also is practical in cloud-assisted IoT-related applications.

1. Introduction

In recent years, the cloud computing and Internet of Things (IoT) technologies have developed rapidly and become widely used. By leveraging the powerful computing capability and massive storage resources of cloud servers, the collected IoT data can be outsourced to cloud servers to save local storage and computing resources [1]. However, to guarantee the privacy of the user's sensitive information, the data should be encrypted before being outsourced, so that only the data in ciphertext format would be stored at the cloud server [2, 3]. Data encrypted with classic cryptographic schemes does not support equality test, keyword

search, calculation, and other operations on ciphertexts, so that users need to download their outsourced data to the local and then complete the corresponding operations after decryption. Thus, this process would bring huge computing and communication burdens to users, while failing to reflect the advantages of cloud computing services [4, 5].

To enable equality test on outsourced ciphertexts, many public key encryption schemes [6–8] and identity-based encryption schemes [9–12] have been proposed in the single server model. After the cloud server received the authorization from the user, it is able to perform the equality test on outsourced ciphertexts or some related operations such as encrypted data classification [13, 14] based on the equality

test, without decryption. However, since these solutions were proposed in the single cloud server model, the authorized cloud server would be able to launch keyword guessing attacks on outsourced ciphertexts to infer user data [4, 15], which causes damage to the privacy of users. Specifically, the cloud server is able to generate ciphertexts on many messages using the public keys of some users. Note that the cloud server should hold the authentication from these users. In this way, the cloud server can compare the generated ciphertexts with the stored ones, which would leak the message information if some pairs of ciphertexts are matched.

To resist the above-mentioned keyword guessing attacks faced by outsourced ciphertexts under the single server model, Wu et al. [15] proposed an identity-based encryption scheme under the dual server model for data classification in the mobile health social network. With their scheme, the user can authorize the primary server to generate relevant intermediate parameters, and the secondary server can further determine whether the two ciphertexts encrypted the same plaintext according to these intermediate parameters. These two servers would not collude to launch the attacks on outsourced user data. During the execution of their solution, the secondary server without obtaining the legal authorization of the user can perform the equality test on ciphertexts from the intermediate results generated by the primary server.

1.1. Our Contributions. This paper proposes a public key encryption scheme supporting the authorized equality test on outsourced ciphertexts (PKE-AUT) in the dual server mode. Similar to [15], the primary server and secondary server would not collude for compromising the confidentiality of outsourced data. Without authorization from the data user, both servers are unable to perform any operation on outsourced ciphertexts. After obtaining the same authorization from the data user, the primary server and secondary server sequentially perform the equality test on outsourced ciphertexts; that is, the authorized primary server produces and sends the intermediate parameters to the secondary server, then the authorized secondary server can complete the equality test procedure.

In the proposed PKE-AUT scheme, the authorizations generated for two servers are the same. The authorization is encrypted by the data user, so that only the primary server and secondary server are able to decrypt the authorization with their privacy keys, respectively; in this way, the computing costs for producing authorization can be reduced and the privacy of authentication can be protected during transmission. Security analysis shows that the proposed PKE-AUT scheme can guarantee the privacy of outsourced ciphertexts in two phases before and after the primary and secondary servers are authorized. Efficiency analysis demonstrates that the proposed PKE-AUT scheme is suitable for IoT-related applications.

1.2. Related Works. Many studies have been conducted on the authorized equality test on ciphertexts in different application scenarios. Yang et al. [6] introduced the first probabilistic public key encryption scheme with equality test on ciphertexts (PKEET), where anyone without authorization was able to

check whether the ciphertexts generated with different public keys encrypt the same data. Thus, when deployed in cloud computing, their scheme allows an unauthorized cloud server to compare the outsourced ciphertexts of different users.

Since Yang et al.'s work [6], many encryption schemes supporting the authorized equality test on ciphertexts in the single server model have been proposed [7, 16], such that the cloud server can only compare the ciphertexts after being authorized. In [17], Tang designed an all-or-nothing encryption scheme, where the cloud can test the ciphertexts only after being independently authorized by their owners. In [18], Lee et al. analyzed the security of Huang et al.'s construction [19] and presented a security-enhanced scheme. An identity-based encryption scheme with equality test on ciphertexts (IBEET) was constructed in [20], which combines the PKEET and identity-based encryption technologies. Lee et al. [21] studied the semigeneric constructions of PKEET and IBEET and proved their security under the Computational Diffie-Hellman (CDH) and Computational Bilinear Diffie-Hellman (CBDH) assumptions, respectively.

The mechanism of the equality test on ciphertexts has been used in equi-join in relational databases and secure deduplication of encrypted data. Pang and Ding [22] investigated equi-join across encrypted tables in the database in private key setting, where for an outsourced database, the user is able to control which data tables the cloud server can perform equi-join according to some data fields by issuing authorization. Then, controlled equi-join for encrypted databases in the public key setting was considered in [23]. Also, the technology of the equality test on ciphertexts was employed by Cui et al. [24] and Yan et al. [25] in achieving secure deduplication on outsourced data in clouds, without sacrificing data privacy.

Postquantum encryption schemes supporting the equality test on ciphertexts have also received attention from researchers. Le et al. [26] proposed the first lattice-based sign-cryption scheme with equality test on ciphertexts in the standard model, which was proven secure against insider attacks. Susilo et al. [27] designed an efficient postquantum IBEET scheme with smaller ciphertext and public key size, which enjoys CCA2 security. Nguyen et al. [10] presented a lattice-based IBEET scheme in the standard model, which supports flexible authorization for equality test so that the user is able to control the comparison of their ciphertexts with others.

1.3. Paper Organization. The remainder of this paper is organized as follows. Section 2 introduces the preliminaries for the proposed PKE-AUT scheme. Section 3 describes the system model and security requirements for the PKE-AUT system in the dual server model. A description of our PKE-AUT scheme is presented in Section 4, followed by the security and performance analysis in Section 5. Section 6 concludes the paper.

2. Preliminaries

This section reviews the bilinear groups, the Computational Diffie-Hellman (CDH) problem and the Computational Bilinear Diffie-Hellman (CBDH) problem.

2.1. Bilinear Groups. Let $G = \langle g \rangle$ and G_T be two cyclic groups of prime order q . The map $\widehat{e} : G \times G \longrightarrow G_T$ is a bilinear pairing if it satisfies the following conditions:

(i) *Bilinearity:* for any $g_1, g_2 \in_R G$ and $a, b \in_R \mathbb{Z}_q^*$, we have

$$\widehat{e}(g_1^a, g_2^b) = \widehat{e}(g_1, g_2)^{ab}. \quad (1)$$

(ii) *Nondegeneracy:* there exists $g_1, g_2 \in G$ such that

$$\widehat{e}(g_1, g_2) \neq 1. \quad (2)$$

(iii) *Computability:* for $g_1, g_2 \in_R G$, there is an efficient algorithm to compute $\widehat{e}(g_1, g_2)$

2.2. Complexity Assumptions. The security of our construction relies on the following two assumptions.

CDH assumption. Let $G = \langle g \rangle$ be a cyclic group of prime order q . Given a tuple (g, g^a, g^b) where $a, b \in_R \mathbb{Z}_q^*$, there is no probabilistic polynomial-time algorithm \mathcal{A} to compute g^{ab} with nonnegligible probability.

CBDH assumption. Let $G = \langle g \rangle$ and G_T be two cyclic groups of prime order q and satisfy bilinear pairing $\widehat{e} : G \times G \longrightarrow G_T$. Given a tuple (g, g^a, g^b, g^c) where $a, b, c \in_R \mathbb{Z}_q^*$, there is no probabilistic polynomial-time algorithm \mathcal{A} to compute $\widehat{e}(g, g)^{abc}$ with nonnegligible probability.

3. System Model and Security Requirements

3.1. System Model. As shown in Figure 1, the PKE-AUT system under the dual server model consists of four types of entities, namely, trusted authority, primary server, secondary server, and users. The trusted authority is responsible for initializing the system, picking the security parameter, and producing public system parameters. Both data sender and data receiver are system users. Before being uploaded to the primary server, the data is encrypted using the public keys of the data receiver and two servers, so that only the data in the ciphertext format is outsourced. The data receiver is able to retrieve the data from the primary server for decryption with his private key and issue the same authorization to the primary and secondary servers, so that the two servers can jointly perform equality test on ciphertexts.

In the PKE-AUT system, the primary server and secondary server are assumed not to collude. All outsourced data are stored at the primary server in ciphertext format to protect their privacy. After being authorized, the primary server can perform the partial equality test procedure on outsourced ciphertexts, where the intermediate results would be produced and sent to the secondary server for processing. The second server further determines whether the ciphertexts encrypt the same data according to the intermediate results and gives the final equality test result to the data user. This equality test procedure with two phases can be executed in multiuser setting; that is, the primary and secondary

servers can perform the equality test on ciphertexts of multiple users according to their authorization.

3.2. Security Requirements. In the PKE-AUT system under the dual server model, the primary server and the secondary server are independent and would not collude to attack the outsourced data. A secure PKE-AUT system has to satisfy the following requirements.

(i) *Data privacy against the primary server:* user data are stored at the primary server. Although the primary server is authorized to perform the equality test on ciphertexts, it cannot obtain the plaintexts from ciphertexts.

(ii) *Data privacy against the secondary server:* after obtaining the authorization for conducting equality test from users, the secondary server cannot deduce the plaintext information of outsourced data from the received intermediate results.

(iii) *Privacy protection on authentication:* the authentication generated by the data user can only be decrypted by the primary server and secondary server.

3.3. System Framework. A PKE-AUT scheme is composed of nine procedures, namely, the system setup, user key generation, server key generation, data encryption, data decryption, authentication generation, authentication recovery, primary server equality test, and secondary server equality test.

System setup: on input of the security parameter 1^λ , which is carried out by the trusted authority, outputs the system public parameters Para . We denote $\text{Para} \leftarrow \text{Setup}(1^\lambda)$.

User key generation: on input of the system public parameters Para , the user key generation procedure, which is carried out by each user U_i , generates a pair of public key pk_i and secret key sk_i . We denote $(pk_i, sk_i) \leftarrow \text{UKeyGen}(\text{Para})$.

Server key generation: on input of the system public parameters Para , the server key generation procedure, which is carried out by each server S_j including the primary server S_1 and secondary server S_2 , generates a pair of public key spk_j and secret key ssk_j . We denote $(spk_j, ssk_j) \leftarrow \text{SKeyGen}(\text{Para})$.

Data encryption: on input of the public keys pk_i, spk_1, spk_2 of data receiver U_i , primary server S_1 and secondary server S_2 , and a message m , the data encryption procedure, which is run by the data sender, generates a ciphertext C and outsources it to the primary server S_1 . We denote $C \leftarrow \text{Encrypt}(pk_i, spk_1, spk_2, m)$.

Data decryption: on input of the secret key sk_i of user U_i , the public keys spk_1, spk_2 of primary server S_1 and secondary server S_2 , and a ciphertext C , the data decryption procedure, which is run by the data receiver, outputs a plaintext m or \perp that signifies an error in decryption. We denote $m/\perp \leftarrow \text{Decrypt}(sk_i, spk_1, spk_2, C)$.

Authentication generation: on input of the secret key sk_i of user U_i and the public keys spk_1, spk_2 of primary server S_1

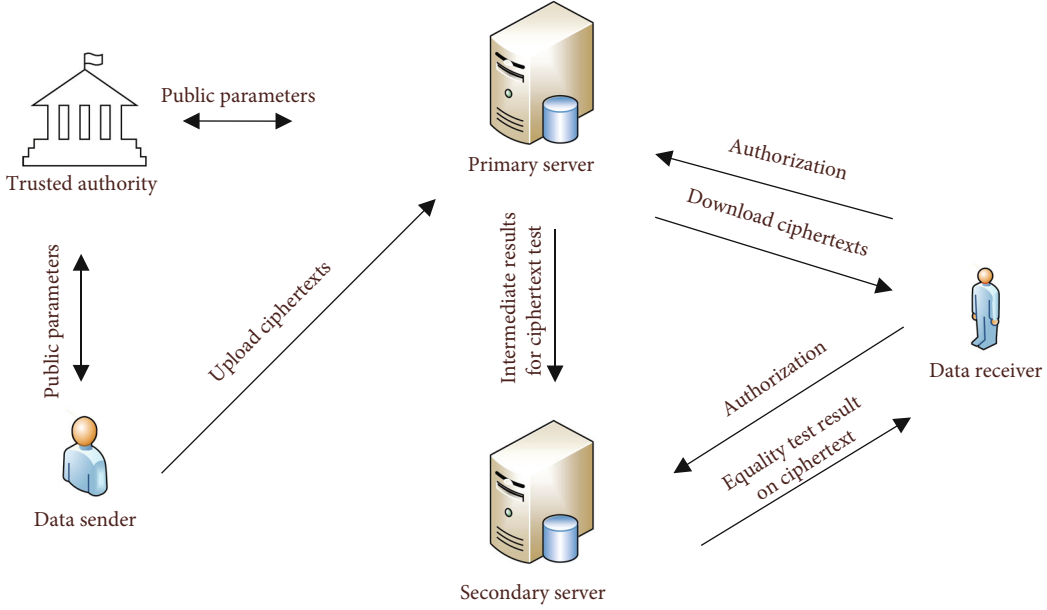


FIGURE 1: System model of PKE-AUT.

and secondary server S_2 , the authentication generation procedure, which is run by user U_i , generates a ciphertext authentication Z_i for two servers. Note that two servers have the same ciphertext authentication Z_i . We denote $Z_i \leftarrow \text{AuthGen}(sk_i, spk_1, spk_2)$.

Authentication recovery: on input of a ciphertext authentication Z_i , the secret key ssk_1 of primary server S_1 (resp., ssk_2 of secondary server S_2), and the public key spk_2 of secondary server S_2 (resp., spk_1 of primary server S_1), the authentication recovery procedure, which is run by the primary server S_1 (resp., secondary server S_2), outputs a plaintext authentication r_i or \perp that signifies an error in recovery. We denote $r_i/\perp \leftarrow \text{AuthRec}(Z_i, ssk_1, spk_2)$ or $r_i/\perp \leftarrow \text{AuthRec}(Z_i, ssk_2, spk_1)$.

Primary server equality test: on input of the authentications r_i and r_ℓ of two users U_i and U_ℓ , respectively, their public keys pk_i and pk_ℓ , their ciphertexts C and C' , and the secret key ssk_1 of the primary server S_1 , the first equality test procedure, which is run by the primary server S_1 , outputs an intermediate result Θ and gives it to the secondary server S_2 . We denote $\Theta \leftarrow \text{TestS}_1(r_i, r_\ell, pk_i, pk_\ell, C, C', ssk_1)$.

Secondary server equality test: on input of the authentications r_i and r_ℓ of two users U_i and U_ℓ , respectively, their public keys pk_i and pk_ℓ , an intermediate result Θ , and the secret key ssk_2 of the secondary server S_2 , the second equality test procedure, which is run by the secondary server S_2 , outputs 1 if C and C' encrypt the same message or 0 otherwise. We denote $1/0 \leftarrow \text{TestS}_2(r_i, r_\ell, pk_i, pk_\ell, \Theta, ssk_2)$.

A PKE-AUT scheme must be *sound* in the sense that (1) each ciphertext produced by the data encryption procedure is decryptable by the data decryption procedure; (2) the ciphertext authentication produced by the authentication generation procedure can be recovered by the authentication recovery procedure; (3) for any two ciphertexts that encrypt

the same message, which may be generated by different users, the two equality test procedures must finally output 1; and (4) for any two ciphertexts that encrypt different messages, which may be generated by different users, the two equality test procedures must finally output 0 with overwhelming probability.

Definition 1 (soundness). A PKE-AUT scheme is sound if, for any security parameter λ , any public parameters $\text{Para} \leftarrow \text{Setup}(1^\lambda)$, any public/secret key pairs of two users $(pk_i, sk_i) \leftarrow \text{UKeyGen}(\text{Para})$ and $(pk_\ell, sk_\ell) \leftarrow \text{UKeyGen}(\text{Para})$, and any public/secret key pairs of two servers $(spk_1, ssk_1) \leftarrow \text{SKeyGen}(\text{Para})$ and $(spk_2, ssk_2) \leftarrow \text{SKeyGen}(\text{Para})$, the following conditions hold:

- (i) For any message m , $\text{Decrypt}(sk_i, spk_1, spk_2, \text{Encrypt}(pk_i, spk_1, spk_2, m)) = m$.
- (ii) $\text{AuthRec}(\text{AuthGen}(sk_i, spk_1, spk_2), ssk_1, spk_2) = r_i$ and $\text{AuthRec}(\text{AuthGen}(sk_i, spk_1, spk_2), ssk_2, spk_1) = r_i$.
- (iii) For any two messages m, m' such that $C \leftarrow \text{Encrypt}(pk_i, spk_1, spk_2, m)$ and $C' \leftarrow \text{Encrypt}(pk_\ell, spk_1, spk_2, m')$, if $m = m'$, then $\text{TestS}_2(r_i, r_\ell, pk_i, pk_\ell, \Theta, ssk_2) = 1$; otherwise, $\Pr[\text{TestS}_2(r_i, r_\ell, pk_i, pk_\ell, \Theta, ssk_2) = 1] \leq \epsilon(\cdot)$, where $r_i = \text{AuthRec}(\text{AuthGen}(sk_i, spk_1, spk_2), ssk_1, spk_2) = \text{AuthRec}(\text{AuthGen}(sk_i, spk_1, spk_2), ssk_2, spk_1)$, $r_\ell = \text{AuthRec}(\text{AuthGen}(sk_\ell, spk_1, spk_2), ssk_1, spk_2) = \text{AuthRec}(\text{AuthGen}(sk_\ell, spk_1, spk_2), ssk_2, spk_1)$, and $\Theta \leftarrow \text{TestS}_1(r_i, r_\ell, pk_i, pk_\ell, C, C', ssk_1)$, and $\epsilon(\cdot)$ denotes a negligible function.

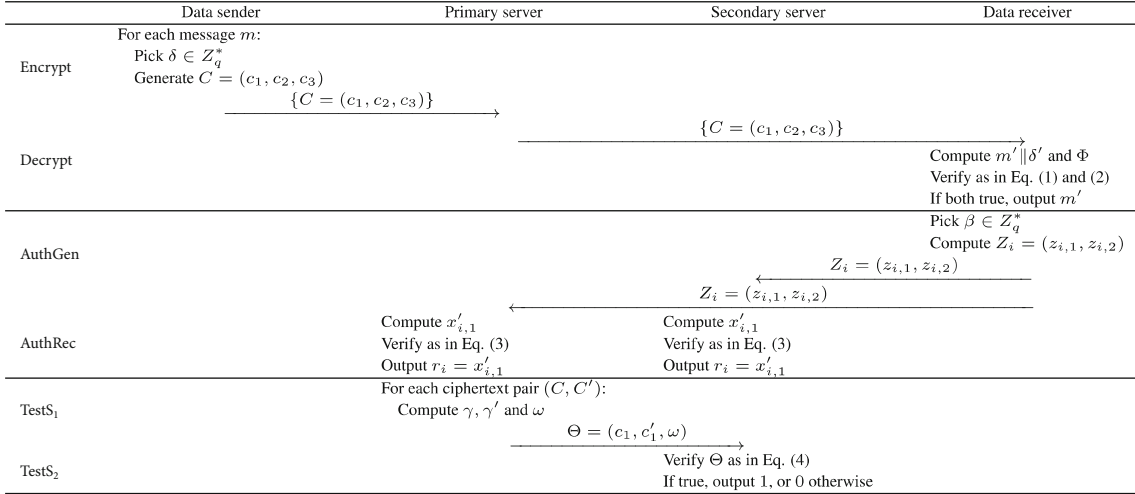


FIGURE 2: A procedure of the proposed PKE-AUT scheme.

4. PKE-AUT Construction

4.1. Concrete Construction. This section presents our PKE-AUT construction on bilinear groups in the dual server model, where a running procedure is shown in Figure 2. The frequently used symbols are summarized in Table 1.

4.1.1. System Setup. With security parameter 1^λ , the trusted authority picks two cyclic groups $G = \langle g \rangle$ and G_T of prime order q , which satisfy bilinear mapping $\hat{e} : G \times G \rightarrow G_T$. It also chooses four cryptographic hash functions $H_1 : G \times G_T \rightarrow G$, $H_2 : G \times G \rightarrow \{0, 1\}^{\tau_G + \log q}$, $H_3 : G_T \rightarrow \{0, 1\}^{\log q}$, and $H_4 : \{0, 1\}^{\tau_m} \rightarrow G$, where τ_G denotes the element size in group G and τ_m represents the size of messages. The system public parameters are $\text{Para} = (\lambda, G, G_T, q, \hat{e}, g, H_1, H_2, H_3, H_4)$.

4.1.2. User Key Generation. Each user U_i randomly picks three elements $x_{i,1}, x_{i,2}, x_{i,3} \in Z_q^*$ and computes

$$\chi_{i,1} = g^{x_{i,1}}, \chi_{i,2} = g^{x_{i,2}}, \chi_{i,3} = g^{x_{i,3}}. \quad (3)$$

Thus, the public key and secret key of user U_i are $pk_i = (\chi_{i,1}, \chi_{i,2}, \chi_{i,3})$ and $sk_i = (x_{i,1}, x_{i,2}, x_{i,3})$, respectively.

4.1.3. Server Key Generation. The primary server S_1 randomly selects two elements $y_{1,1}, y_{1,2} \in Z_q^*$ and computes

$$\rho_{1,1} = g^{y_{1,1}}, \rho_{1,2} = g^{y_{1,2}}. \quad (4)$$

Thus, the public key and secret key of primary server S_1 are $spk_1 = (\rho_{1,1}, \rho_{1,2})$ and $ssk_1 = (y_{1,1}, y_{1,2})$, respectively. In a similar way, the secondary server S_2 is able to generate its public key $spk_2 = (\rho_{2,1}, \rho_{2,2})$ and secret key $ssk_2 = (y_{2,1}, y_{2,2})$.

TABLE 1: Notations.

Symbol	Meaning
λ	Security parameter
G, G_T	Cyclic groups of prime order q satisfying bilinear pairing $\hat{e} : G \times G \rightarrow G_T$
H_1, H_2, H_3, H_4	Cryptographic hash functions
g	A generator of G
$sk_i = (x_{i,1}, x_{i,2}, x_{i,3})$	Private key of user U_i
$pk_i = (\chi_{i,1}, \chi_{i,2}, \chi_{i,3})$	Public key of user U_i
$spk_1 = (\rho_{1,1}, \rho_{1,2})$	Public key of primary server S_1
$ssk_1 = (y_{1,1}, y_{1,2})$	Secret key of primary server S_1
$spk_2 = (\rho_{2,1}, \rho_{2,2})$	Public key of secondary server S_2
$ssk_2 = (y_{2,1}, y_{2,2})$	Secret key of secondary server S_2
δ, β	Random elements in Z_q^*
$C = (c_1, c_2, c_3)$	Ciphertext of message m
$Z_i = (z_{i,1}, z_{i,2})$	Authentication of user U_i in ciphertext format
r_i, r_ℓ	Authentications of users U_i and U_ℓ
$\Theta = (c_1, c'_1, \omega)$	Intermediate result of equality test
γ, γ'	Temporary elements for computing ω

4.1.4. Data Encryption. For a message $m \in \{0, 1\}^{\tau_m}$, the data sender randomly picks $\delta \in Z_q^*$ and computes the ciphertext $C = (c_1, c_2, c_3)$ as follows:

$$\begin{aligned} c_1 &= g^\delta, \\ c_2 &= H_4(m) \cdot H_1\left(\chi_{i,1}^\delta \parallel \hat{e}(\chi_{i,2}, \rho_{1,1})^\delta\right) \cdot H_1\left(\chi_{i,1}^\delta \parallel \hat{e}(\chi_{i,2}, \rho_{2,1})^\delta\right), \\ c_3 &= (m || \delta) \oplus H_2\left(\chi_{i,3}^\delta \parallel H_4(m)\right), \end{aligned} \quad (5)$$

where \parallel denotes the concatenation of strings and \oplus represents the XOR operation. Then, the ciphertext $C = (c_1, c_2, c_3)$ is sent to the primary server S_1 .

4.1.5. Data Decryption. Given a ciphertext $C = (c_1, c_2, c_3)$, the data receiver computes

$$m' \parallel \delta' = c_3 \oplus H_2(c_1^{x_{i,3}} \parallel \Phi), \quad (6)$$

where

$$\Phi = \frac{c_2}{H_1(c_1^{x_{i,1}} \parallel \widehat{e}(c_1, \rho_{1,1})^{x_{i,2}}) \cdot H_1(c_1^{x_{i,1}} \parallel \widehat{e}(c_1, \rho_{2,1})^{x_{i,2}})}, \quad (7)$$

then verifies

$$c_1 \stackrel{?}{=} g^{\delta'}, \quad (8)$$

$$\Phi \stackrel{?}{=} H_4(m'). \quad (9)$$

If both equalities hold, then the data receiver outputs m' , otherwise \perp .

4.1.6. Authentication Generation. Data user U_i randomly picks an element $\beta \in Z_q^*$ and computes the ciphertext authentication $Z_i = (z_{i,1}, z_{i,2})$ as follows:

$$\begin{aligned} z_{i,1} &= g^\beta, \\ z_{i,2} &= x_{i,1} \oplus H_3(\widehat{e}(\rho_{1,2}, \rho_{2,2})^\beta). \end{aligned} \quad (10)$$

Data user U_i sends the ciphertext authentication $Z_i = (z_{i,1}, z_{i,2})$ to two servers S_1 and S_2 .

4.1.7. Authentication Recovery. The primary server S_1 computes

$$x'_{i,1} = z_{i,2} \oplus H_3(\widehat{e}(z_{i,1}, \rho_{2,2})^{y_{1,2}}), \quad (11)$$

and verifies

$$\chi_{i,1} \stackrel{?}{=} g^{x'_{i,1}}. \quad (12)$$

If the equality in (12) is satisfied, then the primary server S_1 outputs plaintext authentication $r_i = x'_{i,1}$, otherwise outputs symbol \perp . The secondary server can run the recovery procedure to obtain the same plaintext authentication $r_i = x'_{i,1}$ in the similar way.

4.1.8. Primary Server Equality Test. For ciphertext $C = (c_1, c_2, c_3)$ of user U_i and ciphertext $C' = (c'_1, c'_2, c'_3)$ of user U_ℓ , the primary server S_1 generates the intermediate result

$\Theta = (c_1, c'_1, \omega)$ according to their authentications r_i and r_ℓ as follows. The primary server S_1 computes

$$\begin{aligned} \gamma &= \frac{c_2}{H_1(c_1^{r_i} \parallel \widehat{e}(\chi_{i,2}, c_1)^{y_{1,1}})}, \\ \gamma' &= \frac{c'_2}{H_1(c_1^{r_\ell} \parallel \widehat{e}(\chi_{\ell,2}, c'_1)^{y_{1,1}})}. \end{aligned} \quad (13)$$

It continues to compute

$$\omega = \frac{\gamma}{\gamma'}. \quad (14)$$

The intermediate result $\Theta = (c_1, c'_1, \omega)$ is sent to the secondary server S_2 .

4.1.9. Secondary Server Equality Test. For the received intermediate result $\Theta = (c_1, c'_1, \omega)$, the secondary server S_2 verifies

$$\omega \stackrel{?}{=} \frac{H_1(c_1^{r_i} \parallel \widehat{e}(\chi_{i,2}, c_1)^{y_{2,1}})}{H_1(c_1^{r_\ell} \parallel \widehat{e}(\chi_{\ell,2}, c'_1)^{y_{2,1}})}. \quad (15)$$

If the equality in (15) is satisfied, then the secondary server S_2 outputs 1; otherwise, it outputs 0.

4.2. Soundness

Theorem 1. *The proposed PKE-AUT scheme in the dual server model is sound.*

Proof.

(1) For data decryption, since

$$\begin{aligned} \Phi &= \frac{c_2}{H_1(c_1^{x_{i,1}} \parallel \widehat{e}(c_1, \rho_{1,1})^{x_{i,2}}) \cdot H_1(c_1^{x_{i,1}} \parallel \widehat{e}(c_1, \rho_{2,1})^{x_{i,2}})} \\ &= \frac{H_4(m) \cdot H_1(\chi_{i,1}^\delta \parallel \widehat{e}(\chi_{i,2}, \rho_{1,1})^\delta) \cdot H_1(\chi_{i,1}^\delta \parallel \widehat{e}(\chi_{i,2}, \rho_{2,1})^\delta)}{H_1(c_1^{x_{i,1}} \parallel \widehat{e}(g^\delta, \rho_{1,1})^{x_{i,2}}) \cdot H_1(c_1^{x_{i,1}} \parallel \widehat{e}(g^\delta, \rho_{2,1})^{x_{i,2}})} \\ &= \frac{H_4(m) \cdot H_1(c_1^{x_{i,1}} \parallel \widehat{e}(\chi_{i,2}, \rho_{1,1})^\delta) \cdot H_1(c_1^{x_{i,1}} \parallel \widehat{e}(\chi_{i,2}, \rho_{2,1})^\delta)}{H_1(c_1^{x_{i,1}} \parallel \widehat{e}(\chi_{i,2}, \rho_{1,1})^\delta) \cdot H_1(c_1^{x_{i,1}} \parallel \widehat{e}(\chi_{i,2}, \rho_{2,1})^\delta)} \\ &= H_4(m), \end{aligned} \quad (16)$$

we have

$$\begin{aligned} m' \parallel \delta' &= c_3 \oplus H_2(c_1^{x_{i,3}} \parallel \Phi) \\ &= \left((m \parallel \delta) \oplus H_2(\chi_{i,3}^\delta \parallel H_4(m)) \right) \oplus H_2(g^{\delta x_{i,3}} \parallel H_4(m)) \\ &= (m \parallel \delta) \oplus H_2(\chi_{i,3}^\delta \parallel H_4(m)) \oplus H_2(\chi_{i,3}^\delta \parallel H_4(m)) \\ &= m \parallel \delta. \end{aligned} \quad (17)$$

Thus, the equalities in (8) and (9) hold.

(2) For authentication recovery, since

$$\begin{aligned}
x'_{i,1} &= z_{i,2} \oplus H_3(\widehat{e}(z_{i,1}, \rho_{2,2})^{y_{1,2}}) \\
&= (x_{i,1} \oplus H_3(\widehat{e}(\rho_{1,2}, \rho_{2,2})^\beta)) \oplus H_3(\widehat{e}(g^\beta, \rho_{2,2})^{y_{1,2}}) \\
&= x_{i,1} \oplus H_3(\widehat{e}(\rho_{1,2}, \rho_{2,2})^\beta) \oplus H_3(\widehat{e}(\rho_{1,2}, \rho_{2,2})^\beta) \\
&= x_{i,1},
\end{aligned} \tag{18}$$

the equality in (12) is satisfied.

(3) For equality test on ciphertexts, since

$$\begin{aligned}
\gamma &= \frac{c_2}{H_1(c'_1 \parallel \widehat{e}(\chi_{i,2}, c_1)^{y_{1,1}})} \\
&= \frac{H_4(m) \cdot H_1(\chi_{i,1}^\delta \parallel \widehat{e}(\chi_{i,2}, \rho_{1,1})^\delta) \cdot H_1(\chi_{i,1}^\delta \parallel \widehat{e}(\chi_{i,2}, \rho_{2,1})^\delta)}{H_1(c'_1 \parallel \widehat{e}(\chi_{i,2}, g^\delta)^{y_{1,1}})} \\
&= \frac{H_4(m) \cdot H_1(\chi_{i,1}^\delta \parallel \widehat{e}(\chi_{i,2}, \rho_{1,1})^\delta) \cdot H_1(\chi_{i,1}^\delta \parallel \widehat{e}(\chi_{i,2}, \rho_{2,1})^\delta)}{H_1(c'_1 \parallel \widehat{e}(\chi_{i,2}, \rho_{1,1})^\delta)} \\
&= H_4(m) \cdot H_1(\chi_{i,1}^\delta \parallel \widehat{e}(\chi_{i,2}, \rho_{2,1})^\delta), \\
\gamma' &= \frac{c'_2}{H_1(c'_{1r} \parallel \widehat{e}(\chi_{\ell,2}, c'_1)^{y'_{1,1}})} \\
&= \frac{H_4(m') \cdot H_1(\chi_{\ell,1}^\delta \parallel \widehat{e}(\chi_{\ell,2}, \rho_{1,1})^{\delta'}) \cdot H_1(\chi_{\ell,1}^\delta \parallel \widehat{e}(\chi_{\ell,2}, \rho_{2,1})^{\delta'})}{H_1(c'_{1r} \parallel \widehat{e}(\chi_{\ell,2}, g^{\delta'})^{y'_{1,1}})} \\
&= \frac{H_4(m') \cdot H_1(\chi_{\ell,1}^\delta \parallel \widehat{e}(\chi_{\ell,2}, \rho_{1,1})^{\delta'}) \cdot H_1(\chi_{\ell,1}^\delta \parallel \widehat{e}(\chi_{\ell,2}, \rho_{2,1})^{\delta'})}{H_1(c'_{1r} \parallel \widehat{e}(\chi_{\ell,2}, \rho_{1,1})^{\delta'})} \\
&= H_4(m') \cdot H_1(\chi_{\ell,1}^\delta \parallel \widehat{e}(\chi_{\ell,2}, \rho_{2,1})^{\delta'}),
\end{aligned} \tag{19}$$

we have

$$\omega = \gamma/\gamma' = \frac{H_4(m) \cdot H_1(\chi_{i,1}^\delta \parallel \widehat{e}(\chi_{i,2}, \rho_{2,1})^\delta)}{H_4(m') \cdot H_1(\chi_{\ell,1}^{\delta'} \parallel \widehat{e}(\chi_{\ell,2}, \rho_{2,1})^{\delta'})}. \tag{20}$$

Also, we know

$$\begin{aligned}
\frac{H_1(c'_1 \parallel \widehat{e}(\chi_{i,2}, c_1)^{y_{2,1}})}{H_1(c'_{1r} \parallel \widehat{e}(\chi_{\ell,2}, c'_1)^{y'_{2,1}})} &= \frac{H_1(g^{\delta' x_{i,1}} \parallel \widehat{e}(\chi_{i,2}, g^\delta)^{y_{2,1}})}{H_1(g^{\delta' x'_{\ell,1}} \parallel \widehat{e}(\chi_{\ell,2}, g^{\delta'})^{y'_{2,1}})} \\
&= \frac{H_1(g^{\delta x_{i,1}} \parallel \widehat{e}(\chi_{i,2}, \rho_{2,1})^\delta)}{H_1(g^{\delta' x'_{\ell,1}} \parallel \widehat{e}(\chi_{\ell,2}, \rho_{2,1})^{\delta'})} \\
&= \frac{H_1(\chi_{i,1}^\delta \parallel \widehat{e}(\chi_{i,2}, \rho_{2,1})^\delta)}{H_1(\chi_{\ell,1}^{\delta'} \parallel \widehat{e}(\chi_{\ell,2}, \rho_{2,1})^{\delta'})}.
\end{aligned} \tag{21}$$

It can be seen that if and only if $m = m'$, the equality in (15) is satisfied.

Therefore, the proposed PKE-AUT scheme in the dual server model is sound. \square

5. Analysis and Comparison

5.1. Security Analysis

Theorem 2. *The proposed PKE-AUT scheme in the dual server model can protect the privacy of outsourced data against the primary server.*

Proof. The ciphertext in the proposed PKE-AUT scheme has the similar form in Lee et al.'s scheme [18]. The difference lies in that for generating the second element c_2 in ciphertext, all the public keys of the data receiver and two servers should be used in the proposed PKE-AUT scheme; in this way, these two servers after being authorized are allowed to jointly perform the equality test on ciphertexts with their private keys. The proof is similar to that of Theorem 4.1 in [18], except for a small difference in the simulation on the decryption oracle; that is, the proposed PKE-AUT scheme offers the indistinguishability under adaptive chosen ciphertext attacks (IND-CCA) against the primary server assuming the CDH and CBDH assumptions hold. \square

Theorem 3. *The proposed PKE-AUT scheme in the dual server model can protect the privacy of outsourced data against the secondary server.*

Proof. In the proposed PKE-AUT scheme, all outsourced ciphertexts are stored at the primary server. During the process of equality test on ciphertexts, only the intermediate result $\Theta = (c_1, \gamma, c'_1, \gamma')$ is delivered to the secondary server by the primary server. Note that the pairs (c_1, γ) and (c'_1, γ') have the similar form of Lee et al.'s scheme [18], where the difference lies in that their scheme also has another element for enabling decryption by the user. Thus, the proof is similar to that of Theorem 4.1 in [18]; that is, the proposed PKE-AUT scheme is IND-CCA secure against the secondary server under the CDH and CBDH assumptions. \square

Theorem 4. *The proposed PKE-AUT scheme in the dual server model can protect the privacy of authentication.*

Proof. The ciphertext authentication generated by the proposed PKE-AUT scheme has the similar format as the ciphertexts in Boneh and Franklin's identity-based encryption scheme (Section 4 of [28]). The difference is that in the input to the hash function H_3 , the public keys of two servers are both used in evaluating $\widehat{e}(\cdot, \cdot)$, whereas the user identity and public parameters are used in Boneh and Franklin's scheme [28]. Thus, the proof is similar to that of Theorem 4.1 in [28]; that is, the authentication in the proposed PKE-AUT scheme enjoys the indistinguishability under chosen plaintext attacks (IND-CPA) assuming the CBDH assumption holds. \square

5.2. Performance Analysis. This section analyzes the performance of the proposed PKE-AUT scheme and compares with existing schemes, where only resource-intensive operations such as exponentiation, bilinear pairing, and map-to-point hash function are considered. The comparison with Wu et al.'s scheme [15] is shown in Table 2, where Pair , Expo , Hash denote the evaluation costs of a bilinear pairing $\hat{e}(\cdot, \cdot)$, an exponentiation in group G , and a map-to-point hash function, respectively.

It can be seen from Table 2 that, for producing a pair of public and secret keys for each user, our UKeyGen procedure requires 3 exponentiations in group G . Although our UKeyGen procedure has one more exponentiation than Wu et al.'s scheme [15], it does not take any map-to-point hash evaluation. The SKeyGen procedure in our PKE-AUT scheme is executed by the primary server and secondary server, respectively, for generating their public and secret keys. Thus, their key pairs have the same form, where each takes 2 exponentiations in group G . While in Wu et al.'s scheme [15], the two servers run different key generation procedures, which implies their key pairs are in different form and take two and one exponentiation in group G , respectively.

In the data encryption phase, the exponentiations in group G_T in our PKE-AUT scheme and Wu et al.'s scheme [15] can be transformed into exponentiations in group G ; in this way, the corresponding parameters can be used in multiple steps and the efficiency can be improved. In this case, the Encrypt of our PKE-AUT scheme takes one less bilinear pairing operation than that in Wu et al.'s scheme [15] for encrypting a message. Note that our PKE-AUT scheme is able to concurrently authorize the primary server and secondary server to perform the equality test on ciphertexts, which makes the ciphertext contain more elements than that of Wu et al.'s scheme [15]. Thus, for data decryption, our PKE-AUT scheme should take more computations than Wu et al.'s scheme [15].

In our PKE-AUT scheme, the data user is able to generate the ciphertext authentication for two servers; that is, the same ciphertext authentication can be recovered by both the primary server and the secondary server with their respective secret keys. Thus, the computing costs for authentication generation can be reduced compared to issuing an authentication for each server separately. Since the exponentiation in group G_T can be converted to the one in group G , both AuthGen and AuthRec procedures have the same computing costs, that is, two exponentiations in group G and one map-to-point hash evaluation. In Wu et al.'s scheme [15], the privacy of authentication is not considered.

With authentication, the primary server and secondary server can cooperatively perform the equality test on ciphertexts. In our PKE-AUT scheme, both equality test procedures for two servers should take 4 more exponentiations in group G than Wu et al.'s scheme [15], since the generation of the second element c_2 in the ciphertext of our PKE-AUT scheme requires more input parameters for achieving the equality test on the ciphertext by two servers. It can be seen that the two servers in both schemes do not have the same computing costs, since the secondary server needs to

TABLE 2: Comparison of computing costs.

Procedure	Our PKE-AUT scheme	Wu et al.'s scheme [15]
UKeyGen	3Expo	2Expo + 1Hash
SKeyGen	2Expo	2Expo/1Expo
Encrypt	4Expo + 2Pair + 3Hash	5Expo + 3Pair + 2Hash
Decrypt	4Expo + 2Pair + 3Hash	2Expo + 1Pair
AuthGen	2Expo + 1Pair	—
AuthRec	2Expo + 1Pair	—
TestS_1	4Expo + 2Pair + 2Hash	2Pair + 2Hash
TestS_2	4Expo + 2Pair + 2Hash	4Pair + 2Hash

TABLE 3: Comparison of communication costs.

	Our PKE-AUT scheme	Wu et al.'s scheme [15]
Ciphertext	$2\tau_G + \tau_m + \log q$	$5\tau_G + \log q$
Authentication	$\tau_G + \log q$	τ_G
Intermediate result	$3\tau_G$	$6\tau_G$

run two bilinear pairings in generating the result of the equality test on a pair of ciphertexts.

The communication costs of our PKE-AUT scheme and Wu et al.'s scheme [15] are compared in Table 3. In our scheme, each ciphertext has three elements, while the ciphertext in Wu et al.'s scheme [15] contains five elements. Note that the message space of Wu et al.'s scheme [15] is cyclic group G . Thus, when both schemes have the same message space G , the ciphertext size of their scheme would be $2\tau_G$ more than our PKE-AUT scheme. The authentication token was not encrypted for protecting privacy in Wu et al.'s scheme [15], which only contains one element in group G . For the equality test procedure by the primary server, the generated intermediate result $\Theta = (c_1, c_1', \omega)$ in our PKE-AUT scheme has three elements in group G , while Wu et al.'s scheme [15] requires six elements in G .

Moreover, we analyze the performance of our PKE-AUT scheme and compare with Wu et al.'s scheme [15] in the dual server model according to the experimental results of cryptographic operations in [29, 30]. In [29], the experiments were conducted on a platform with Windows 7 operating system, Intel I7-4700@3.40 GHz CPU and 4GB memory. Moreover, the MIRACL Cryptographic SDK [31] was invoked with $\log p = 512$. The execution time of some cryptographic operations are summarized in Table 4.

The performance of all procedures of our PKE-AUT scheme and Wu et al.'s scheme [15] is depicted in Figures 3 and 4, respectively. The case where each procedure is executed once is considered for both schemes. It can be seen that the proposed PKE-AUT scheme is more efficient than Wu et al.'s scheme [15] in encrypting a message. Although the decryption and equality test procedures take more time than Wu et al.'s scheme [15], our PKE-AUT scheme supports strict and symmetric authorization for

TABLE 4: Execution time of cryptographic operations.

Operation	Execution time (milliseconds)
Pair	4.211
Expo	1.709
Hash	4.406

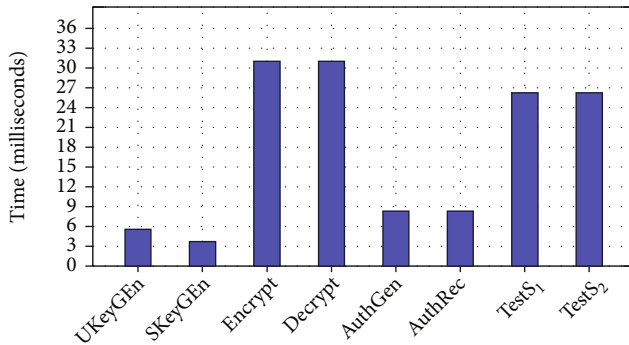


FIGURE 3: Performance of each procedure in our PKE-AUT scheme.

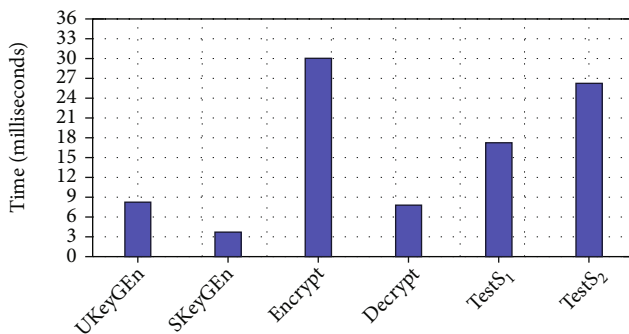


FIGURE 4: Performance of each procedure in Wu et al.'s scheme [15].

equality test on ciphertexts. Thus, to achieve this, the public keys of two servers have to be used in generating the ciphertext in our PKE-AUT scheme, which makes the efficiency of decryption and equality test reduced slightly.

6. Conclusion

To address the issues of privacy protection and resistance of keyword guessing attacks on outsourced ciphertexts in clouds, this paper presented a public key encryption scheme supporting the authorized equality test on ciphertexts in the dual server mode (PKE-AUT). User data can be only stored at the primary server to save local storage costs. With the same authentication, the primary server and secondary server can jointly carry out the equality test on ciphertexts of the corresponding users. The mechanism of the equality test on ciphertexts can be run in a multiuser setting, such that after being authorized, the two servers can compare

the ciphertexts of these multiple users. Security analysis showed that the proposed PKE-AUT scheme guarantees the privacy of outsourced ciphertexts against two servers, as well as the privacy of authentication. Performance analysis and comparison demonstrated the practicality of the proposed PKE-AUT scheme.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This article is supported in part by the National Natural Science Foundation of China under projects 61862012 and 61962012; the Guangxi Natural Science Foundation under grants 2019GXNSFFA245015 and 2019GXNSFGA245004; and the PCNL Major Key Project under grants PCL2021A09-4 and PCL2021A02-3.

References

- [1] H. Deng, Z. Qin, L. Sha, and H. Yin, "A flexible privacy-preserving data sharing scheme in cloud-assisted IoT," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11601–11611, 2020.
- [2] W.-B. Kim, D. Seo, D. Kim, and I.-Y. Lee, "Group delegated ID-based proxy reencryption for the enterprise IoT-cloud storage environment," *Wireless Communications and Mobile Computing*, vol. 2021, 12 pages, 2021.
- [3] X. Liu, R. H. Deng, K.-K. R. Choo, and Y. Yang, "Privacy-preserving outsourced clinical decision support system in the cloud," *IEEE Transactions on Services Computing*, vol. 14, no. 1, pp. 222–234, 2021.
- [4] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Server-aided public key encryption with keyword search," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2833–2842, 2016.
- [5] Y. Wang, H. H. Pang, N. H. Tran, and R. H. Deng, "CCA secure encryption supporting authorized equality test on ciphertexts in standard model and its applications," *Information Sciences*, vol. 414, pp. 289–305, 2017.
- [6] G. Yang, C. H. Tan, Q. Huang, and D. S. Wong, "Probabilistic public key encryption with equality test," in *Proceedings of the 2010 international conference on topics in cryptology, CT-RSA'10*, pp. 119–131, Berlin, Heidelberg, 2010.
- [7] S. Ma, M. Zhang, Q. Huang, and B. Yang, "Public key encryption with delegated equality test in a multi-user setting," *The Computer Journal*, vol. 58, no. 4, pp. 986–1002, 2014.
- [8] Y. Wang, Q. Huang, H. Li, J. Huang, G. Yang, and W. Susilo, "Public key authenticated encryption with designated equality test and its applications in diagnostic related groups," *IEEE Access*, vol. 7, pp. 135999–136011, 2019.
- [9] T. Wu, S. Ma, Y. Mu, and S. Zeng, "ID-based encryption with equality test against insider attack," in *Australasian conference on information security and privacy*, pp. 168–183, Cham, 2017.
- [10] G. L. D. Nguyen, W. Susilo, D. H. Duong, H. Q. Le, and F. Guo, "Lattice-based IBE with equality test supporting flexible authorization in the standard model," in *Progress in Cryptology – INDOCRYPT 2020*, K. Bhargavan, E. Oswald, and M.

- Prabhakaran, Eds., pp. 624–643, Springer International Publishing, Cham, 2020.
- [11] Y. Ling, S. Ma, Q. Huang, X. Li, Y. Zhong, and Y. Ling, “Efficient group ID-based encryption with equality test against insider attack,” *The Computer Journal*, vol. 64, no. 4, pp. 661–674, 2021.
- [12] Y. Xu, M. Wang, H. Zhong, and S. Zhong, “IBEET-AOK: ID-based encryption with equality test against off-line KGAs for cloud medical services,” *Frontiers of Computer Science*, vol. 15, no. 6, article 156814, 2021.
- [13] Y. Wang, Y. Ding, Q. Wu, Y. Wei, B. Qin, and H. Wang, “Privacy-preserving cloud-based road condition monitoring with source authentication in VANETs,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1779–1790, 2019.
- [14] H. Xiong, Y. Hou, X. Huang, and Y. Zhao, “Secure message classification services through identity-based signcryption with equality test towards the internet of vehicles,” *Vehicular Communications*, vol. 26, article 100264, 2020.
- [15] L. Wu, Y. Zhang, and D. He, “Dual server identity-based encryption with equality test for cloud computing,” *Journal of Computer Research and Development*, vol. 54, no. 10, pp. 2232–2243, 2017.
- [16] Q. Tang, “Towards public key encryption scheme supporting equality test with fine-grained authorization,” in *Proceedings of the 16th Australasian Conference on Information Security and Privacy, ACISP’11*, pp. 389–406, Berlin, Heidelberg, 2011.
- [17] Q. Tang, “Public key encryption supporting plaintext equality test and user-specified authorization,” *Security and Communication Networks*, vol. 5, no. 12, pp. 1351–1362, 2012.
- [18] H. T. Lee, S. Ling, J. H. Seo, and H. Wang, “CCA2 attack and modification of Huang et al.’s public key encryption with authorized equality test,” *The Computer Journal*, vol. 59, no. 11, pp. 1689–1694, 2016.
- [19] K. Huang, R. Tso, Y.-C. Chen, S. M. M. Rahman, A. Almogren, and A. Alamri, “PKE-AET: public key encryption with authorized equality test,” *The Computer Journal*, vol. 58, no. 10, pp. 2686–2697, 2015.
- [20] S. Ma, “Identity-based encryption with outsourced equality test in cloud computing,” *Information Sciences*, vol. 328, pp. 389–402, 2016.
- [21] H. T. Lee, S. Ling, J. H. Seo, and H. Wang, “Semi-generic construction of public key encryption and identity-based encryption with equality test,” *Information Sciences*, vol. 373, pp. 419–440, 2016.
- [22] H. Pang and X. Ding, “Privacy-preserving ad-hoc equi-join on outsourced data,” *ACM Transactions on Database Systems (TODS)*, vol. 39, no. 23, pp. 1–40, 2014.
- [23] Y. Wang and H. H. Pang, “Probabilistic public key encryption for controlled equijoin in relational databases,” *The Computer Journal*, vol. 60, no. 4, pp. 600–612, 2016.
- [24] H. Cui, R. H. Deng, Y. Li, and G. Wu, “Attribute-based storage supporting secure deduplication of encrypted data in cloud,” *IEEE Transactions on Big Data*, vol. 5, no. 3, pp. 330–342, 2019.
- [25] Z. Yan, M. Wang, Y. Li, and A. V. Vasilakos, “Encrypted data management with deduplication in cloud computing,” *IEEE Cloud Computing*, vol. 3, no. 2, pp. 28–35, 2016.
- [26] H. Q. Le, D. H. Duong, P. S. Roy, W. Susilo, K. Fukushima, and S. Kiyomoto, “Lattice-based signcryption with equality test in standard model,” *Computer Standards & Interfaces*, vol. 76, article 103515, 2021.
- [27] W. Susilo, D. H. Duong, and H. Q. Le, “Efficient post-quantum identity-based encryption with equality test,” in *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 633–640, Hong Kong, 2020.
- [28] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.
- [29] D. He, S. Zeadally, B. Xu, and X. Huang, “An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.
- [30] W. Xiong, R. Wang, Y. Wang, F. Zhou, and X. Luo, “CPPA-D: efficient conditional privacy-preserving authentication scheme with double-insurance in VANETs,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3456–3468, 2021.
- [31] *MIRACL Cryptographic SDK: multiprecision integer and rational arithmetic cryptographic library*, <https://github.com/miracl/miracl>.

Research Article

Security Guarantee for Vehicular Message Transmission Based on Dynamic Social Attributes

Lishui Chen ¹, Jing Wang ¹, Xing Chen ¹ and Yifu Zhang ²

¹The 54th Research Institute of CETC, Shijiazhuang, Hebei 050081, China

²Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Lishui Chen; 78015159@qq.com

Received 26 August 2021; Revised 28 October 2021; Accepted 22 November 2021; Published 20 December 2021

Academic Editor: Hui Zhu

Copyright © 2021 Lishui Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Effective message forwarding between vehicles can reduce the occurrence of traffic accidents and improve the driving experience. Vehicle clustering can improve message utilization, but attackers in the network pose a serious threat to message forwarding. Based on vehicle clustering, we propose a message forwarding strategy for Vehicular Ad hoc Network. Specifically, the vehicles are clustered based on their directions and speeds. Besides, the friendship of vehicles is evaluated in terms of the interaction friendship and reference friendship. Based on the friendship of vehicles, the optimal vehicle can be selected as the cluster head. Thereafter, the double key technology is designed to encrypt vehicular messages such that the messages can be forwarded more safely and efficiently. The analysis results show that the proposed strategy can effectively improve the message delivery rate, reduce the message leakage rate, and improve the network performance.

1. Introduction

As an important basis of intelligent transport system (ITS), Vehicular Ad hoc Network (VANET) is committed to the realization of intelligent traffic management and intelligent dynamic information services [1, 2]. Through vehicle to vehicle (V2V) communication and vehicle to infrastructure (V2I) communication, VANETs can reduce traffic accidents, improve road use efficiency, and promote the realization of traffic intelligence and information construction [3–5].

VANETs can improve the performance of ITS through information interaction, and the behavior of vehicles is similar to that of mobile nodes [6], so VANETs belong to a kind of continuous ad hoc wireless mobile network. The topology of VANETs is usually unstable since it changes dynamically with the high-speed movement of vehicles. In addition, the communication between vehicles in VANETs is mainly based on wireless link, which provides an opportunity for malicious vehicles to launch attacks. Therefore, the dynamic network structure of VANETs leads to serious security and privacy threats to vehicles and drivers [7], which makes it urgent to design an effective secure communication strategy.

To deal with the aforementioned issues, researchers usually design the security mechanism by using the authentication method based on public key infrastructure [8]. However, as a vehicle needs to store a large number of key pairs and their corresponding certificates that need to be transmitted with the message, the efficiency of those schemes in improving the network performance is low.

Given the high similarity of messages acquired or transmitted by vehicles in a certain range of VANETs, when a vehicle receives a useful message, the message also has reference value for its adjacent vehicles. In this case, if the vehicles independently repeat the security message transmission, it will not only cause the waste of communication resources but also be difficult to improve the communication efficiency. Although the message sharing method can solve the above problems to a certain extent, the effectiveness of the message in the communication process is not considered in the traditional broadcast method, which is prone to collision and loss of messages. As a result, the effectiveness of message sharing cannot be improved [9]. On the other hand, clustering technology is usually to group nodes in a network according to a certain relationship to enable

message transmission. Cluster communication can not only realize message sharing but also reduce the propagation of irrelevant and redundant messages, as well as reduce routing overhead and broadcast storm problems. Hence, message transmission efficiency and network performance can be improved [10]. Different clustering mechanisms have different optimization objectives and objects. Researchers have conducted in-depth research on cluster communication, and different clustering methods are introduced in detail in reference [11].

According to the characteristics of VANETs, the clustering methods of VANETs include static clustering based on the base station (BS) and dynamic clustering based on the vehicle [12]. Static clustering based on BS takes BS as cluster head, and the surrounding vehicles transmit messages to BS, and then, the BS transmits messages to other vehicles around [13]. The advantage of static clustering method is that it is easy to distinguish clusters. However, due to the long distance between two BSs and the fast change of network topology, static clustering usually leads to high message transmission delay, thus greatly reducing the accuracy and effective utilization of the message [14]. Hence, dynamic clustering based on V2V communication emerges. In this type of clustering, vehicles are screened and clustered according to certain rules, including location, speed, vehicle attribute relationship, and destination. Nevertheless, how to cluster vehicles reasonably with consideration of communication security and communication effectiveness remains an open issue.

Motivated by this, we propose a friendship assessment of security message forwarding (FASMF) strategy in VANETs. Firstly, considering the factors that affect the vehicle clustering performance and combining with the evaluation of friendship, the appropriate vehicle is selected as the cluster head. The cluster head is responsible for collecting the messages from its cluster members or other adjacent cluster heads and realizes the secure forwarding of messages by using double key message encryption within and between clusters. The effectiveness of the proposed scheme is finally validated by extensive simulations.

The remainder of the paper is organized as follows. Section 2 introduces the related works. Section 3 evaluates the vehicle friendship. Section 4 introduces the vehicle clustering scheme. Section 5 proposes a secure double key message forwarding strategy. In Section 6, simulation results are presented. Section 7 finally concludes the paper.

2. Related Work

In the literature, vehicular message transmission can be improved through vehicle clustering. In [15], the authors aimed to cluster a wide range of driving encounter scenarios based only on multivehicle GPS trajectories, where a generic unsupervised learning framework was proposed. In [16], a stochastic analysis of the impact of cluster instability on generic routing overhead was presented. In [17], the authors proposed to employ network representation learning to achieve accurate vehicle trajectory clustering, which could reduce the time and space resources. In [18], a power control

scheme in an uplink clustering network was studied for a densely vehicular network with node clustering idea. In [19], an integrated network architecture for secure group communication was proposed by taking advantage of the software-defined network technology in fifth-generation mobile networks. However, those works only focused on the vehicle cluster based on wireless communication parameters, where social relationship between vehicles was not taken into account.

Recently, some works tried to improve the security performance of intracluster and intercluster message transmissions. In [20], the authors proposed a tool called as cryptographic mix-zone to enhance vehicle privacy, in which the safety messages of vehicles were encrypted using a group secret key. In [21], a ternary join exit tree was constructed to secure communication and efficient key updating for vehicles in a platoon. In [22], an efficient security risk analysis method was proposed through fitting for evaluating the risks of attacks in the context of AV and CAVs. In [23], an efficient privacy-preserving data aggregation and dynamic pricing service PADP in V2G IoT were proposed, by designing an identity-based sequential aggregate signed data based on factoring and a threshold homomorphic encryption. However, those schemes usually introduced large amount of extraoverheads, which may degrade the delay or energy efficiency performance of VANETs.

3. Evaluation of Vehicle Friendship

The evaluation result of the vehicle-friendly relationship is the basis of the clustering strategy in this paper. The friendship of the vehicle in the network is evaluated by calculating the friendship of the vehicle. Specifically, the vehicle with high friendship is selected as the cluster head first, to ensure the reliability of message forwarding and improve the efficiency of message transmission. The direct interaction history behavior of both sides of the vehicle is selected to evaluate the interaction friendship, and the reference friendship provided by other neighboring vehicles is taken as the main factor to evaluate the vehicle friendship comprehensively.

3.1. Interactive Friendship. The vehicle has mobility and can be operated across geographical locations. If there is historical interaction between the vehicles that meet, the vehicle will obtain the vehicle interaction friendship according to the historical friendship calculated by the historical interaction experience and the interval of meeting again. If the interaction between vehicle v_i and v_j is more successful, the friendship of vehicle v_i to v_j is greater, which indicates that vehicle v_i has more sufficient evidence to forward the message to the vehicle v_j ; on the contrary, if the number of successful interactions between vehicle v_i and v_j is not frequent enough, the friendship of vehicle v_i to vehicle v_j will be reduced. Therefore, taking the number of successful interactions between vehicles as a parameter can directly evaluate the historical friendship. If there are $\text{Sum}_{i,j}$ historical interaction records between vehicle v_i and v_j in the historical interaction, the historical friendship $\langle \text{His, Fre, Deg} \rangle_{i,j}$ of vehicle

v_i to v_j can be expressed as:

$$\langle \text{His, Fre, Deg} \rangle_{i,j} = \begin{cases} \frac{\text{Suc}_{i,j}}{\text{Sum}_{i,j}} \frac{1}{\sqrt{\text{Fai}_{i,j}}}, & \text{Sum}_{i,j} \neq 0 \text{ and } \text{Sum}_{i,j} = \text{Suc}_{i,j} + \text{Fai}_{i,j}, \\ 0, & \text{Sum}_{i,j} = 0 \text{ and } \text{Sum}_{i,j} = \text{Suc}_{i,j} + \text{Fai}_{i,j}, \end{cases} \quad (1)$$

where $\text{Suc}_{i,j}$ is the number of successful transactions and $\text{Fai}_{i,j}$ is the number of failed transactions. At the same time, if there is no interactive record, the historical friendship is 0.

In addition, the time interval of vehicles meeting again will inevitably affect the level of friendship between vehicles, and the time interval of vehicles meeting again is negatively correlated with the friendship of historical interaction. That is to say, the longer the interval, the lower the referential value of historical interaction, and the less its impact on current friendship; on the contrary, the shorter the interval, the higher the value of historical cross reference, should improve the impact on the current friendship. To solve the appealing problem, Δt_{\max} is used to represent the effective time window size of historical behavior, δ is the decay rate factor, and its value is defined according to the specific application. Then, based on the historical friendship $\langle \text{His, Fre, Deg} \rangle_{i,j}$, forgetting factor α is introduced, whose value is as follows:

$$\alpha = \begin{cases} \frac{e^{\Delta t_{\max}/\delta} - e^{(t-t')/\delta}}{e^{\Delta t_{\max}/\delta} - 1}, & t - t' < \Delta t_{\max}, \\ 0, & \text{else,} \end{cases} \quad (2)$$

where t is the current time and t' is the time of the last interaction.

In conclusion, according to the historical friendship $\langle \text{His, Fre, Deg} \rangle_{i,j}$ and forgetting factor α , the interactive friendship $\langle \text{Mul, Fre, Deg} \rangle_{i,j}$ of vehicle v_i to v_j is shown in

$$\langle \text{Mul, Fre, Deg} \rangle_{i,j} = \begin{cases} \frac{\sum_{t=t'}^{\Delta t_{\max}} \alpha \cdot \langle \text{His, Fre, Deg} \rangle_{i,j}^t}{\Delta t_{\max}}, & t - t' < \Delta t_{\max}, \\ 0, & \text{else.} \end{cases} \quad (3)$$

By introducing the forgetting factor, when the historical interaction occurs beyond the effective time length, the forgetting factor α is 0, which indicates that the past interaction has lost its value; as the past interaction time t' approaches the current time t , the value of α tends to 1, which indicates that the past interaction is valuable. Therefore, using the forgetting factor can reduce the impact on network security caused by the transformation of ordinary vehicles into malicious vehicles and improve network stability.

3.2. Reference Friendship. In the process of evaluating the friendship of vehicle v_i to v_j , not only the friendship formed by the historical interaction with vehicle v_i but also the evaluation factors of other vehicle v_k to v_j and the friendship evaluation of vehicle v_i to vehicle v_k should be considered,

so as to obtain the recommended friendship $\langle \text{Rec, Fre, Deg} \rangle_{i,j}$ of vehicle v_i to v_j . In order to make the recommendation reliable, the average friendship of all neighbor recommended vehicles v_k to v_i and v_j is calculated as the value of recommended friendship $\langle \text{Rec, Fre, Deg} \rangle_{i,j}$ as shown in

$$\langle \text{Rec, Fre, Deg} \rangle_{i,j} = \frac{1}{n} \left(\sum_{k=1}^n \langle \text{Mul, Fre, Deg} \rangle_{i,k} \cdot \langle \text{Mul, Fre, Deg} \rangle_{k,j} \right), \quad (4)$$

where n denotes the number of neighbor vehicles, $\langle \text{Mul, Fre, Deg} \rangle_{i,k}$ denotes the interactive friendship of vehicle v_i to neighbor vehicle v_k , and $\langle \text{Mul, Fre, Deg} \rangle_{k,j}$ denotes the friendship of vehicle v_j provided by v_k .

As neighbor vehicle v_k may carry out malicious recommendation attacks, vehicle v_i does not fully trust the friendship $\langle \text{Mul, Fre, Deg} \rangle_{k,j}$ provided by v_k . In order to prevent malicious attack from vehicle v_k , vehicle v_i introduces a penalty factor $\text{Pui}_{i,k}$. The size of $\text{Pui}_{i,k}$ is determined by the number of interaction failures during the historical interaction between v_i and v_k .

$$\text{Pui}_{i,k} = \arctan \frac{\text{Fai}_{i,k}}{\text{Sum}_{i,k}}. \quad (5)$$

We introduce a penalty factor when the number of unsuccessful communication between vehicles increases in a short period. That is, when the vehicle behaves as malicious behavior, the friendship value of the vehicle decreases rapidly, to achieve the purpose of the abrupt decline of friendly degree. At the same time, to prevent the collusion attack between neighbor vehicle v_k and v_j leading to the rapid increase of recommendation friendship, we also consider the adjustment factor $\text{Re } g_{i,k}$, which means that with the increase of the number of successful interactive communication between vehicles, its size is closer to 1, but the approaching speed will not increase suddenly. Therefore, the calculation of the adjustment factor $\text{Re } g_{i,k}$ is as follows:

$$\text{Re } g_{i,k} = 1 - \frac{\text{Suc}_{i,k}}{1 + \text{Suc}_{i,k}}. \quad (6)$$

In conclusion, according to the recommended friendship, penalty factor, and adjustment factor, the reference friendship $\langle \text{Con, Fre, Deg} \rangle_{i,j}$ of vehicle v_i to v_j is calculated as follows:

$$\langle \text{Con, Fre, Deg} \rangle_{i,j} = \text{Pui}_{i,k}^{-1} \cdot \text{Re } g_{i,k} \cdot \langle \text{Rec, Fre, Deg} \rangle_{i,j}. \quad (7)$$

The introduction of penalty factor $\text{Pui}_{i,k}$ and adjustment factor $\text{Re } g_{i,k}$ in the calculation of reference friendship can not only effectively prevent the occurrence of malicious recommendation behavior of neighbor vehicle v_k but also reduce the influence of collusion attack between vehicles on the network, to prevent the rapid growth of reference friendship.

3.3. Friendship Integration. As mentioned above, measuring the friendship of vehicles from the above two aspects can improve the reliability of cluster heads, but the interaction between vehicles is different, so the influence of interaction friendship and reference friendship on friendship is also different. Therefore, it is necessary to allocate the weight dynamically.

Firstly, if the interaction between vehicle v_i and v_j is more frequent, the more information of vehicle v_i to v_j is, vehicle v_i has sufficient evidence to evaluate vehicle v_j ; secondly, if there is less interaction between vehicle v_i and v_j , vehicle v_i has less information about vehicle v_j , so it needs to rely more on reference to evaluate the vehicle. Therefore, the dynamic weight distribution can be achieved more accurately by taking the interaction frequency factor between vehicles as the parameter. When vehicle v_i interacts with v_j at time t , according to the historical interaction records, the proportion of all interaction time between vehicle v_i and v_j before time t in the whole time can be calculated as follows:

$$\rho = \frac{1}{\text{Sum}_{i,j}} \sum_{n=1}^{\text{Sum}_{i,j}} \frac{t_w(n)}{t_w(n) + t_s(n)}, \quad (8)$$

where $t_w(n)$ is the duration of the n th interaction and $t_s(n)$ is the interval of the n th interaction.

The more interaction times between nodes, the greater the proportion of interaction time, indicating that the interaction between v_i and v_j is more frequent. Therefore, the interaction frequency factor between vehicles is defined as

$$\omega_1 = \frac{\text{Sum}_{i,j}}{\text{Sum}_i} \times e^{\rho-1}, \quad (9)$$

where Sum_i denotes the total number of interactions of vehicle v_i before time t . Thus, the expression of friendship $\langle \text{Fre}, \text{Deg} \rangle_{i,j}$ holds as follows

$$\langle \text{Fre}, \text{Deg} \rangle_{i,j} = \omega_1 \langle \text{Mul}, \text{Fre}, \text{Deg} \rangle_{i,j} + \omega_2 \langle \text{Con}, \text{Fre}, \text{Deg} \rangle_{i,j}, \quad (10)$$

where ω_2 represents the weight of reference friendship and $\omega_2 = 1 - \omega_1$.

4. Vehicle Clustering

To improve the transmission performance of VANETs, this section proposes a clustering algorithm based on friendship, which mainly includes three processes: dynamic cluster generation, cluster head selection, and dynamic cluster maintenance.

4.1. Cluster Generation. Because the vehicles in the same direction have similar speeds and have a relatively stable communication environment for a certain period, therefore, to maintain the stability of the cluster to the maximum

extent and minimize the maintenance cost, the driving direction and speed of the vehicle are used as the basis for vehicle clustering.

Firstly, the scene is established based on the two-dimensional coordinate axis. The position and speed of vehicle v_i and v_j at time t are, respectively, represented by (x_i, y_i) , $S_i(t) \leftarrow (S_{ix}(t), y_{iy}(t))$ and (x_j, y_j) , $S_j(t) \leftarrow (S_{jx}(t), y_{jy}(t))$, the relative direction $O_{ij}(t)$, and the relative distance $D_{ij}(t)$.

Direction is the primary considered factor in the proposed clustering algorithm. At t time, the relative direction between vehicle v_i and v_j is calculated as follows:

$$O_{i,j}(t) = \cos \vartheta = \frac{S_i(t)S_j(t)}{|S_i(t)\|S_j(t)|} = \frac{S_{ix}(t)S_{jx}(t) + S_{iy}(t)S_{jy}(t)}{\sqrt{S_{ix}^2(t) + S_{iy}^2(t)}\sqrt{S_{jx}^2(t) + S_{jy}^2(t)}}, \quad (11)$$

where ϑ is the driving angle between vehicle v_i and vehicle v_j . When the value of ϑ is between $[-4/\pi, 4/\pi]$, it means that the vehicles have the same driving direction and can generate clusters; otherwise, it indicates the opposite direction of travel and cannot generate clusters.

The V2V communication in VANETs adopts DSRC technology, and its communication range is limited. Therefore, distance is the reference content of VANET clustering algorithm. At t time, the relative distance between vehicle v_i and v_j is

$$D_{i,j}(t) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (12)$$

It is necessary for vehicle v_i and v_j to form a cluster when $D_{i,j}(t)$ is in the DSRC transmission range. If $D_{i,j}(t)$ is beyond its range, vehicle v_i and v_j cannot form a cluster.

4.2. Selection of Cluster Head. To sum up, vehicle clustering is carried out concerning vehicle driving direction and speed, and the structure diagram of vehicle clustering is shown in Figure 1.

Among them, ordinary vehicles can only participate in the interaction as service requester or service provider, and cluster head (with ordinary vehicle identity) is responsible for the maintenance and management of the blacklist of the cluster and the relay of intercluster communication.

As mentioned earlier, cluster heads play an important role in the communication process of VANETs [24]. To ensure the high reliability of the leader and reduce the computing cost of the vehicle, we adopt the method of combining the friendship evaluation with the roadside unit- (RSU-) assisted selection of cluster heads.

When selecting a cluster head, not only the friendship is calculated according to the interaction behavior but also the relative mobility of vehicles should be considered. The main reason of considering the relative mobility of vehicles instead of the driving speed is that the vehicles in VANETs are all moving. The relative mobility RM_i of vehicle v_i can be

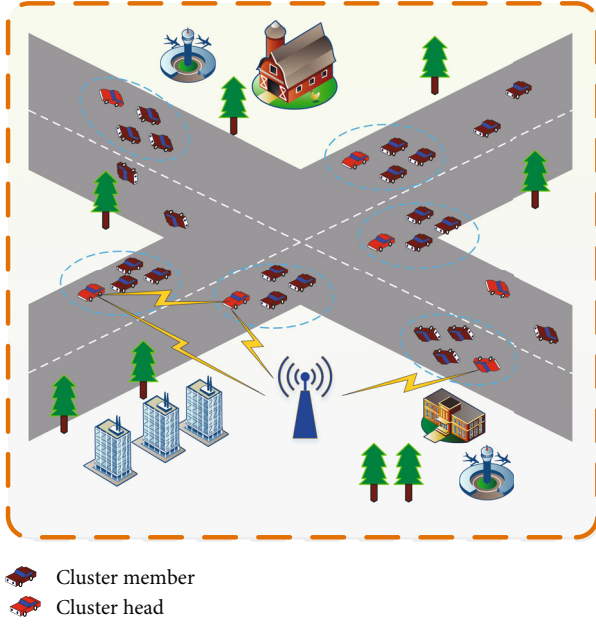


FIGURE 1: Vehicle cluster structure.

calculated in the following [25]:

$$RM_i = 1 - \frac{1/n \sum_{j=1}^n V_{i,j}}{2V_{\max}}, \quad (13)$$

$$V_{i,j} = \frac{D_{i,j}}{T}, \quad (14)$$

where $V_{i,j}$ denotes the relative speed between v_i and v_j and T denotes the time length of a cluster remaining duration. According to the above expression, the value of RM_i is within the range $[0,1]$. The larger RM_i of a cluster head holds, the more stability the cluster is.

As mentioned above, vehicle-friendship conditions $\langle Fre, Deg \rangle_{i,j}$ and RM_i are two factors for selecting cluster heads. However, compared with the size of RM_i , the friendly state $\langle Fre, Deg \rangle_{i,j}$ of the vehicle has priority over RM_i . The reason is that the stability of the cluster can be maintained not only by selecting the larger RM_i but also by the subsequent cluster maintenance process. However, if the friendly vehicle is selected as V_{head} , the probability of a malicious vehicle as V_{head} will be increased accordingly, which will have an irreversible negative impact on network stability and security.

Therefore, the specific process of selecting V_{head} is as follows: firstly, the friendship of vehicle v_j is calculated and sent to RSU by cluster vehicle v_i according to the friendship assessment proposed in section 2-B; then, RSU selects the most friendly vehicle as cluster head V_{head} . If there are vehicles with the same friendship, V_{head} is selected according to the size of RM_i . Algorithm 1 describes how to select V_{head} . According to Algorithm 1, each vehicle can calculate and report its friendship to other vehicles independently; as a result, the friendship

value comparison among all the vehicles needs computation complexity $O(n)$. Moreover, if there exist some vehicles with the same highest friendship evaluation, the cluster head is chosen by further comparing the relative mobility of those vehicles, the number of which (denoted by constant C) is far lower than n . In summary, the computation complexity of the cluster head selection holds as $O(nC)$.

4.3. Maintenance of Dynamic Cluster. Due to the rapid movement of vehicles in the network, the vehicles in the cluster will leave continuously or the vehicles outside the cluster will join at any time, which leads to the change of the network topology of the cluster. Therefore, to maintain the relative stability of the cluster topology as far as possible and reduce the impact of the change of the vehicle cluster structure on the network performance, this section aims to formulate the corresponding dynamic cluster maintenance strategy to ensure the stability of the whole network as far as possible.

Firstly, the departure of different types of vehicles in the cluster can be divided into the following two cases:

- (1) If cluster head vehicle v_{head} wants to leave, the cluster will no longer exist
- (2) If vehicle v_i is ready to leave the current cluster, v_i will first send the departure message to v_{head} , and then, vehicle v_i can leave after v_{head} confirms. At the same time, v_{head} sends the message of vehicle v_i leaving to the vehicles in the cluster to update the information in time. When v_i leaves the region of the original cluster and enters the region of other clusters, that is, v_i detects a new v'_{head} instead of the original v_{head} , v_i will join a new cluster as a cluster member or become a cluster head according to the cluster head selection process

Secondly, the addition of vehicles outside the cluster can also be divided into the following three cases:

- (1) If vehicle v_j joins the network for the first time, it can use the supervision of neighbor vehicles and calculate the friendship according to Equation (10) as the friendship of v_j
- (2) If the vehicle v_j moves from cluster C_1 to cluster C_2 , in order to reduce the observation time, the behavior of v_j in cluster C_1 , i.e., $v_{\text{head}-1}$ is the friendship issued by v_j as the recommended friendship from $v_{\text{head}-1}$ to $v_{\text{head}-2}$, and the reference friendship of $v_{\text{head}-2}$ to C_1 is calculated according to formula (7).

In addition, according to the direct interaction history of $v_{\text{head}-2}$ and v_j , the interaction friendship of $v_{\text{head}-2}$ to v_j is calculated by using formula (3). Furthermore, the friendship of v_j in cluster C_2 is calculated according to formula (10).

It is worth noting that in the above two cases, if the calculated friendship to vehicles outside the cluster is lower


```

1: Calculate  $O_{i,j}(t) \leftarrow$  Equation (11),  $D_{i,j}(t) \leftarrow$  Equation (12) to generate a cluster head
2: if failed then
3:   return
4: else
5:   vehicle  $v_i$  and  $v_j, \dots, v_n$  are in the cluster
6:   calculate the friendship degree for all vehicles each other in the cluster by Equation (10) and sends them to RSU
7:   if the friendship degree of  $v_i$  is the highest then
8:     the vehicle  $v_i$  is selected as cluster head
9:   else
10:    calculate the mobility of  $v_i$  and  $v_j$ 
11:    if it still cannot select the cluster head then
12:      return to the step 6
13:    else
14:      the mobility of  $v_i$  higher than  $v_j$ 
15:      the vehicle  $v_i$  is selected as cluster head
16:    end if
17:  end if
18: end if

```

ALGORITHM 1: Optimal power allocation algorithm.

than the average friendship in cluster C_2 , then $v_{\text{head}-2}$ will not be added to the cluster.

- (3) If two adjacent vehicle clusters merge, it is similar to reclustering, and a new cluster head needs to be reselected

5. Secure Double Key Message Forwarding

Although the friendly degree management method can effectively solve the problem of network internal attack in the process of cluster head selection, it cannot prevent an external attack in the process of message sending. Therefore, we design a secure and effective message forwarding strategy based on vehicle clustering, which mainly includes communication key generation and cluster communication.

5.1. Communication Key Generation. For different communication objects, this section uses a dual key system composed of vehicle's own key and cluster key [26]. Bilinear pairing is mainly used to generate key, and bilinear pairing is a way to realize identity based encryption. It defines three multiplicative cyclic groups G_1 , G_2 , and G_T of order q ; g_1 and g_2 are generators of G_1 and G_2 , respectively, and defines a mapping relation $e : G_1 \times G_2 \rightarrow G_T$ on these three groups. At the same time, it is assumed that he/she is completely credible, and the public parameters $\{g_1, g_2, G_1, G_2, G_T\}$ of the system are published.

As mentioned above, after the cluster head is generated, RSU will send the information in the cluster to him/her through the secure channel. Then, he/she selects a random number $s_i \in Z_n^*$ for v_i as its temporary private key and calculates the corresponding temporary public key $P_i = g_2^{s_i}$ in a short time. Finally, he/she generates the cluster key key_c of the cluster through the corresponding private key s_i of each vehicle in the cluster and constructs

the polynomial as follows:

$$f_c(x) = \text{key}_c + (x - s_1)(x - s_2) \cdots (x - s_n), \quad (15)$$

where $i = 1, 2, \dots, n$ and n is the number of vehicles in the cluster. He/she sends the polynomial $f_c(x)$ to the corresponding v_{head} , and then, v_{head} sends $f_c(x)$ to the vehicles in the cluster. At this time, each vehicle (including v_{head}) in the cluster can calculate the cluster key $f_c(x) = \text{key}_c$ by substituting the private key s_i .

However, considering that the cluster key will change with the change of cluster structure, and the same key is not suitable for long-term use, therefore, to protect the forward and backward security of the cluster, the double key should be transient and valid only when the cluster structure remains unchanged. Next, according to the cluster maintenance in section 3-C, the corresponding cluster key management scheme is developed.

When vehicle v_i leaves, he/she needs to update the cluster key of cluster C and delete its corresponding polynomial factor $(x - s_i)$ in polynomial $f_c(x)$ to protect the backward security of the cluster. Specifically, v_{head} sends the message that v_i leaves the cluster and the original $f_c(x)$ to him/her at the same time, and then, he/she regenerates $f'_c(x)$ according to section 4-A.

When vehicle v_j joins, to protect the forward security of cluster C , he/she also needs to update the cluster key of cluster C . Contrary to the case of vehicle v_i leaving cluster C , in this case, its corresponding polynomial factor $(x - s_j)$ needs to be added to the polynomial $f_c(x)$ so that v_j can participate in the communication of cluster C . Similarly, v_{head} sends the message that v_j joins the cluster with the original $f_c(x)$ to him/her at the same time, and then, he/she regenerates $f'_c(x)$ according to section 4-A. In the process of key update, to avoid excessive communication overhead caused by the

key update, as long as cluster C exists, that is, v_{head} does not leave C , he/she does not need to update the key for the original vehicle, just update the cluster key.

5.2. Message Forwarding. Cluster communication process is divided into intercluster communication and intracluster communication. When two cluster heads $v_{\text{head-1}}$ and $v_{\text{head-2}}$ confirm the communication, they will encrypt the messages to be sent with each other's public key, respectively. Take $v_{\text{head-2}}$ sending messages to $v_{\text{head-1}}$ as an example. $v_{\text{head-2}}$ encrypts the message M with $v_{\text{head-1}}$'s public key:

$$\left[E_{P_{\text{head-1}}}(\text{PID}_{\text{head-2}} || M) \right], \quad (16)$$

where $M = \{\text{Content}_M || \text{Distance}_M || \text{Time}_M\}$ and Content_M represent the content of the forwarded message, respectively, Distance_M and Time_M represent the time and place of message M , respectively, $P_{\text{head-1}}$ is the public key of $v_{\text{head-1}}$ and $\text{PID}_{\text{head-2}}$ is the pseudonym of $v_{\text{head-2}}$.

After receiving the encrypted message from $v_{\text{head-2}}$, cluster head $v_{\text{head-1}}$ decrypts the packet with its private key and uses the message:

$$\left[D_{S_{\text{head-1}}}(E_{P_{\text{head-1}}}(\text{PID}_{\text{head-2}} || M)) \right]. \quad (17)$$

Through intercluster communication, cluster head $v_{\text{head-1}}$ can obtain the specific content, time, and place of the message M from $v_{\text{head-2}}$. But at this time, only $v_{\text{head-1}}$ in cluster C_1 gets the message, so in order to make other vehicles in the cluster get the message, $v_{\text{head-1}}$ has the responsibility to forward the message to the members in the cluster. Firstly, $v_{\text{head-1}}$ encrypts the message through the cluster key Key_c of cluster C_1 and forwards it to the vehicles in the cluster:

$$\left[E_{\text{key}_c}(\text{PID}_{\text{head-1}} || M) \right]. \quad (18)$$

After receiving the message from $v_{\text{head-1}}$, the vehicles in cluster C_1 can use key_c to decrypt the ciphertext to obtain the message M :

$$\left[D_{S_{\text{head-1}}}(E_{\text{key}_c}(\text{PID}_{\text{head-1}} || M)) \right]. \quad (19)$$

It can be seen that by using clustering technology, only cluster head vehicles are required to participate in message forwarding to ensure that multiple vehicles can obtain messages at the same time, to reduce the communication resource consumption of independent communication between vehicles.

In addition, if $v_{\text{head-1}}$ obtains a malicious message from the malicious cluster head $v_{\text{head-2}}$ and forwards it to the member vehicles in the cluster, once the vehicles in the cluster, including $v_{\text{head-1}}$ and the member vehicles in the cluster, recognize that the message is false or malicious, they need to send the result to other vehicles immediately. If the malicious message is found in the vehicle $v_{\text{head-1}}$, the warning message is sent to the member vehicles in the cluster C_1

through the communication mode of the vehicles; if the vehicle that finds the malicious message is a cluster member vehicle, the cluster member vehicle forwards the warning message to the cluster head.

6. Simulation Results

In this section, the performance of the proposed strategy is validated through extensive simulations by NS2 software. To verify the effectiveness of the proposed FASMF algorithm, the proposed FASMF strategy is compared with NSTCM [20], SGC [19], and TJET [21] in terms of average message delivery rate, average message delay, and cluster stability. The simulation parameter settings are shown in Table 1.

6.1. Influence of Vehicle Number on Network Performance. With the increase of the number of vehicles in the cluster, the average delivery rate, average delay, and cluster stability of FASMF strategy and NSTCM, SGC, and TJET are shown in Figures 2, 3, and 4, respectively.

It can be seen from Figure 2 that with the increase in the number of vehicles, the message delivery rates of the FASMF strategy and the other three strategies are on the rise. Although NSTCM uses encryption to protect the security of message transmission in VANETs, it uses the method of region division to form a cluster of vehicles in the region; the vehicles in the cluster still exist independently without any social relationship, resulting in the low overall message delivery rate. In addition, the FASMF strategy, SGC, and TJET strategy proposed in this paper contain the cluster head and the corresponding security policy and forward the message to the member vehicles in the cluster through the cluster head, so it has a high message delivery rate. However, the message delivery rate of the FASMF strategy is higher than that of the SGC and TJET strategies. The reason is that SGC and TJET strategies lack corresponding security measures for messages in the process of message transmission, and SGC lacks a corresponding cluster head selection process. TJET only selects cluster heads according to the front and rear positions of vehicles, so the latter two cannot guarantee the reliability of cluster heads. After the cluster head fails, the message cannot be delivered in time.

Figure 3 shows that as the number of vehicles increases, the average message delay of the four strategies is on the rise. The delay of the NSTCM strategy is the lowest, mainly because the vehicles in its encrypted area can communicate directly and reduce the communication time. Compared with SGC and TJET strategies, the FASMF strategy has a lower average delay. The main reason is that although FASMF uses double key message encryption transmission to increase its delay, because it uses friendship to select cluster heads, the calculation process of the double key scheme adopted in this paper is smaller, while SGC and TJET strategies both use more complex key schemes; at the same time, TJET uses the tree to manage vehicles in a distributed way, which results in the highest delay.

TABLE 1: Simulation parameter setting.

Parameter	Parameter value
Network area (m^2)	3000 × 3500
Number of vehicles	50-450
Vehicle moving model	SUMO
Vehicle speed (km/h)	0-45
Vehicle communication protocol	802.11p
Vehicle communication mode	DSRC
Initial friendship	0
Simulation time (h)	6

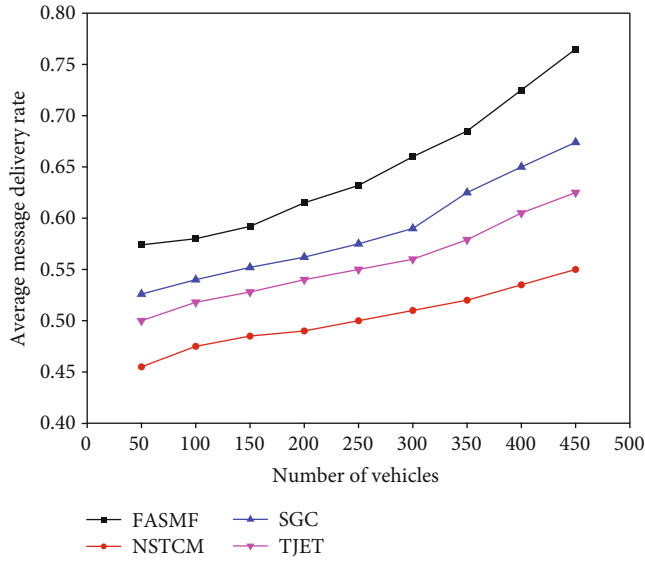


FIGURE 2: Influence of vehicle number on the average delivery rate of messages in each mechanism.

In Figure 4, with the increase of the number of vehicles in the network, the average lifetime of cluster heads of each strategy increases gradually, which indicates that the stability of clusters is positively correlated with the increase of the number of vehicles. However, the NSTCM strategy has no cluster head in the encrypted area and only uses the encryption algorithm to protect the security of the area. It cannot identify the attacker. Once the attacker enters the area and obtains the key, the security of the vehicle and communication in the area is threatened, so the network stability is the worst. In addition, the average survival time of cluster heads of the SGC strategy is longer than that of TJET. The main reason is that the vehicle tracking strategy is adopted in TJET, and the cluster head selection is determined only by the location of vehicles, so the stability of TJET is lower than SGC. Compared with the SGC strategy, the FASMF designed in this paper considers the factors of vehicle speed, direction, and so on in the process of cluster formation and uses the social friendship between vehicles to select the cluster head. If the friendship of new members is greater than the current cluster head, the FASMF will update the cluster head, so it has a better stability.

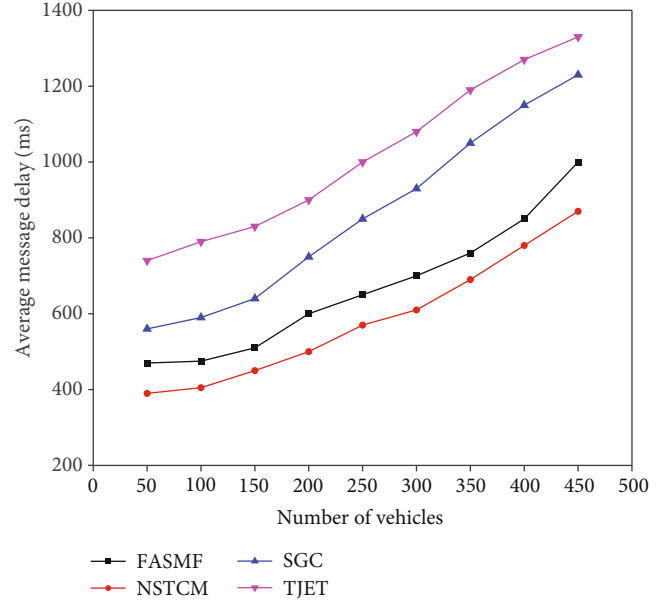


FIGURE 3: Influence of vehicle number on the average delay of messages in each mechanism.

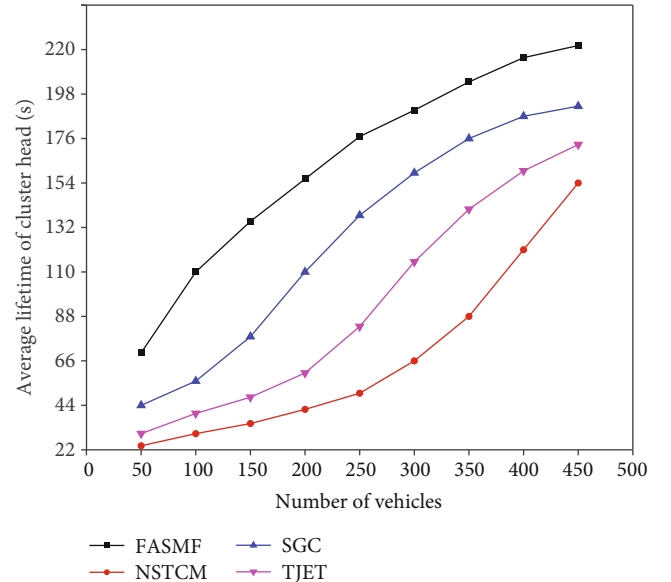


FIGURE 4: Influence of the number of vehicles on the average survival time of cluster heads.

6.2. Influence of Vehicle Speed on Network Performance. In VANETs, the speed of vehicles not only affects the formation of vehicle clusters but also the fast mobility of vehicles affects the stability of vehicle clusters. The analysis of vehicle speed on the average delivery rate, message average delay, and network stability of the proposed FASMF strategy and NSTCM, SGC, and TJET strategies are shown in Figures 5, 6, and 7, respectively.

As shown in Figure 6, the average message delivery rate of the FASMF strategy, NSTCM strategy, SGC strategy, and TJET strategy decreases with the increase of vehicle speed. This is because the faster the vehicle speed is, the frequent

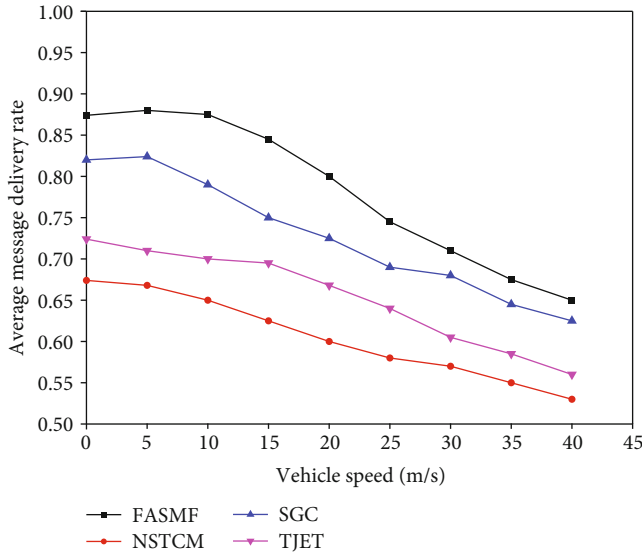


FIGURE 5: Influence of driving speed on average message delivery rate of each mechanism.

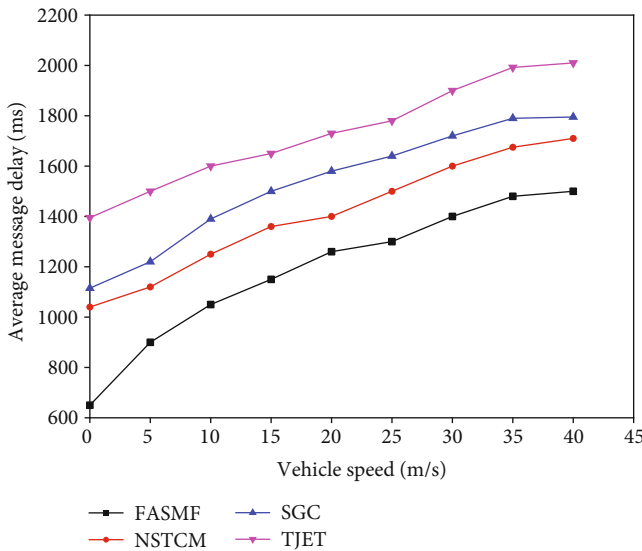


FIGURE 6: Influence of driving speed on average message delay of each mechanism.

interruption of communication links between vehicles increases the difficulty of message forwarding, and the average message delivery rate of the four schemes decreases. However, the FASMF strategy proposed in this paper involves reasonable cluster head selection, clustering, and cluster maintenance scheme, so compared with the other three strategies, the message delivery rate is higher.

As shown in Figure 7, with the increase of vehicle speed, the average message delay of FASMF, NSTCM, SGC, and TJET strategies increases in this paper. This is because the faster the driving speed is, the faster the relative relationship between vehicles will change, the easier the cluster head will be replaced by other vehicles, and more time will be consumed in cluster maintenance. In

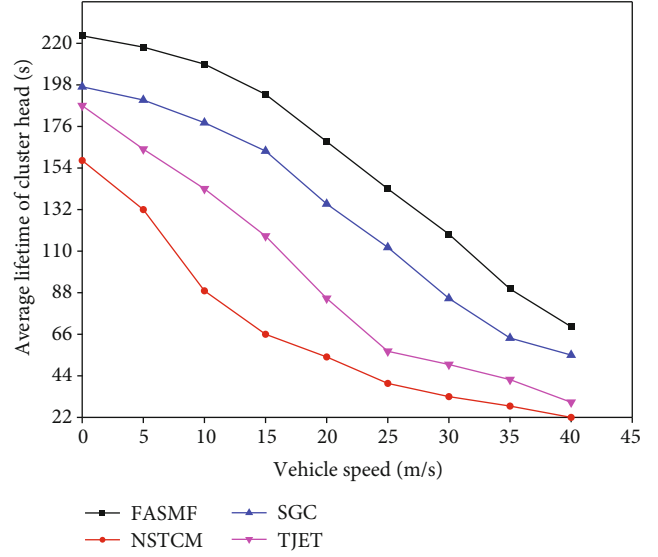


FIGURE 7: Influence of driving speed on average survival time of cluster heads.

addition, the higher the vehicle speed, the lower the stability of the cluster, and the frequent interruption of the communication link will also lead to the increase of the average message delay. However, because of the physical factors such as relative location factors and relative speed, the FASMF strategy introduced in this paper has stronger cluster stability than the other three strategies, so the time delay is the lowest. NSTCM uses the method of regional encryption, so when the vehicle speed increases, it will generate more time only in the cross-region, so its delay is lower than the SGC and TJET strategies.

As shown in Figure 7, the average lifetime of cluster heads of FASMF, NSTCM, SGC, and TJET strategies shows an overall downward trend. The reason is that the topology of VANETs is greatly affected by vehicle speed. The increase of speed leads to frequent changes in the network topology of clusters, which makes it difficult to maintain the relative stability of clusters. The NSTCM strategy uses area encryption, and the vehicles in the area can communicate. Therefore, network stability is most affected by the increase of speed, and the increase of vehicle speed reduces the duration of vehicles in the region. In addition, the average time of cluster heads of SGC and TJET strategies is lower than the FASMF strategy proposed in this paper. This is because the FASMF strategy designed in this paper not only takes into account the mobility of vehicles but also takes into account the friendship of vehicles, so the cluster heads still have strong stability.

7. Conclusion

To enhance the security of VANET message forwarding process and improve the efficiency of message forwarding, we propose a VANET message forwarding mechanism with the dynamic evaluation of friendship by combining friendship evaluation and double key method. The vehicles on

the road are divided into several clusters by clustering technology, and then, the vehicles in the cluster select the cluster head as the vehicle of intercluster communication according to the result of friendship evaluation and forward the message using message encryption. The results show that vehicle mobility as a factor of vehicle clustering can effectively improve the message delivery rate and reduce the message leakage rate, and the proposed strategy can effectively enhance the stability of the cluster topology.

Data Availability

The experiment data used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 12 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank the anonymous reviewers for their thorough and constructive comments that have helped improve the quality of the paper.

References

- [1] J. Xiong, R. Bi, Y. Tian, X. Liu, and D. Wu, "Towards light-weight, privacy-preserving cooperative object classification for connected autonomous vehicles," *IEEE Internet of Things Journal*, 2021.
- [2] D. Wu, R. Bao, Z. Li, H. Wang, H. Zhang, and R. Wang, "Edge-cloud collaboration enabled video service enhancement: a hybrid human-artificial intelligence scheme," *IEEE Transactions on Multimedia*, vol. 23, pp. 2208–2221, 2021.
- [3] A. Vangala, B. Bera, S. Saha, A. K. Das, N. Kumar, and Y. Park, "Blockchain-enabled certificate-based authentication for vehicle accident detection and notification in intelligent transportation systems," *IEEE Sensors Journal*, vol. 21, no. 14, pp. 15824–15838, 2021.
- [4] B. Zhao, X. Liu, W.-N. Chen, W. Liang, X. Zhang, and R. H. Deng, "Price: privacy and reliability-aware real-time incentive system for crowdsensing," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17584–17595, 2021.
- [5] H. Zhu, F. Wang, R. Lu, F. Liu, G. Fu, and H. Li, "Efficient and privacy-preserving proximity detection schemes for social applications," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2947–2957, 2018.
- [6] J. Li, J. Ma, Y. Miao, F. Yang, X. Liu, and K.-K. R. Choo, "Secure semantic-aware search over dynamic spatial data in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 8912–8925, 2021.
- [7] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 760–776, 2019.
- [8] F. Qu, Z. Wu, F.-Y. Wang, and W. Cho, "A security and privacy review of VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2985–2996, 2015.
- [9] X. Li, H. Zhang, Y. Miao et al., "Can bus messages abnormal detection using improved SVDD in internet of vehicle," *IEEE Internet of Things Journal*, 2021.
- [10] D. Wu, B. Yang, H. Wang, C. Wang, and R. Wang, "Privacy-preserving multimedia big data aggregation in large-scale wireless sensor networks," *Acm Transactions on Multimedia Computing Communications & Applications*, vol. 12, no. 4s, pp. 1–19, 2016.
- [11] J. Yu and P. Chong, "A survey of clustering schemes for mobile ad hoc networks," *IEEE Communications Surveys Tutorials*, vol. 7, no. 1, pp. 32–48, 2005.
- [12] T. Maniak, R. Iqbal, and F. Doctor, "Hierarchical spatial-temporal state machine for vehicle instrument cluster manufacturing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4131–4140, 2021.
- [13] M. Yang, B. Ai, R. He et al., "Measurements and cluster-based modeling of vehicle-to-vehicle channels with large vehicle obstructions," *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 5860–5874, 2020.
- [14] Z. Li, Y. Jiang, Y. Gao, L. Sang, and D. Yang, "On buffer-constrained throughput of a wireless-powered communication system," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 2, pp. 283–297, 2019.
- [15] W. Wang, A. Ramesh, J. Zhu, J. Li, and D. Zhao, "Clustering of driving encounter scenarios using connected vehicle trajectories," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 3, pp. 485–496, 2020.
- [16] K. Abboud and W. Zhuang, "Impact of microscopic vehicle mobility on cluster-based routing overhead in <roman>VANETs</roman>," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5493–5502, 2015.
- [17] W. Wang, F. Xia, H. Nie et al., "Vehicle trajectory clustering based on dynamic representation learning of internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3567–3576, 2021.
- [18] Z. Liu, Y.-A. Xie, Y. Yuan, K. Ma, K. Y. Chan, and X. Guan, "Robust power control for clustering-based vehicle-to-vehicle communication," *IEEE Systems Journal*, vol. 14, no. 2, pp. 2557–2568, 2020.
- [19] C. Lai, H. Zhou, N. Cheng, and X. S. Shen, "Secure group communications in vehicular networks: a software-defined network-enabled architecture and solution," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 40–49, 2017.
- [20] L. Zhang, "Otibaagka: a new security tool for cryptographic mix-zone establishment in vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2998–3010, 2017.
- [21] C. Xu, R. Lu, H. Wang, L. Zhu, and C. Huang, "TJET: ternary join-exit-tree based dynamic key management for vehicle platooning," *IEEE Access*, vol. 5, pp. 26973–26989, 2017.
- [22] J. Cui and B. Zhang, "Vera: a simplified security risk analysis method for autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 10494–10505, 2020.
- [23] L. Chen, J. Zhou, Y. Chen, Z. Cao, X. Dong, and K.-K. R. Choo, "PADP: efficient privacy-preserving data aggregation and dynamic pricing for vehicle-to-grid networks," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7863–7873, 2021.

- [24] Y. Tian, Z. Zhang, J. Xiong, L. Chen, J. Ma, and C. Peng, "Achieving graph clustering privacy preservation based on structure entropy in social IoT," *IEEE Internet of Things Journal*, 2021.
- [25] Z. Cui, J. Sun, X. U. Songyan, and X. Jiang, "A secure clustering algorithm of ad hoc network for colony UAVs," *Journal of Shandong University(Natural Science)*, vol. 53, no. 7, pp. 54–62, 2018.
- [26] X. Liu, R. H. Deng, K.-K. R. Choo, and Y. Yang, "Privacy-preserving outsourced clinical decision support system in the cloud," *IEEE Transactions on Services Computing*, vol. 14, no. 1, pp. 222–234, 2021.

Research Article

Decentralized Certificate Management for Network Function Virtualisation (NFV) Implementation in Telecommunication Networks

Junzhi Yan , Na Li, Bo Yang, Min Li, Li Su, and Shen He

China Mobile Research Institute, Beijing 100053, China

Correspondence should be addressed to Junzhi Yan; j.z.yan@163.com

Received 11 August 2021; Accepted 23 September 2021; Published 18 October 2021

Academic Editor: Ximeng Liu

Copyright © 2021 Junzhi Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The certificate management complexity and cost increase when PKI technology is leveraged into Network Function Virtualisation (NFV), a significant enabling technology for 5G networks. The expected security of PKI cannot be met due to the unavailability of the certificate revocation inquiry in the telecommunication operator's core network. This paper analyses the issues and challenges during the NFV implementation and proposes a blockchain-based decentralized NFV certificate management mechanism. During instantiation, the Virtual Network Functions (VNF) instance generates certificates according to the certificate profile provided in the VNF package. The certificate management unit is responsible for the certificate enrolment, renewal, and revocation. The certificates submitted to the decentralized certificate management system by the instance will be recorded into the ledger after validation and consensus. The experiment and analysis show the transaction throughput, and the transaction delay is noncritical in practice, which could be fulfilled by the proposed mechanism. The certificate inquiry performance is critical, which can be facilitated by the decentralized deployment of inquiry nodes.

1. Introduction

Network Function Virtualisation (NFV), featured as decoupling software from hardware, flexible network function deployment, and dynamic operation, is a significant enabling technology for 5G networks. In NFV, network functions are implemented by vendors in software components known as Virtual Network Functions (VNFs), which are deployed on cloud infrastructure or massively distributed servers instead of dedicated hardware [1].

The architectural framework of NFV defined by the European Telecommunication Standardization Institute (ETSI) is depicted in Figure 1. It enabled the execution and deployment of VNF on NFV infrastructure comprising a pool of network, storage, and computing resources. The NFV infrastructure is usually a decentralized cloud infrastructure in which servers are distributed over various locations. The operation, deployment, and execution of network services and VNFs in NFV infrastructure are controlled by

an orchestration and management system, whose performance is steered by NFV descriptors.

Typically, NFV is capable of overcoming certain 5G challenges, such as reducing the energy cost by maximizing the resource usage, scaling, and mobilizing VNFs from one resource to another, ensuring VNF performance operations [3]. A VNF is a virtualisation of a network function in a legacy nonvirtualised network. In 5G networks, Network Functions (NFs) defined in 3GPP TS 23.501 [4] are implemented on NFV infrastructure.

PKI public key certificates are widely used by the VNF, MANO (Management and Orchestration), and OSS/BS-S/EM (Operation Support Systems, Business Support System, Element Management) in NFV. These certificates are used for authentication and secure communication. The NFs in 5G networks use TLS protocol to connect each other [5]. However, some issues and challenges arise during the NFV deployment. These issues and challenges are related with the certificate cost, across-domain trust, CRL/OCSP

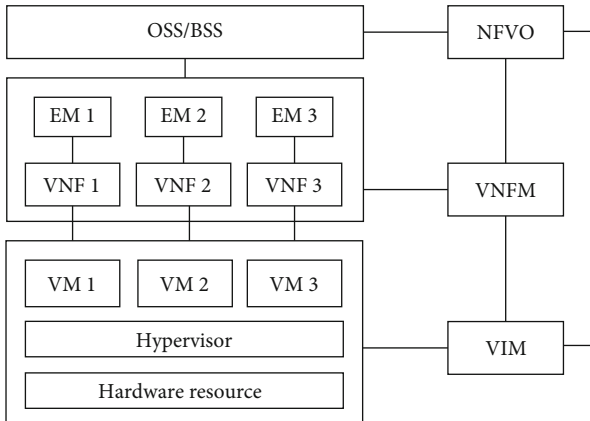


FIGURE 1: Architectural framework of NFV defined by ETSI [2].

(Certificate Revocation List/Online Certificate Status Protocol) services, certificate validation, and certificate maintenance. The essence of some issues is the lack of trust among the multiple participants in the NFV deployment, while others are related with the intranet structure of the operator's core network. Blockchain featured as decentralization and tamper resistance may benefit PKI technology [6]. The blockchain-based decentralized PKI is a significant trend for PKI technology, which could be used to facilitate the certificate management for NFV deployment.

The main contribution of this paper is the blockchain-based decentralized NFV certificate management mechanism, which is aimed at solving the issues during NFV implementation in telecommunication networks. Section 2 discusses the related researches. The issues and challenges aroused during the NFV implementation are presented in Section 3. Section 4 provides the framework of decentralized NFV certificate management mechanism and the certificate management method. The performance is evaluated and analyzed in Section 5. The conclusion is provided in Section 6.

2. Related Works

ETSI has published series of NFV standards, of which ETSI GS NFV 002 defines the architectural framework [2], ETSI GS NFV 001 provides a list of use cases and examples of target network functions for virtualisation [7], and ETSI GR NFV SEC 005 analyses the certificate management using traditional PKI technology [8].

The use of NFV technology in telecommunication networks, especially in 5G networks, has attracted much attention. An overview of enabling technologies like NFV and SDN (Software Defined Network) for 5G was provided in [9]. It highlighted challenges for ensuring an envisaged 5G networking system. The work highlighted that base station virtualisation and wireless resource sharing to formulate appropriate requirements. A flexible 5G architecture design for network slicing, built on SDN and NFV technologies, was presented in [10]. It emphasized schemes which provide effective substrate resource utilization for NS. In [11], the performance deterioration issue of virtualised access points

occurring due to NFV implementation was addressed, and an overcoming approach was presented. Blockchain as an emerging technology has been leveraged to mobile networks in many researches. A blockchain-based secure key management scheme was proposed in [12] to improve the trustworthiness of the base station. The incentive mechanism combining edge computing was addressed in [13, 14]. The blockchain-based collaboration perception and privacy-preserving were studied in [15].

Some typical researches focusing on the decentralized PKI have appeared for years. A blockchain-based PKI framework in mobile networks was proposed in [16]. It focused on the problems when traditional PKI is leveraged into mobile networks. It provided some scenarios and application cases in mobile networks. The optimization for the certificate storage in blockchain-based PKI system was analyzed in [17]. The provided methods are aimed at improving the storage efficiency of specific nodes in blockchain-based PKI system. Research in [18] focused on the trust among multiple CAs (Certification Authority) using blockchain and provided some use cases in mobile networks. The implementation of blockchain-based PKI management framework in [19] used the standard X.509v3 certificate with an addition to the extension fields to indicate its location in the blockchain. The smart contract of each CA contained one list with all issued certificates and another list for revoked certificates. BlockPKI [20] required multiple CAs to perform a complete domain validation from different vantage points for an increased resilience to compromise and hijacking, scale to a high number of CAs by using an efficient multi-signature scheme, and provided a framework for paying multiple CAs automatically. SCPKI [21] worked on Ethereum blockchain and used an entity or authority in the system to verify another entity's identity. It could be used to detect rogue certificates when they are published.

Standard development organizations such as ISO/IEC and ITU-T have begun to study and standardize blockchain-based PKI and certificate management technology. These works are focusing on the profile and the mechanism of blockchain-recorded certificates. However, these normative works are still under development.

Considering that there will be lots of devices in the telecommunication networks which do not support decentralized solutions, the hybrid PKI certificate management supporting traditional and decentralized solutions will coexist in a long time. So, we will focus on the framework and methods proposed in [16, 19], which reuse the standard X.509 certificates [22] and be compatible to traditional solutions. Our innovation is to make the framework compatible to the NFV infrastructure and the certificate management in the operator's domain and make the NFV components support the decentralized PKI certificate management.

3. Issues and Challenges

There are mainly three kinds of certificates use cases in NFV, i.e., VNF certificate use case, MANO certificate use case, and OSS/BSS/EM certificate use case, which had been discussed in [8]. A VNF component instance (VNFCI) needs one or

more certificates provisioned to attest its identity to the VNFM or EM to establish a secure connection between them. During NFV implementation, the number of VNF certificates is far more than that in the other two use cases. The management of VNF certificates will be discussed in this paper. However, the certificates in the other two use cases could use the same method as VNF certificates.

By using traditional solutions, each instance of VNF could enroll certificates to CA/RA (Certification Authority/Registration Authority) directly or by a delegator such as VNFM [8]. However, the issues and challenges are as follows:

3.1. Cost of Certificates. VNFs are implemented with one or more VNF components. While a VNF component instance composed of various VNFCIs could have multiple logical identities, each of which is represented by a certificate, to communicate with different peers [8]. As a result, there will be a huge number of certificates required for the VNFs in 5G networks. It will be costly to use certificates issued by commercial CAs. The telecommunication operators prefer to use their own CA, vendor's CA or designated CAs to provide certificate service due to the cost. This may cause the problem of trust across CA domains.

3.2. Trust across CA Domains. A VNFCI may communicate with another VNFCI in another telecommunication operator's network. These two peers may be equipped with certificates issued by different CAs. The traditional methods to deal with multiple CAs include trusted root list, cross certification, and bridge CA. The trusted root list relies on the list maintained by the relying party. The list is usually preconfigured into the devices. It will be costly to update the list. Cross certification is suitable for a small amount of CAs. If there are a large amount of CAs, the cross relationship will make a complex structure. Besides, the usage of certificate policies will be limited after multiple mappings. The certificate chain of bridge CA will be much longer, and the validation will cost more computing resources.

3.3. CRL/OCSP Unavailable due to Intranet Implementation. The devices, including the network functions of 5G network, deployed in the telecommunication operator's core network cannot access the Internet. It makes CRL/OCSP unavailable for these devices and network functions. Moreover, the telecommunication operator's core network is usually divided into different security domains. These security domains are isolated physically or logically. The entity in one security domain cannot communicate with the entities in another security domain directly. In practice, the CA/RA service and CRL/OCSP services are usually deployed in a different security domain from the telecommunication operator's core network. It means the entities in the core network cannot access the CRL/OCSP services. Unless, the telecommunication operator deploys the CRL/OCSP services in each security domain, which is a complex and costly work.

3.4. Certificate Validation. When a VNFCI issues a certificate from the CA/RA, the identity of VNFCI will be validated by RA. The subject field in the certificate may be an

IP address, FQDN, or other unique identifiers, which is related with the deployment. It is impossible for the RA to validate the subject field, unless an endorsement for the identity in the subject field is provided. The endorsement needs to be provided by some designated administrators. During implementation, there will be kinds of administrators responsible for corresponding identifiers. Thus, the deep cooperation between the RA and the administrators is significant and it makes the certificate validation complicated.

3.5. Certificate Maintenance. The certificate needs to be renewed when the validation period expires. Or else, it will not be trusted by the relying party. In 5G networks, there will be more than thousands of VNF certificates. It has to be ensured each certificate be renewed before it expires and be revoked once it is insecure or the VNFCI is terminated.

The essence of the above issues is the lack of trust among the multiple participants (such as vendors, telecommunication operator, and CA/RAs) during the NFV implementation. A trusted information sharing and endorsement method is necessary to solve the issues. The blockchain is featured as decentralization and tamper resistance. The endorsement and consensus mechanisms in blockchain help to make the information submitted to the participants in the blockchain system be trusted. It provides a decentralized way to solve the issues of the NFV certificate management.

4. Decentralized NFV Certificate Management

4.1. Framework. Figure 2 shows the participants included in the NFV certificate management scenario. The vendors and service providers develop the VNF packages. The packages contain the certificates issued by the vendor. During implementation, these VNFs will be instantiated with new certificates trusted by the operators. The operators are in different trusted domains. In our new framework, we aim to support both traditional PKI solution and decentralized solution. So, CAs will be included in the decentralized framework.

We make some improvements to the blockchain-based PKI framework proposed in [16] and make it more suitable to the NFV environment. The framework consists of submission nodes, validator nodes, and inquiry nodes. The VNF has kinds of identities. The validators usually are unable to validate the consistency of the VNF's identities and identities in the certificate, unless there is an endorsement. The endorsements can be made by the administrators, and then we add endorsers in the framework. A certificate management unit is also added which acts as the submission node. The framework for VNF certificate management is shown in Figure 3.

The VNFCI is the owner of the certificate. The Certificate Management Unit (CMU) works as a client to submit certificates and related information into the blockchain-based NFV certificate management system. The CMU could be a function in NFV architecture, e.g., located in VNFM, and it also could be independent to the NFV architecture.

The endorser is the node to endorse the identity in the submitted certificates. It could be the administrator of the network or the trusted third party (e.g., CA). Only the

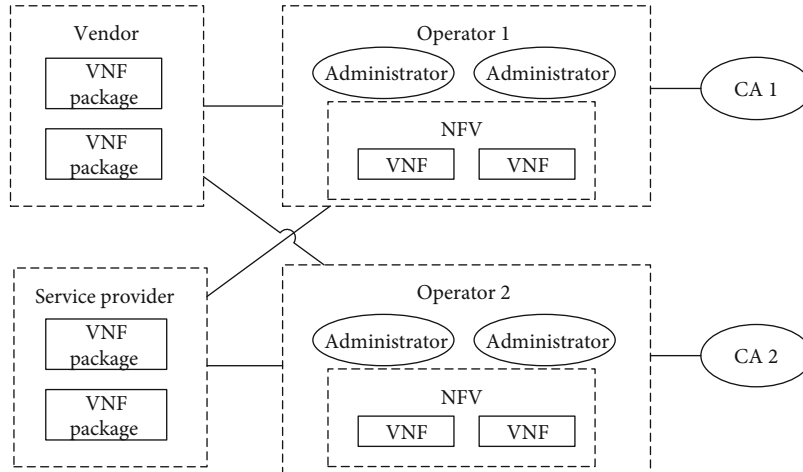


FIGURE 2: Participants included in the NFV certificate management scenario.

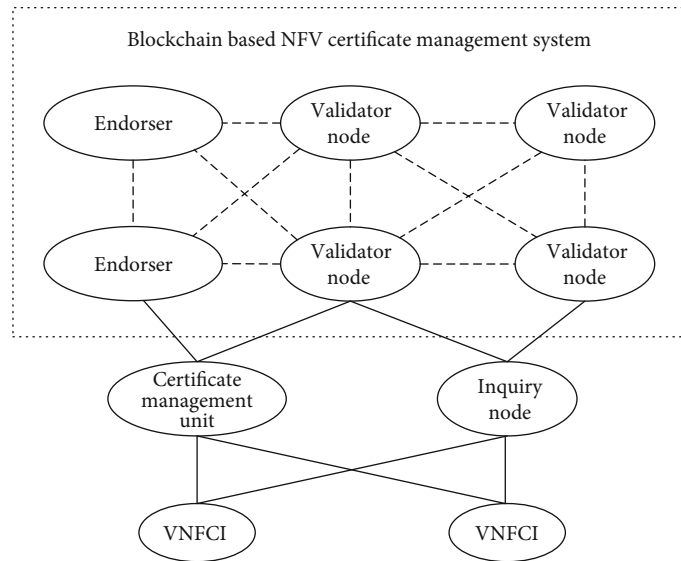


FIGURE 3: Framework for decentralized NFV certificate management system.

endorsed certificates and requests could be processed by the validator nodes.

The validator node is the node to verify the received requests and generate new blocks. It validates the certificates and request according to the policies. The validator nodes are held by vendors, service providers, operators, and CAs. One node could act as both an endorser and a validator.

The inquiry node provides certificate inquiry services. It needs to receive new blocks, but do not need to participate into the generation of new blocks. The inquiry nodes are held and deployed by any party which is capable to access the blockchain-based certificate management system. To support traditional solution, the inquiry node supports OCSP protocol and acts as a proxy. It transmits the OCSP request from the relying party to the corresponding destination and transmits the OCSP respond from the OCSP server to the relying party.

4.2. Certificate Enrolment. During instantiation, VNFCI needs to enroll certificates to communicate with other VNFCI or MANO/OSS/BSS/EM. The certificate could be a certificate issued by CA/RA as described in [8]. It could also be a self-signed certificate generated by the VNFCI. Both of these two kinds of certificates will be discussed in this paper.

The VNF configuration is based on parameterization captured at the design time, included in the VNF package, and complemented during the VNF instantiation. Before a VNF is installed, the VNF package will be on-boarded by NFVO. The VNF package includes a component of VNFD (Virtualised Network Function Descriptor), which is a deployment template describing a VNF in terms of deployment and operational behaviour requirements [23]. The VNFD is a static description file. The metadata in the VNFD is not changed during the whole VNF lifecycle. However, some parameters in the VNFD could be declared to be

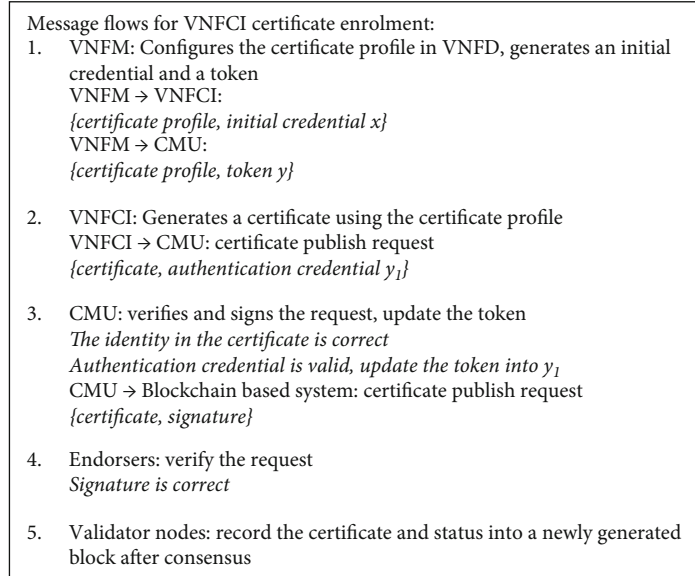


FIGURE 4: Message flows for VNFCI certificate enrolment

configurable during the VNF design phase [24]. We add the certificate profile into VNFD and make it be configurable. During the VNF instantiation, the VNFM accesses to the VNFD and configures the certificate profile. The parameters used to configure the certificate profile could be defined by the administrator. The VNFCI enrolls a certificate as follows, and the message flow is shown in Figure 4.

- (1) The VNFM configures the certificate profile and initial credential for each VNFCI which are included in the VNFD and sends the certificate profile and token for each VNFCI to the CMU

The VNF parameters describing the certificate profile in the VNFD are declared to be configurable during the VNF design phase and be configured by the VNFM during or after the VNF instantiation [24]. The certificate profile declares the information used to generate the certificate, such as the subject, key usage, and basic constraint [25].

The subject field identifies the entity associated with the public key stored in the subject public key field and contains a distinguished name. The distinguished name may be an FQDN, a serial number, or other kinds of names, according to the operator's policy. It is suggested to include the operator's information in the distinguished name field, so as to identify the HPLMN (Home Public Land Mobile Network) in roaming scenarios. Multiple names could be addressed in the SAN (Subject Alternative Name) field [25]. The address of the inquiry node could be included in the extension field of the certificate.

The VNFM sends the certificate profile and a token to CMU. The token and information in the certificate profile will be used to validate the submitted VNFCI certificates. For the sake of simplicity, we use a token which is the value of multiple hash operations on the initial credential. The ini-

tial credential is kept as a secret by the VNFCI. Denote the initial credential by x , the token by y . Then, we have

$$H(H(\dots H(x))) = y. \quad (1)$$

y is the value of multiple times (e.g., n times) hash operations of x .

- (2) The VNFCI generates a self-signed certificate and submits the certificate publish request to the CMU

The public-private key pair used to generate a self-signed certificate is generated using the methods addressed in [8]. The VNFCI generates the certificate using the information and certificate profile provided in the VNFD and then generates the authentication credential based on the initial credential. The authentication credential is the value (denoted by y_1) of multiple hash operations (e.g., $n - 1$ times) on the initial credential (x), of which the hash value equals the token (y). The VNFCI submits certificate publish request to the CMU, while the request consists of the certificate and the authentication credential.

To support the traditional solution, the VNFCI can enroll the certificate via CMU or from RA/CA directly by using protocols such as CMP (Certificate Management Protocol). Then, VNFCI submits the CA-issued certificate in the certificate publish request to the CMU.

- (3) The CMU verifies the certificate publish request, signs the request, and transmits it to the blockchain-based certificate management system

The CMU verifies the certificate in the request to ensure it is consistent with the certificate profile, and the information contained in the certificate is correct (e.g., the information in the subject field is valid). The authentication

credential is verified to ensure it is consistent with the token. Then, the CMU signs the request and submits it to the blockchain-based certificate management system. The token could only be used once so as to protect against replay attacks. Thus, CMU updates the token from y into y_1 . The one-time token makes it possible for the VNFCI to enroll multiple certificates.

If it is a CA issued certificate in the request, the CMU verifies and signs the request and transmits it to the blockchain-based system.

- (4) The endorsers in the blockchain-based system verify the request and endorse the verified request

The endorsers verify the signature of the certificate publish request. After verification, the endorsers sign the request with their own private keys. The endorsement methods are the same to the self-signed certificates and CA-issued certificates. However, the endorsement policy is made and can be configured by the participants. The endorsers can even verify the identity of each VNFCI if necessary.

- (5) The validator nodes record the certificates in the endorsed requests and their statuses into the ledger after consensus

The two kinds of certificates are recorded into the ledger. The inquiry node can inquiry these certificates and their statuses and provide inquiry service to the relying party.

During implementation, the certificates submitted to the blockchain system can be replaced by the certificate hashes. The CMU needs to use certificates hashes in the certificate publish request before it signs the request and submits the request to the blockchain-based system. Then, the size of the transactions will be smaller, and the storage resource requirement will be less, which was discussed in [16].

4.3. Certificate Revocation. A VNFCI certificate needs to be revoked, when it is insecure or the VNFCI is terminated. The VNFCI generates and submits a certificate revocation request to the CMU. The certificate revocation request can be generated by the CMU according to the policy. The certificate revocation request contains the certificate or its identifier, and then it is signed by the CMU.

The CMU submits the certificate revocation request to the blockchain-based certificate management system. The endorsers and validator nodes verify the request and then update the status of the certificate as “revoked” in the ledger. If it is a CA-issued certificate, the CMU will transmit the revocation request to the corresponding RA and forward the response to the VNFCI.

4.4. Certificate Renewal. The certificate to be expired needs to be renewed. The certificate renewal request is initiated by the VNFCI. The CMU could indicate the VNFCI to initiate a certificate renewal process.

The VNFCI generates the certificate renewal request and submits it to the CMU. The request contains the certificate to be renewed or its identifier, the new certificate, and the signature signed by the private key corresponding to the cer-

tificate to be renewed. The CMU submits the certificate renewal request to the blockchain-based certificate management system. The endorsers and validator nodes verify the request and then record the new certificate into the ledger and update the status of the former certificate as “revoked” in the ledger. If it is a CA-issued certificate, the CMU will transmit the certificate renewal request to the corresponding RA and forward the response to the VNFCI.

4.5. Certificate Inquiry. Ideally, the NFV infrastructure of all the telecommunication operators utilize the blockchain-based certificate management solution. However, in practice, some operators may use blockchain-based solution while others use traditional PKI solution. The certificate inquiry is discussed as follows in nonroaming scenario and roaming scenario, in which the VPLMN (Visited Public Land Mobile Network) uses the blockchain-based solution.

- (1) Nonroaming scenario

When a VNFCI receives a certificate from another VNFCI, it inquires the certificate and its status from the inquiry node of the blockchain-based certificate management system. The inquiry node finds the inquired certificate and its status and feedbacks them to the relying party. The relying party verifies the certificate and its status to ensure the certificate is valid. Both the CA-issued certificates and the self-signed certificates can use the same inquiry method. If some VNFCI uses a CA-issued certificate, the OCSP service can be achieved by the inquiry node. The relying party needs to send the OCSP request and receive OCSP response via the inquiry node.

- (2) Roaming scenario

Figure 5 depicts a simplified certificate inquiry architecture in the case of local break out scenario which was defined in [4]. It shows an example of local break out scenario. Usually, each operator only trusts its own system, including the NFV certificate management system. In this case, the VPLMN uses the blockchain-based solution, HPLMN 1 uses the traditional PKI solution, and HPLMN 2 uses the blockchain-based solution which is independent to the VPLMN.

The inquiry node of the VPLMN connects the CRL/OCSP servers used by HPLMN 1 and the inquiry node in NFV certificate management system of HPLMN 2. When a VNFCI in the VPLMN receives a certificate from the VNFCI of another PLMN, it connects the inquiry node in VPLMN for the certificate status. The certificate of this blockchain-based solution contains the operator’s information in the distinguished name filed. The traditional CA-issued certificate contains the CRL/OCSP server’s address. As a result, the inquiry node in the VPLMN connects the CRL/OCSP server of HPLMN 1 and the inquiry node of HPLMN 2, according to the information included in the certificate. The inquiry node of the VPLMN inquires the certificate status and then feedbacks the status to the VNFCI in the VPLMN.

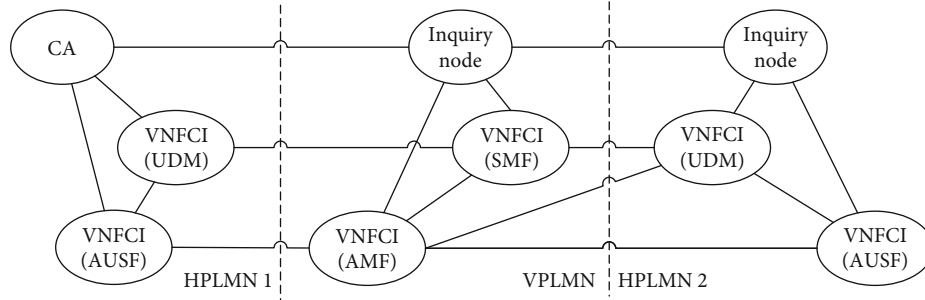


FIGURE 5: Certificate inquiry architecture for roaming 5G system.

5. Performance Evaluation

In this section, we perform experimental measurements to evaluate the performance of our decentralized certificate management framework. And then we make some analysis about the influence to the VNF performance.

5.1. Experimental Setup. We used Hyperledger Fabric to build a blockchain system including 2 organizations and 2 peers per organization. There is one orderer node to provide ordering service. We use the Solo consensus mechanism in these experiments. The peer nodes and orderer node run on dependent physical servers. The Apache JMeter is used to test the performance, which also runs on a physical server. Each physical server has 4 CPUs (Intel Xeon 2.3 GHz) with 16 GB RAM. All physical servers are connected with 1 Gbps network. We used the native Fabric V1.4.6 with no optimization to evaluate the performance. At least 50 times of experiments were made under each circumstance, and the average results were used in the evaluation.

5.2. Transaction Efficiency of Certificate Management. Transaction throughput, which is the number of transactions could be processed in a given time period, determines the efficiency of a blockchain-based system. Figure 6 shows the overall transaction throughput of the decentralized NFV certificate management framework. We set the block interval to 2 s and 0.25 s, respectively, and record the transaction throughput under different block sizes. We found the maximum transaction throughput is around 500 tps, and it changes little when the block size is more than 50. It performs better, however, not significantly, when the block interval is set to 0.25 s. When the block size is more than 1000, the transaction throughput reaches 550 tps.

The certificates can be replaced by their hashes when recorded into the ledger. Figure 7 shows the performance when the certificates hashes are used. In Figure 7(a), the block interval is set to 0.25 s, and we found the transaction throughput exceeds 600 tps when the block size is more than 1000. Then, we set the block size to 3000 and record the transaction throughput under different block intervals, as shown in Figure 7(b). We found it performs better when the block interval is 0.25 s. Generally, the transaction throughput achieves more than 600 tps when the certificates hashes are used.

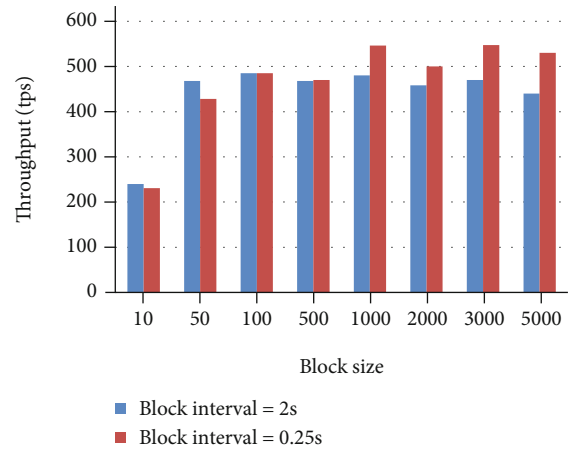


FIGURE 6: Transaction throughput of the decentralized certificate management framework.

5.3. Transaction Efficiency Evaluation. When we evaluate the transaction efficiency of certificate management in NFV environment, we first need to recall the performance benchmark about NFV. The certificate management such as enrolment happens during the initiation of a VNF, and we have to focus on the metrics related to the deployment of VNFs.

The ETSI GS NFV TST 009 [26] specifies vendor agnostic definitions of performance metrics and the associated methods of measurement for benchmarking networks supported in the NFVI. The key metrics are network related such as latency, throughput, delay variation, and loss. In IETF RFC 8172 [27], the metric of time to deploy VNFs is defined. It is the time taken to create 100s of virtual machines and VNFs and make them work properly, in case the general purpose hardware is already deployed. In the work of [28], a similar KPI called deployment process delay is considered. In the process, a service instance is instantiated within the booted virtual machines and setup an operational network services.

In NFV scenario, the certificate enrolment happens during instantiation. It happens once or no more than several times for each VNFCI. Usually, the validity period of a certificate is 1-year long. However, it could be configured according to the operator’s policy. The longer is the validity, and the less certificate renewal is needed. Each certificate can only be revoked once. As a result, the certificate for each VNFCI needs no more than two transactions (certificate enrolment/renewal

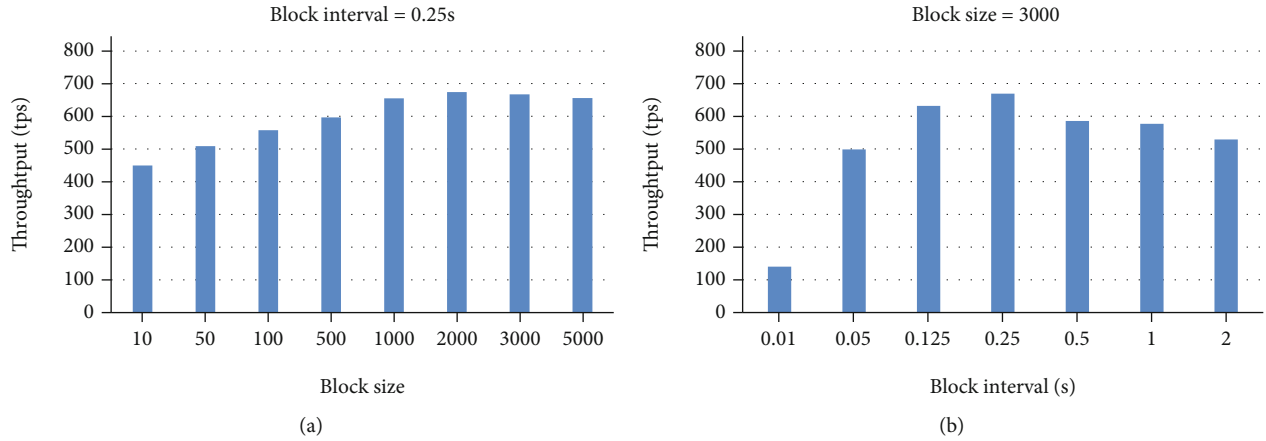


FIGURE 7: Transaction throughput when the certificate hashes are used.

and certificate revocation) per year on average. Even if there are millions of VNFCIs in the operator's network (more than 100s of the VNFs defined in [27]), about 10s of transactions happen per minute. Theoretically, the transaction efficiency is noncritical in this decentralized system. The result of the experiment shows the decentralized framework supports more than 500 transactions per second, which fulfils the requirement defined in [27].

5.4. Transaction Delay. Each certificate management request may result in a new record in the ledger. Transaction delay means the time from the certificate management request submitted to the blockchain-based system to the time that the request be processed and recorded into a new block or be rejected.

We focused on the deployment delay of VNFs defined in [27, 28]. The research in [28] compares the deployment process delay for the two platforms of OSM-4 and ONAP-B. The experiment contains an aggregate of 5 VNFCIs. The result shows the deployment process delay of aggregation level is 134 s. While the deployment process delay of each VNFCI varies from 20 s to 36 s. According to the result of our experiment in Figure 6, we observe the average delay of certificate management is less than 1 s (when the block interval is set to 2 s). It will increase 2%-5% of the deployment process delay of a VNFCI in [28]. And it will increase less than 1% of the aggregate deployment process delay of a VNF.

During the deployment of VNFs in the operator's network, each VNF may contain numeral VNFCIs. It usually takes minutes to instantiate a VNF. So, the delay of seconds is acceptable, even there are several VNFCIs which need to enroll certificates.

5.5. Performance of Certificate Inquiry. In the traditional PKI system, the certificate status is inquired by using CRL or OCSP service, which is a centralized service provided by the trusted third party. In the blockchain-based certificate management system, each node capable to access the ledger could provide certificate status inquiry service. This makes the inquiry service be decentralized. The inquiry performance of each inquiry

node depends on the service and the hardware, which is not related to blockchain. More than one inquiry node can be deployed to enhance the inquiry performance, if necessary. When the inquiry node is deployed on the edge of the operator's core network and Internet, it could provide local certificate inquiry service for the entities in the core network. This may greatly enhance the availability and efficiency of certificate status inquiry service.

5.6. Other Considerations. There are some considerations to address the issues and challenges in clause 3.

- (i) *Cost of Certificates.* There is no need for the operators and vendors to deploy and maintain the PKI infrastructure for the NFV implementation, so the cost is reduced
- (ii) *Trust across CA Domains.* The nodes in the decentralized system consist of operators, vendors, service providers, and traditional CAs, which are in different trusted CA domains. The endorsement and consensus mechanisms make all the records in the ledger be trusted by the multiple participants from different CA domains according to the policy. It makes the trust between different trusted domains be available
- (iii) *CRL/OCSP Service of Intranet Implementation.* The inquiry node can be deployed on the edge between the intranet and the Internet. It provides certificate inquiry service for the devices in the operator's core network
- (iv) *Certificate Validation.* The CMU submits certificates and related information into the blockchain. The endorser, which could be the administrator of the network or the trusted third party, will endorse the identity in the submitted certificates. The CMU and the endorsers work together to endorse the identity and validate the certificate
- (v) *Certificate Maintenance.* The CMU could be used to maintain the certificates. It can be used to indicate

the VNFCI to initiate a certificate renewal process and can be used to revoke certificates

6. Conclusion

Decentralized PKI is a significant direction for PKI technology. This paper analyses the issues and challenges related to the certificate management aroused during the NFV implementation in the telecommunication networks and proposes a blockchain-based decentralized NFV certificate management mechanism. The mechanism could establish the trust among the participants in the NFV implementation, such as vendors, service providers, operators, and even traditional CAs. It could ease the work load of the certificate management, reduce the cost to deploy and maintain the CA, and make certificate status inquiry available in the operator's core network. The experiment and analysis show the performance of transaction efficiency is noncritical and fulfils the requirement in practice. The high performance of the certificate inquiry could be facilitated by the decentralized deployment of inquiry nodes. This work could also facilitate the certificate usage in other scenarios in the telecommunication networks.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

This research was performed as part of the employment of the authors in China Mobile Research Institute. Part of this work has been presented in EAI Mobimedia 2021.

Conflicts of Interest

The authors declare that there is no conflict of interest.

References

- [1] S. Sridharan, "A literature review of network function virtualization (NFV) in 5G networks," *International Journal of Computer Trends and Technology*, vol. 68, no. 10, pp. 49–55, 2020.
- [2] ETSI GS NFV 002, "Network functions virtualisation (NFV); architectural framework," 2014, https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf.
- [3] J. Ordóñez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: concepts, architectures, and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.
- [4] 3GPP TS 23 501, "3rd generation partnership project; technical specification group services and system aspects; system architecture for the 5G system," 2020, https://www.3gpp.org/ftp//Specs/archive/23_series/23.501/23501-g70.zip.
- [5] 3GPP TS 33 501, "3rd generation partnership project; technical specification group services and system aspects; security architecture and procedures for 5G system," 2020, https://www.3gpp.org/ftp//Specs/archive/33_series/33.501/33501-g50.zip.
- [6] T. Hepp, F. Spaeh, A. Schoenhals, P. Ehret, and B. Gipp, "Exploring potentials and challenges of blockchain-based public key infrastructures," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs)*, pp. 847–852, Paris, France, 2019.
- [7] ETSI GS NFV 001, "Network functions virtualisation (NFV); use cases," 2013, https://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf.
- [8] ETSI GR NFV-SEC 005, "Network functions virtualisation (NFV); trust; report on certificate management," 2019, https://www.etsi.org/deliver/etsi_gr/NFV-SEC/001_099/005/01.01.01_60/gr_NFV-SEC005v010101p.pdf.
- [9] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, "NFV and SDN-key technology enablers for 5G networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2468–2478, 2017.
- [10] F. Z. Yousaf, M. Gramaglia, V. Friderikos et al., "Network slicing with flexible mobility and QoS/QoE support for 5G networks," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1195–1201, Paris, France, 2017.
- [11] X. Wang, C. Xu, G. Zhao, and S. Yu, "Tuna: an efficient and practical scheme for wireless access point in 5G networks virtualization," *IEEE Communications Letters*, vol. 22, no. 4, pp. 748–751, 2018.
- [12] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.
- [13] J. Xiong, X. Chen, Q. Yang, L. Chen, and Z. Yao, "A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2347–2360, 2020.
- [14] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, "A blockchain-based machine learning framework for edge services in IIoT," *IEEE Transactions on Industrial Informatics*, 2021.
- [15] J. Xiong, R. Bi, Y. Tian, X. Liu, and D. Wu, "Towards lightweight, privacy-preserving cooperative object classification for connected autonomous vehicles," *IEEE Internet of Things Journal*, 2021.
- [16] J. Yan, X. Hang, B. Yang, L. Su, and S. He, "Blockchain based PKI and certificates management in mobile networks," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1764–1770, Guangzhou, China, 2020.
- [17] J. Yan, B. Yang, L. Su, and S. He, "Storage optimization for certificates in blockchain based PKI system," in *Blockchain Technology and Application (CBCC 2020)*, vol. 1305 of Communications in Computer and Information Science, , pp. 116–125, Springer, 2021.
- [18] J. Yan, J. Peng, M. Zuo, and K. Wang, "Blockchain based PKI certificate system," *Telecom Engineering Technics and Standardization*, vol. 2017, no. 11, pp. 16–20, 2017.
- [19] A. Yakubov, W. Shbair, A. Wallbom, and D. Sanda, "A blockchain-based PKI management framework," in *2018 IEEE/IFIP Network Operations and Management Symposium (NOMS 2018)*, pp. 1–6, Taipei, Taiwan, 2018.
- [20] L. Dykciak, L. Chuat, P. Szalachowski, and A. Perrig, "BlockPKI: an automated, resilient, and transparent public-key infrastructure," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 105–114, Singapore, 2018.

- [21] M. Al-Bassam, "SCPki: a smart contract based PKI and identity system," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pp. 35–40, Abu Dhabi, United Arab Emirates, 2017.
- [22] ITU-T X 509, "The directory: public-key and attribute certificate frameworks," 2019, <https://www.itu.int/itu-t/recommendations/rec.aspx?rec=14033>.
- [23] ETSI GS NFV-IFA 011, "Network functions virtualisation (NFV) release 4; management and orchestration; VNF descriptor and packaging specification," 2020, https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/011/04.01.01_60/gs_NFV-IFA011v040101p.pdf.
- [24] ETSI GS NFV-IFA 008, "Network functions virtualisation (NFV); management and orchestration; Ve-Vnfm reference point - interface and information model specification," 2019, https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/008/02.07.01_60/gs_NFV-IFA008v020701p.pdf.
- [25] IETF RFC 5280, "Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," 2008, <https://datatracker.ietf.org/doc/rfc5280/>.
- [26] ETSI GS NFV-TST 009, "Network functions virtualisation (NFV) release 3; testing; specification of networkixng benchmarks and measurement methods for NFVI," 2020, 2020, https://www.etsi.org/deliver/etsi_gs/NFV-TST/001_099/009/03.04.01_60/gs_NFV-TST009v030401p.pdf.
- [27] IETF RFC 8172, "Considerations for benchmarking virtual network functions and their infrastructure," 2017, <https://datatracker.ietf.org/doc/rfc8172/>.
- [28] G. Yilma, Z. Yousaf, V. Sciancalepore, and X. Costa-Perez, "Benchmarking open source NFV MANO systems: OSM and ONAP," *Computer Communications*, vol. 161, pp. 86–98, 2020.

Research Article

Securing Open Banking with Model-View-Controller Architecture and OWASP

Deina Kellezi, Christian Boegelund, and Weizhi Meng 

Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark

Correspondence should be addressed to Weizhi Meng; yuxin.meng@my.cityu.edu.hk

Received 3 August 2021; Accepted 3 September 2021; Published 21 September 2021

Academic Editor: Ximeng Liu

Copyright © 2021 Deina Kellezi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In 2015, the European Union passed the PSD2 regulation, with the aim of transferring ownership of bank accounts to the private person. As a result, Open Banking has become an emerging concept, which provides third-party financial service providers open access to bank APIs, including consumer banking, transaction, and other financial data. However, such openness may also incur many security issues, especially when the data can be exposed by an API to a third party. Focused on this challenge, the primary goal of this work is to develop one innovative web solution to the market. We advocate that the solution should be able to trigger transactions based on goals and actions, allowing users to save up money while encouraging positive habits. In particular, we propose a solution with an architectural model that ensures clear separation of concern and easy integration with Nordea's (the largest bank in the Nordics) Open Banking APIs (sandbox version), and a technological stack with the microframework Flask, the cloud application platform Heroku, and persistent data storage layer using Postgres. We analyze and map the web application's security threats and determine whether or not the technological frame can provide suitable security level, based on the OWASP Top 10 threats and threat modelling methodology. The results indicate that many of these security measures are either handled automatically by the components offered by the technical stack or are easily preventable through included packages of the Flask Framework. Our findings can support future developers and industries working with web applications for Open Banking towards improving security by choosing the right frameworks and considering the most important vulnerabilities.

1. Introduction

Traditional banks often run their services independently and maintain their own users, while it is hard to obtain the data from other customers. Such data obstacle restricts many services such as product and service innovations and business operation across different banks (e.g., money transfer) [1]. From October 2015, a revised Payment Services Directive (PSD2) has been adopted in Europe aiming to enhance the development and use of innovative online and mobile payments through giving consumers more choice and higher security for online payments in the EU. Open Banking refers to the practice of securely sharing financial data, based on the customer consent. The data exchange between the bank and authorized third parties is enabled via Application Program-

ming Interfaces (APIs). With the radical transformation of financial sector and new regulations, banks are demanded to develop Open Banking APIs that enable the following two properties: (1) securing access to bank account data and information and (2) allowing transactions to be completed among different accounts. As a result, Open Banking is an important sharing data solution with the aim of eliminating barriers to data access while increasing the customers' control over their data.

1.1. Motivation. One of the largest banks in the Nordics, Nordea, released the first version of their Open Banking API in January 2019. We notice that they also released a sandbox version that allows potential third-party providers to access the APIs in a test environment. Online banking

applications are one of the most lucrative targets for attacks, although many have been mitigated through Nordea’s own protocols, i.e., the production APIs require a multistep authentication through nemID (a common log-in solution for Danish Internet banks) when signing up. However, breaking into an application, by gaining access to a user’s password, can give intruders direct access to triggering transactions. The application security itself, on a range of different areas such as data storage, injections, and communication, should therefore be considered very important to mitigate during development, as this can easily result in data breaches.

After more investigation, we find that 3300 developers are currently registered as developers for Nordea Open Banking, but only one product has been realized so far. The adoption of Open Banking exposes data to more actors than ever before, especially new companies and startups, and therefore, also an enlargement of the security risks that the financial industry is facing, with existing risks being increased and new risks being introduced [2]. Further, the threat becomes higher when leveraging applications on a web platform, with possibly insecure protocols that might not be possible on a desktop or phone application.

1.2. Contribution. Due to the complicated process of obtaining a financial license to use actual production data, in this work, we collaborate with Nordea Bank (Denmark) and delimit the problem by using only their sandbox version to develop the solution of triggering transactions based on users’ habits and model the potential risks and threats. In this work, we first identify the background of a technology stack that can be used for development support. Then, we develop a web application that can enable persistent data storage and a high level of security and explain the system architecture and the API communication. The OWASP Top 10 list of the Ten Most Critical Web Application Security Risks methodology is used to investigate the potential threats and risks. Based on the identified threats, we also suggest the integration of Bcrypt algorithm [3] for storage security, which uses a 128-bit salt and encrypts a 192-bit magic value. Our contributions can be summarized as below.

(i) We investigate the Nordea Open Banking APIs by collaborating with Nordea Bank in Denmark, regarding access authorization, account information services, and payment initialization services

(ii) We then design a web application and introduce the system architecture based on the Model-View-Controller architecture (MVC), including three parts such as model, controller, and view. Our approach can handle the API integration through an abstraction layer with the MVC

(iii) To identify potential risks and threats, we use the methodology of OWASP Top 10 with a threat modelling method for categorizing the threats in six different areas, such as threat agents, exploitability, weakness prevalence, weakness detectability, technical impacts, and business impacts

(iv) Our results found that many security threats like Broken Authentication can be handled automatically by the components offered by the technical stack or can be eas-

ily preventable through included packages of the Flask Framework. However, TLS Layer in Nordea’s Open Banking API may cause some crashes with HTTPS

In comparison to the previous study [4], this work provides more information on Nordea’s Open Banking API, such as sequence diagram, access authorization flow, account information, and payment initialization, and introduces the OWASP Risk Rating Methodology in more detail.

The rest of this work is organized as follows. Section 2 introduces the basic background of the Flask Framework, cloud application platform, OWASP Top 10, database management, hashing and salting, and the Nordea’s Open Banking API. Section 3 reviews the related work, and Section 4 details our proposed web application, including the architecture and object relational database. Section 5 identifies and discusses the potential risks and threats of Nordea’s Open Banking API and our proposed application by leveraging the OWASP Top 10 list. We conclude this work in Section 6.

2. Background

In this work, we adopt the microframework for web development, Flask (for Python) to develop our web application. The Flask can mitigate many security threats by default, supplemented by a number of renowned third-party extensions and packages authenticated by the Flask community, which can be customizable according to the demands. It also provides out-of-the-box abstraction layers for communicating with the popular object relational database-PostgreSQL [5] and the cloud application platform-Heroku [6] for deployment.

2.1. The Flask Framework. A Flask application is initialized by creating an application instance through the Flask class with the application package as argument. The web server then passes all received requests from clients, such as web browsers, to this application instance. The logic is handled by using the Web Server Gateway Interface (WSGI) as protocol, through constantly awaiting requests. The framework is compliment with the WSGI server standard [7].

The application instance also needs to know which part of the logic has to run for each requested URL. This is done through a mapping of URLs to the Python functions, which handle the logic associated with a URL. This association is called route between the URL and the handling function, which can be defined by the `@package.route` decorator. The return value of the function is the response that the client received in the form of a template or a redirect.

2.2. Cloud Application Platform. Heroku [6] is one of the first and largest PaaS (Platform as a Service) providers with their Cloud Application Platform. The developer can deploy an application to Heroku using Git to first clone the source code from the developer branch and then push the application to the Heroku Git server. The command automatically triggers the installation, configuration, and deployment of the application. The platform uses units of computing and dynos to measure the usage of service and perform for different tasks on the server. It also provides a large number of

plugins and add-ons for databases, email support, and many other services. Heroku supports PostgreSQL [5] databases as an add-on, created and configured through the command line client.

2.3. Database Management. The Flask puts no restriction on what database packages can be used and supports a number of different database abstraction layer packages. The web application can run on the PostgreSQL database engine supported by the Object Relational Mapping (ORM) and SQLAlchemy. The selection is based on the following different criteria:

(i) *Easy Usage.* Using a database abstraction layer (object-relational mappers ORMs) such as SQLAlchemy provides transparent conversion of high-level object-oriented operations into low-level database instructions, in comparison to writing raw SQL statements [8]

(ii) *Performance.* ORM conversions can result in a small performance penalty, yet the productivity gain far outweighs the performance degradation. The few outlying queries that degrade the performance can be subsidized by raw SQL statements

(iii) *Portability.* The application platform-Heroku can support a number of different database engine choices, where the most popular and extensible ones are Postgres and MySQL [6]

(iv) *Integration.* The Flask includes several packages designed to handling ORMs, such as Flask-SQLAlchemy [9], which includes engine-specific commands to handle connection

2.4. OWASP Top 10. The Open Web Application Security Project (OWASP) [10] is a worldwide organization focused on improving the security of software. The OWASP has identified a list of the Ten Most Critical Web Application Security Risks that can be used for vulnerabilities mapping, which include:

(1) *Injection.* Injection flaws, such as SQL and ORMs, occur when untrusted data is sent to a field as part of a command or query. The attacker's hostile statements can trick the backend into executing unintended commands

(2) *Broken Authentication.* Application functions related to authentication and session management are often missed or implemented incorrectly, allowing attackers to compromise passwords or session tokens, or to exploit other implementation flaws to infer the user's identity

(3) *Sensitive Data Exposure.* Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and personally identifiable information (PII). Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes without encryption

(4) *XML External Entities (XXE).* Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files

(5) *Broken Access Control.* Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unau-

thorized functionality or data, such as other user's accounts or access rights

(6) *Security Misconfigurations.* Security misconfiguration is the most commonly posed issue. This is commonly a result of insecure default or manual configurations, open cloud storage, misconfigured HTTP headers, and error messages or stack traces containing sensitive data

(7) *Cross-Site Scripting (XSS).* XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser, which can hijack user sessions, deface websites, or redirect the user to malicious sites

(8) *Insecure Deserialization.* Insecure deserialization can lead to remote code execution. Even if deserialization flaws do not result in this, it can be used to perform a different number of attacks such as replay attacks, injection attacks, and privilege escalation attacks

(9) *Using Components with Known Vulnerabilities.* Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If one of these is vulnerable and exploited, it can facilitate data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts

(10) *Insufficient Logging and Monitoring.* Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allow attackers to further intrude systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data

2.5. Hashing and Salting of Sensitive Data. When signing up for the application, the user will need to provide a password and load account data, inevitably sensitive information such as account number. Generally, hash algorithm can be used to securely saving passwords and account numbers on the server side.

2.5.1. Hash Algorithm. Hash algorithm refers to a one-way mathematical function that takes data with an arbitrary length and maps it to a fixed length bit string. The purpose of a hash algorithm is to store the sensitive data securely in the database, simultaneously confirming that the provided password or account number is correct. A good hash algorithm should hold the following properties [11]:

(i) *Preimage Resistance.* For a given h in the output space of the hash function, it is hard to find any message x with $H(x) = h$

(ii) *Second Preimage Resistance.* For a given message x_1 , it is hard to find a second message $x_2 \neq x_1$ with $H(x_1) = H(x_2)$

(iii) *Collision Resistance.* It is hard to find a pair of messages $x_1 \neq x_2$ with $H(x_1) = H(x_2)$

There are a number of different hash algorithms, all with different properties. A selected number of hashing algorithms, for instance, the MD5 algorithm or the SHA1 algorithm, are designed to be fast and efficient. This is preferable when messages should be hashed quickly to check for equality.

However, in terms of passwords, account numbers, or other sensitive information, fast hash algorithms are not always optimal. If there is a security breach that allows attackers gaining access to the data in the database, they can quickly be breached using fast hash algorithms. In particular, MD5 is not recommended, as it is unsalted. This can be done with a precomputed lookup table, also known as a rainbow table. Further, these algorithms can be accelerated significantly by using a GPU [3].

On the other hand, slower hash algorithms such as bcrypt initially create a slower run time. However, when it comes to precomputing hashing values, it is much more difficult, as the algorithm is designed to be slower by an order of magnitude. Brute-forcing the data is therefore way more difficult.

2.5.2. Salting a Hash. There is an observation that users may often choose weak passwords due to the long-term memory limitation [12, 13]. For example, top 10,000 passwords are used by 30% of all users [14]. Even if one were to use bcrypt, a slow hash algorithm, passwords can be quickly be compromised via a parallel set of GPUs. Because of this, we need to enhance the password strength. Salting is an effective technique for this purpose, which entails adding a random string to the beginning or end of the password before hashing. In practice, we have to make the password almost impossible to crack with current technology. A short calculation shows that it is infeasible to guess a salt of 12 characters. Even if we constrict passwords to letters only (a-z and A-Z), of which there are 52 characters, we are able to create: $12^{52} = 1.3 \cdot 10^{56}$ different salts. As a result, even with strong computer power, it is infeasible to guess the salt even with the password. If we were able to check 100 billion salts per second, it needs to cost us: $3.17 \cdot 10^{37}$ years to guess it. In comparison, the age of the universe is around $1.38 \cdot 10^{10}$.

2.6. The Nordea Open Banking API. The Nordea API provides access to a number of different endpoints in order to facilitate the connection to the accounts of the user. Some API endpoints must be used in order to authenticate the user before changing the data, while other endpoints involve a number of side effects, i.e., changing the balance on the accounts [15]. We check a list of the relevant endpoints with Nordea Bank (Denmark), and the following are the most crucial ones:

- (i) Access authorization
- (ii) Account information services
- (iii) Payment initialization services

2.6.1. Access Authorization. To leverage the functionality of the API, the Client ID and Client Secret must be obtained. The values can be retrieved by creating a project on the Nordea Open Banking website. The Client ID and Client Secret are parameters that are configured to the client, and they are never exposed to the actual application user. Once the account has been approved, we must obtain an access token in order to gain access to the API.

Figure 1 describes access authorization flow required to obtain the access token. The faded lines describe the OAuth

flow, handling the multifactor authentication [15]. This is not part of the sandbox version of the API and will therefore not be handled in our approach. It describes the authentication flow that would be present in a production environment, i.e., users with actual accounts using applications that require a NemID authentication (NemID users are assigned a unique ID number that can be used as a username in addition to their CPR-Number or a user-defined username).

2.6.2. Account Information. The account information API includes the possibility to check the contents of the different sample accounts in the sandbox version. We can create new accounts, delete current accounts, and add funds to relevant accounts. This can be done by sending a request to the account endpoint [15]. Based on the URL and the type of request, the function will be different as shown in Table 1.

The flow of account information API depends mainly on which type of request is made. Figure 2 describes two scenarios of requests made to the API. Both of them return a response code with the requested information.

2.6.3. Payment Initialization. The payment initialization API provides functionality to create payments directly in the API, moving funds from one account to another [15].

Figure 3 shows the protocol for payments between the two accounts provided in the request. The final response will confirm whether or not the transaction has been made.

3. Related Work

3.1. Web-Based Solution. In the state-of-the-art, there is few work regarding how web applications or the technical stack can integrate with Open Banking APIs. This is due to two main reasons as below:

- (i) The novelty of most of the interfaces, including Nordea's Open Banking APIs
- (ii) The requirements of developers need to be approved by national financial authorities when using the APIs in production

These factors have delimited the pool of possible researchers to only a few authorized third parties or those using the sandbox version. No official paper has dived into integrating with Nordea's Open Banking API as a third-party provider, nor proposed a model for an architectural model or stack that secures bank account information and transaction functionality in a web application. Nevertheless, a lot of work has generally been done in the field of web application security overall, including several models to identify, analyze, and mitigate possible security breaches under a cyber attack [16]. One example is a study in the field of web application security vulnerabilities detection that conducts a security analysis and threat modelling based on the OWASP Top 10 list and threat modelling [17].

The sandbox version of the Nordea Open Banking API was officially released at the beginning of the project in January 2019. During the attempt to generate the access_token for establishing connection before beginning the development of the application, the error codes were limited to

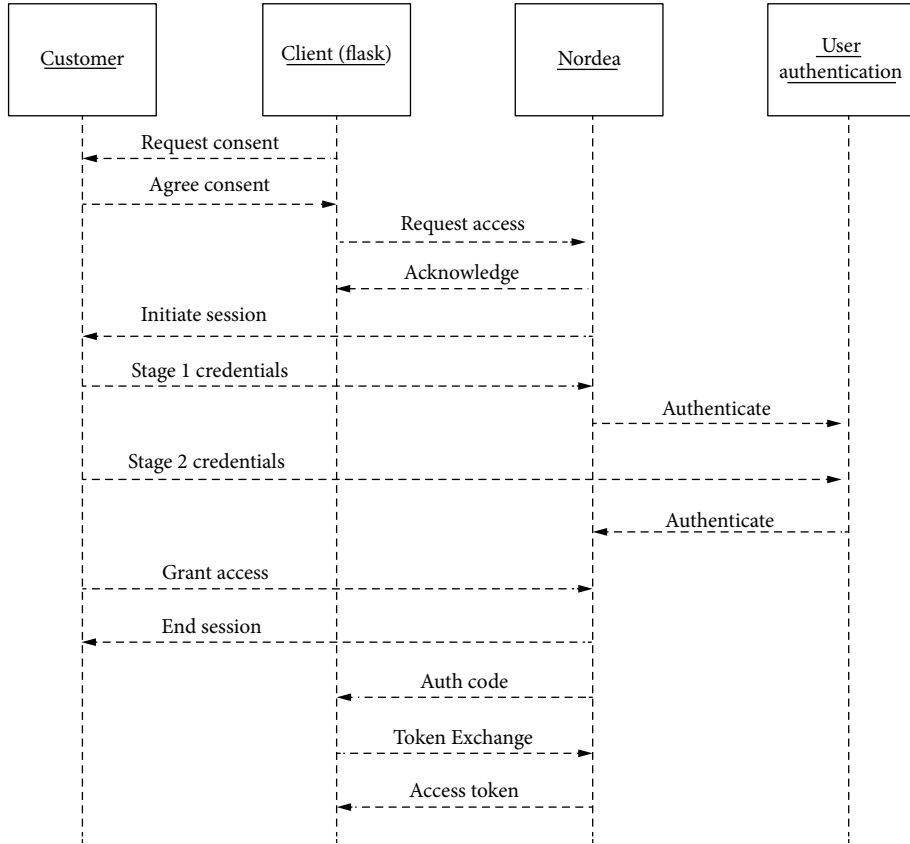


FIGURE 1: Sequence diagram, access authorization flow.

TABLE 1: The request type and the URL with relevant functions.

Type	URL	Function
GET	/v3/accounts	Information about accounts
POST	/v3/accounts	Create new account
POST	/v3/accounts/{ID}	Create transaction

generic server errors. The limited sample codes and lacking documentation on possible error codes (“The error messages are not descriptive at the moment, and this issue is noted. The error messages will be improved over time.”) made it difficult to correct. In order to find a solution, we conducted a simulation with the API simulation tool named Postman [18]. The connection was successful, and the code in Postman worked and did not return any error codes. This led to the conclusion that something was wrong with our implementation of the API calls. To understand the difference between the HTTP packages, the difference between them was negligible. We contacted the senior software architect of Nordea Open Banking. The support team from Nordea Bank tried to assist us in making the API work and assess the possible errors made through logging of their own servers. Ultimately, they did not succeed in resolving the issue. The origin of the error was later found: the redirect URI, a crucial part of the OAuth (OAuth is one of the leading protocols within authentication) 2.0-process was set to an incorrect value.

We thus contributed to the community of developers by using the Nordea Open Banking API and creating a pull request (The PR can refer to: <https://github.com/NordeaOB/examples/pull/7>). At the moment, the sample code only works with version 2 of the API, while the API has been updated to version 3 since then.

3.2. Blockchain-Based Solution. With the advent of blockchain technology, it becomes a popular solution for securing Open Banking. For example, Xu et al. [1] first identified some potential issues of Open Banking, i.e., mutual authentication is hard to be transparently managed, and Access Control List (ACL) controlled by users may pose privacy issues. Then, they introduced PPM, a Provenance-Provided Data Sharing Model for open banking system via blockchain technology, which could employ the programmable smart contracts as the middle witness between users and third-party services to guarantee the reliability and trust communication. Meanwhile, Dong et al. [19] argued that Open Banking may cause a risk of privacy leakage and personal information misconduct. They then introduced BBM, a blockchain-based self-sovereign identity system model, which allowed users to provide their digital identities in the off-line world as same as they use physical identities. Wang et al. [20] also introduced a data privacy management framework based on the blockchain technology, which could be used for Open Banking and the financial sector.

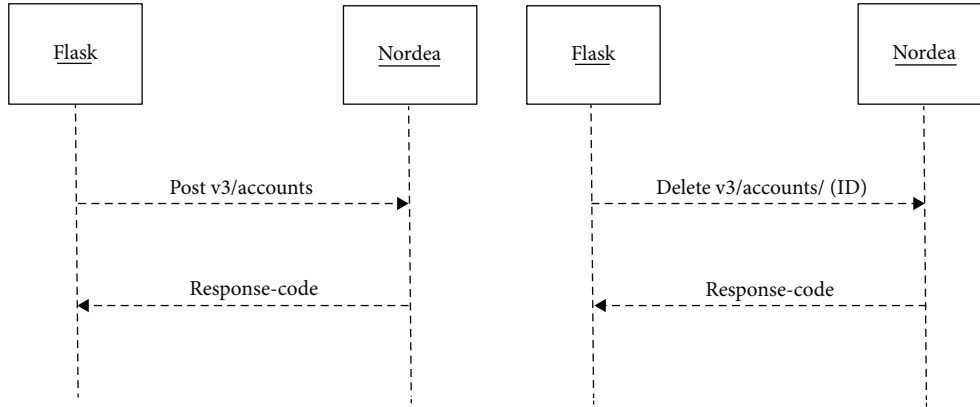


FIGURE 2: Sequence diagram of AIS-API.

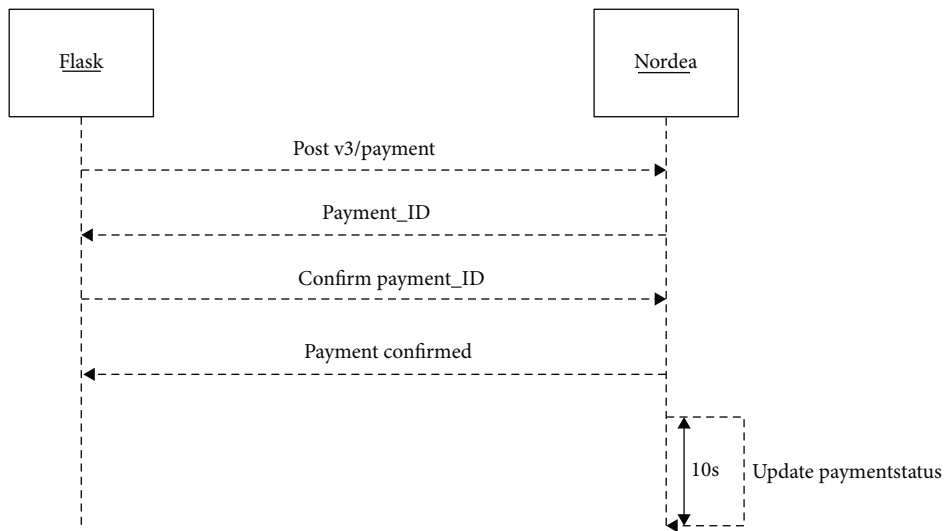


FIGURE 3: Sequence diagram of payment initialization service.

Due to the benefits provided by Open Banking, its data sharing model has been studied in other areas such as Electronic Health Records [21]. Hence, there is a great need to further enhance its security.

3.3. Intrusion Detection Solution. To protect the web application and open banking security, intrusion detection system (IDS) is a basic and necessary mechanism. Based on the detection approaches, it can be identified as either signature-based or anomaly-based detection. For example, an enhanced filter mechanism (EFM) [22] could be used to provide a comprehensive protection, including a context-aware blacklist-based packet filter, an exclusive signature matching component, and a KNN-based false alarm filter. Ma et al. [23] introduced a Distributed Consensus-based Trust Model to evaluate the trustworthiness of IoT nodes, against three typical attacks—tamper attack, drop attack, and replay attack, by sharing certain information. Sohi et al. [24] introduced RNNIDS that could enhance the detection performance by using Recurrent Neural Networks (RNNs) to find complex patterns in attacks and generate

mutants of attacks as well as synthetic signatures. Further, an IDS can work with other security mechanisms towards an enhanced security level.

4. Our Proposed Approach

4.1. The Application Architecture. For defining the architecture, we present a model based on the Model-View-Controller architecture (MVC) specifically adjusted for web development as proposed by Pop and Altar [25]. It was found that developers often combine the HTML code with server side programming languages during the web development and create dynamic web pages and applications. This may lead to highly entangled and unmaintainable code. With an MVC pattern, it is possible to prevent cluttering by separating three overall parts of a web application, including model, controller, and view. The model will also introduce how to handle the API integration through an abstraction layer and how to include it in the MVC.

(i) *Model.* A persistent data storage layer through a data centre or database

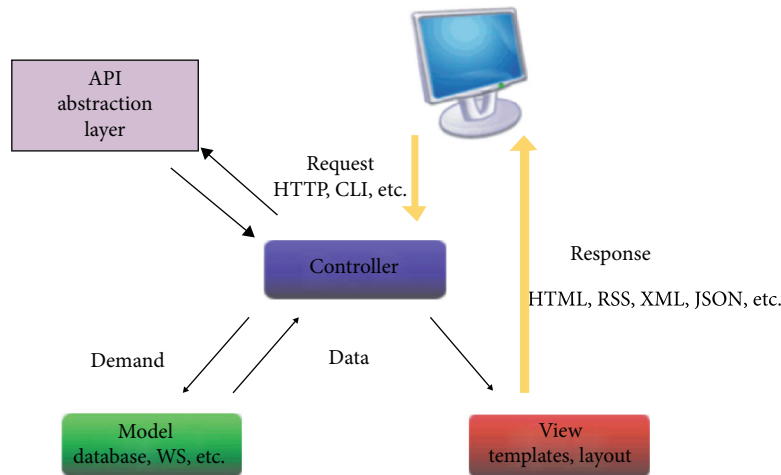


FIGURE 4: MVC architecture for web application.

(ii) *Controller*. The HTTP requests triggered by user actions and general routing of different subpages

(iii) *View*. The HTML code and mark-up languages in the templates rendered to the user as a result of a request

These main components will be built through a modular approach, using blueprints as recommended by the Flask. Figure 4 presents the proposed diagram for the adjusted MVC, which can be further adjusted to include supplementary components for interacting with the API. This model allows us to further propose how this fits into the Flask Framework and an effective abstraction layer integration with the API.

Figure 5 presents the schematic of the application architecture and how the MVC components are implemented.

4.1.1. The Model. As shown in Figure 5, the blue box shows the modelling of the data objects and relationship. This is the direct representation of the schema in the database. Whenever the SQLAlchemy methods start either querying, updating, or deleting data, they are called on the defined data objects in the model, and the database is updated accordingly. This also provides simpler commands for establishing connections to PostgreSQL through the URL of the database as handled by the controller.

4.1.2. The Controller. As shown in Figure 5, the green box shows the controller that is classified into three blueprints:

(i) *auth_controller*. Rendering the pages responsible for signing up and authenticating users logging in

(ii) *main_controller*. Rendering the pages of specific user session, containing URLs for creating habits, checking off habits that are completed, overview over habits, and overview over accounts and settings. This is restricted to authenticated users only

(iii) *admin_controller*. Rendering the pages of administration content included for demonstration purposes that allow to test the different API functionality. This is restricted to users with admin rights only

The blueprints provide a clearer separation of different states in the application, which could be done through appli-

cation dispatching, i.e., creating multiple application objects, however, this would require separate configurations for each of the objects and management at the server level (WSGI). Blueprints instead support the possibility of separation at the Flask application level, ensuring the same application and object configurations across all controllers, and most importantly, the same API access. This means that a Blueprint object works similarly to a Flask application object, but is not an actual application as it is a blueprint of how to construct or extend the application at runtime [26]. When binding a function with the decorator `@auth.route`, the blueprint will record the intention of registering the associated function from the `auth` package blueprint on the application object. It will also prefix the name of the blueprint (given to the Blueprint constructor) `auth` to the function.

Each blueprint handles initialization, routing, and execution in the application. The initialization entails creating a Flask object instance by taking a specific set of configurations for either development, testing, or production environment, establishing connection to the API by obtaining the Client ID and connecting to the database. For the routing, Flask requires us to define routing functions for each of the URL routes for the web application. This allows the Flask to map the incoming request from the user to a specific response, triggering change in the state of the application in the model and rendering the template with the changed data. We have limited the requests to GET and POST methods, following a POST/REDIRECT/GET pattern. Moreover, the controllers are responsible for running the application instance through the main method provided by the Flask.

4.1.3. The View. As shown in Figure 5, the orange box shows the inheritance hierarchy of the templates that primarily consist of HTML and CSS, built upon a number of frameworks. The inheritance is supported by the Jinja2 Template Engine, offered by the Flask, enabling all templates to inherit from a base design, as well as register into their specific controller through the aforementioned blueprints. This also allows dynamic rendering of values provided as argument to the templates when rendered [26].

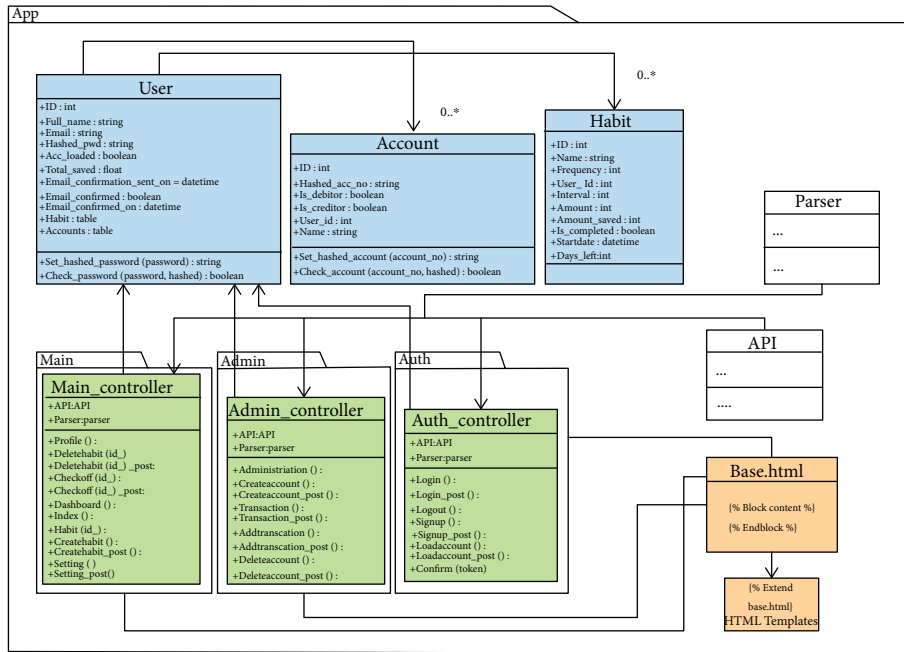


FIGURE 5: Class diagram for MVC.

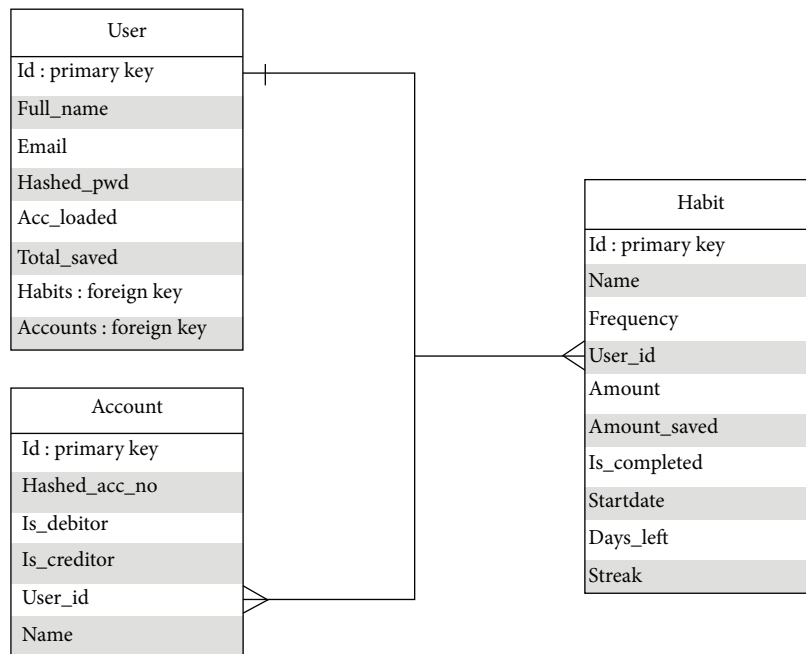


FIGURE 6: Entity-relationship diagram for database.

4.2. *Object-Relational Database.* Ensuring that the application data is stored in an organized and secure way requires a database model. Databases can be modelled in different ways, and we need a model that can effectively represent the following information: users, the individual user’s habits, and the individual user’s accounts. This constitutes an object-relational database [27].

Figure 6 illustrates the entity-relationship diagram for database. It is easy to change the schema for future feature

implementations, which will be relevant for further development of the application. Moreover, it models the entity relations in the application domain in a simple way, i.e., as shown in Figure 6, users that each own a number of habits and a number of accounts, each represented as rows in a table. The tables have a fixed number of columns with the variable names related to the object and a variable number of rows with values. Each table also has a column with a primary key, holding unique identifier for all rows stored in

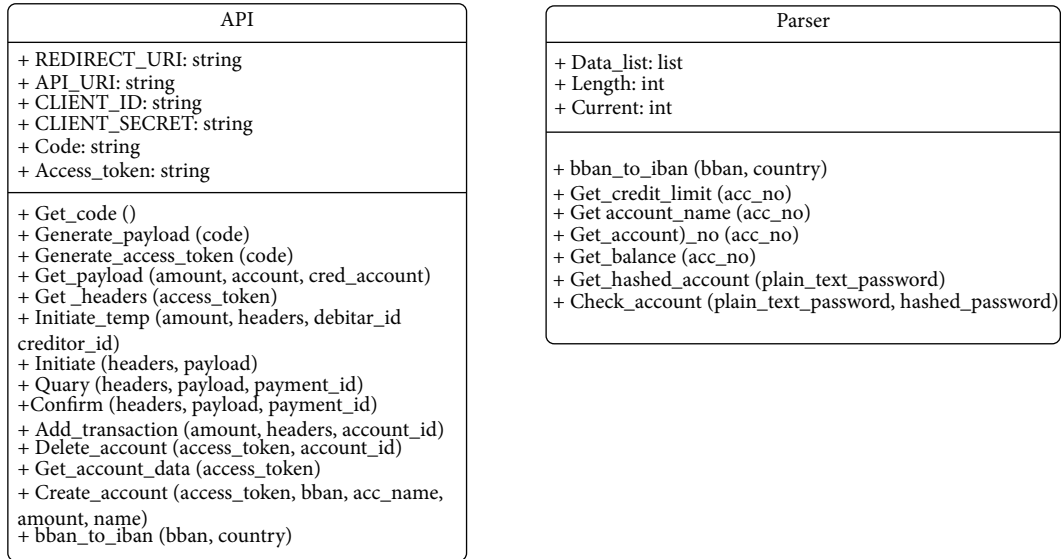


FIGURE 7: Class diagram of API abstraction layer.

that table. The foundation of the relational database model is the foreign keys in the tables that reference the relationship between users and their habits and accounts through lists.

4.2.1. API Abstraction Layer. We present two supplementary APIs and Parser classes to the MVC model. These classes work as abstraction layers for easing communicating with the APIs and filtering out unnecessary data for the application. The purpose is to avoid interacting directly with the API and therefore avoiding unnecessary complexities and errors by encapsulating complex requests in our methods and handling responses accordingly. Figure 7 describes the class diagram of API abstraction layer.

The user's bank and account information can be retrieved directly through the account information services API as a JSON response. The calls to retrieve this response are separated into several methods in the Parser class. The response is first separated into a list object as a field in the Parser and then indexed to extract the needed information. It also consists of different conversion methods to convert different account representations, as well as methods to hash and check account numbers. The API class contains the methods handling the payment initialization service API, hence, triggering transactions between the bank accounts, called whenever habits have been checked off. Through the fields of the API class, we are also able to keep the access token saved across web pages without having to reinstantiate it. Both classes are created as instances for the controller.

4.2.2. Platform Architecture. In order for the application to be deployed in production mode, we propose a platform hosted on the cloud application platform-Heroku [6], with the database connected through Heroku's Postgres add-on.

Figure 8 shows the architecture of the platform where the application is deployed onto. The Flask application itself is as described run through a WSGI server during the development. The application has therefore to be configured to run through HTTP/HTTPS Server instead of running out-

side of the local host. We propose Gunicorn, a WSGI HTTP server, as recommended by Heroku. The application will send the ORM statements to the database through the database driver psycopg2, which is the most popular for the Python language.

5. Evaluation

In this section, in order to investigate the security and effectiveness of our approach with Open Banking, we introduce our evaluation methodology with the OWASP Top 10 and discuss the identified attacks against application integrating with the Open Banking API.

5.1. Methodology. Figure 9 shows the methodology for applying the OWASP Top 10 to the described application and its architecture, which entails systematically going through the list from the most critical to the least critical threat. The OWASP methodology provides a threat modeling method for categorizing the threats in six different areas, which might result in the weighing of threats to change.

Four of these areas are predetermined in the model and should be the basis of the top 10 ranking in the first place. The categorizations for each element in the list can be viewed from the OWASP documentation. However, by observing the two areas of Threat Agents and Business Impact, they can impact how critical a given threat is. If the Threat Agent and/or Business Impact have a low threat level, then, the threat can quickly become irrelevant.

The OWASP provides a comprehensive model for calculating the risk factor of Threat Agents and Business Impacts [28]. However, the limitations imposed by using the sandbox version indicate that we have a nonexisting user base, lacking business context, and problems that arise as a result of using the sandbox to prevent testing some of the factors. It can therefore be difficult to reach feasible estimates of both Threat Agents and Business Impact. The Threat Agents will therefore simply be assumed high across all areas, since the

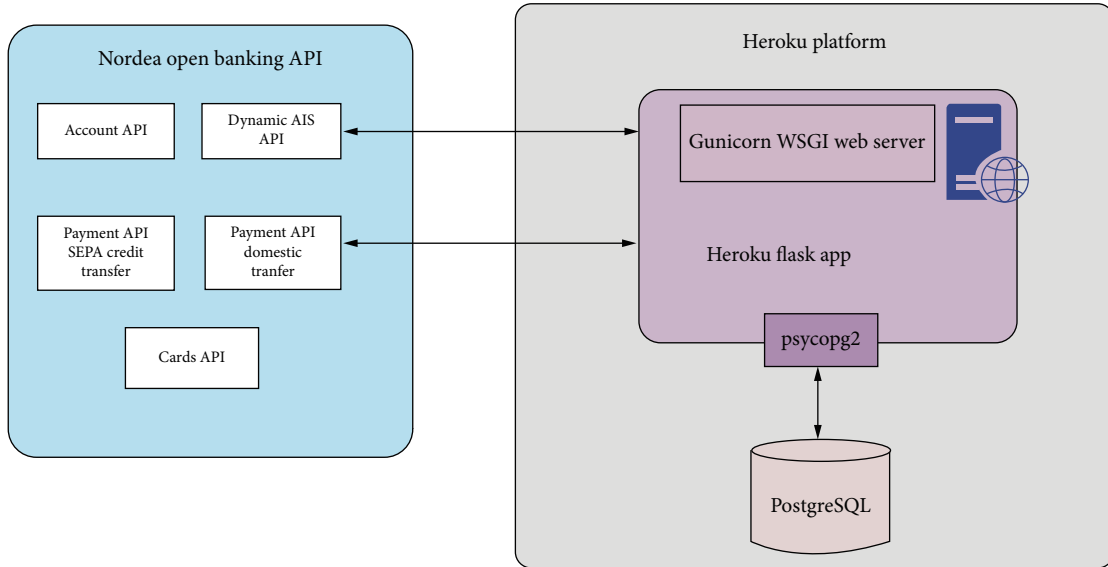


FIGURE 8: Platform architecture.

Threat agents	Exploitability	Weakness prevalence	Weakness detectability	Technical impacts	Business impacts
App specific	EASY: 3	WIDESPREAD: 3	EASY: 3	SEVERE: 3	App/business specific
	AVERAGE: 2	COMMON: 2	AVERAGE: 2	MODERATE: 2	
	DIFFICULT: 1	UNCOMMON: 1	DIFFICULT: 1	MINOR: 1	

FIGURE 9: The OWASP Risk Rating Methodology.

financial industry is generally a critical target due to the possibility of financial rewards. The Business Impact estimation needs to include factors such as financial damage, reputation damage, noncompliance, and privacy violation, data that requires an actual business context. We, therefore, conduct a simple estimate of Business Impact, based on the factors that are critical for the end users and their bank accounts:

(1) *Low*. Security is compromised in areas not containing sensitive data, areas that do not trigger unintentional transactions, or attempted attacks that do not affect the application in any way

(2) *Medium*. Security is compromised such that the attacker gains access to sensitive data in the form of bank data or habits stored in the database

(3) *High*. Security is compromised such that the attacker gains access to functionality using the payment initialization service and can trigger unintentional transactions, leading to either small, substantial, or large financial consequences

Each threat area in the top 10 list will be addressed, with an emphasis on the areas that are estimated as highly for the Business Impact.

5.2. Evaluation on Nordea’s Open Banking API with the OWASP

5.2.1. *Injection and XSS, Threat Agents: 3, Business Impact: 1.* We propose a critical approach regarding user input to prevent injection. A number of tests should be made:

- (i) Input should be filtered
- (ii) Output should be escaped by filtering input

All input fields from the user should be filtered from code-like plain text or injecting raw SQL statements into the database. Submitting unfiltered input into the database can result in a large exposure to SQL injections. This can be detrimental to the privacy of the data; potentially allowing an attacker to access to view the bank information of a user. No further measures need to be proactively taken to prevent injections. ORM SQLAlchemy automatically filters the input of the user, and the Flask Framework automatically escapes output when inserting values into templates, mitigating threats such as JavaScript injection or similar.

5.2.2. *Broken Authentication-Threat Agents: 3, Business Impact: 3.* We propose a number of actions to mitigate broken authentication, as it is one of the most critical threats against the application and the API:

- (i) A set of criteria for the user credentials at sign up
- (ii) Preventing that passwords are saved in plain text

(iii) Using multifactor authentication during either signup and/or login

(iv) A user should only be allowed to enter URLs that they are authenticated to enter

The user is required to provide a username and password at signup, and most applications nowadays provide the possibility of signing up through email. That is, the company is able to authenticate and send information through a mail integration. The username should therefore be a valid email, so we are able to perform multifactor authentication by sending a confirmation email to the address. The Flask-Mail extension provides a simple interface to set up SMTP with your Flask application and to send messages directly from the controller. We also require the password to be at least 10 characters long and include both lowercase and uppercase letters, numbers, and a special sign. Most password breaches occurred as a result of weak password criteria, and setting up a number of requirements for the password is therefore an easy and very effective way of preventing broken authentication.

The authentication can also be broken by gaining access to the database and extracting the plain text version of the password. Therefore, only the hashed password will be stored in the database. The bcrypt hash algorithm, combined with salting, is one of the most effective ways to permit brute force attacks. A salt with a length of 12 characters will result in millions of different combinations, making it almost impossible for an attacker to decode. It does have a larger penalty on the time complexity compared to other hash functions. However, there is still a need to make a trade-off.

The Flask-Login extension provides user session management for Flask and allows us to restrict views through a simple decorator to only authenticated users. The Flask Framework therefore provides an easy way of restricting specific URLs.

5.2.3. Sensitive Data Exposure-Threat Agents: 3, Business Impact: 3. We propose only storing the most important data in the database for the application to run. The remaining data will be exposed during run time from the API response, retrieved by the API abstraction layer. The information stored in the database includes a hashed version of the account number and the name of the account. The rest of the information of that specific account can be retrieved at run time by checking the hashed account number against all the user's accounts in the API. The idea is to keep as much information as possible from an attacker that gains access to the database without compromising functionality.

5.2.4. XML External Entities-Threat Agents: 3, Business Impact: 1. The application accepts no uploads or XML, and therefore, an attack of this nature has no Business Impact. It is therefore not relevant to address.

5.2.5. Broken Access Control-Threat Agents: 3, Business Impact: 3. We propose ensuring that the functionality of the application is only exposed to the specific legitimate user, who is able to check off a number of habits and actions, resulting in automatically transferring funds. It is therefore

necessary to ensure that it is not possible to gain access to this POST request from other sources. For instance, the current user ID in the POST request to the URL would enable an attacker accessing from the outside, since the request could easily be faked. Thus, we need to ensure that it is in fact the legitimate user who performs the check off, and to check the user owning the habit up against the user that is currently in the session. If an attacker is not allowed to check off a habit, but attempts to do it anyway, they are redirected to an error page. We also need to record this attempt in our logging system, which allows us to have an overview of potential security issues and discover possible threat agents.

In order to further strengthen the application, we have implemented protection against Cross-Site Request Forgery (CSRF) with the Flask package CSRFProtect. This is done by adding a hidden field to all forms. This results in the user having to fill out the form on the website in order to have their request accepted, thus, creating a defence against a myriad of automatic scripts. As an additional security measure, CSRF also requires a secret key to sign the token.

5.2.6. Security Misconfiguration-Threat Agents: 3, Business Impact: 2. Misconfiguration can have a number of different sources that can bring disruption to the application, some of which include:

- (i) Revealed stack traces or overly informative error messages
- (ii) Improperly configured permissions
- (iii) Incorrect values for security settings across servers, frameworks, libraries, or databases

We propose using large parts of the security packages and settings offered by the different parts of the technical stack. Flask provides a number of ways to handle custom error messages to the user in order to prevent showing stack traces or overly informative error messages to users. We propose a combination of the following. Message Flashing, that can be included in the templates, is making it possible to record a custom message at the end of a request and access it in the next request and only the next request. The Python logging package also provides the possibility of printing custom messages and stack traces to the console, limiting the information from showing specific request methods and URLs. However, in 2014, as Flask eliminated error and stack traces from application started running in production mode (<https://github.com/pallets/flask/issues/1082>), it is no longer necessary to create custom error messages.

To mitigate improperly configured permissions, the selected cloud service provider will not allow open default sharing permissions to the Internet or other users. This ensures that sensitive data stored within cloud storage is not accessed by illegal users. Heroku PaaS is a large service provider and regular audits with the aim to ensure that permission breaches does not occur.

Lastly, the included Flask packages provide a number of security settings. One example is the Flask LoginManager package, from which it is possible to choose from different levels (none, basic, or strong) of security against user session tampering. The latter ensures that Flask-Login keeps track of the client IP address as well as browser agent during

browsing. If a change is detected, the user will automatically be logged out.

5.2.7. Components with Known Vulnerabilities-Threat Agents: 3, Business Impact: 3. The components we used have no major known vulnerabilities. The Flask Framework is one of the most popular Python microframeworks and therefore has a number of requirements to ensure adequate security. Moreover, the wide community of developers and contributors can ensure that measures are taken to maintain this security level by frequently updating the most popular and renowned packages. The PostgreSQL database [5] is also addressed at several levels:

- (i) *Database File Protection.* All files stored within the database are protected from reading by any account other than the PostgreSQL superuser account
- (ii) Connections from a client to the database server are, by default, allowed only via a local Unix socket, not via TCP/IP sockets
- (iii) Client connections can be restricted by IP address
- (iv) Client connections may be authenticated via other external packages
- (v) Each user in PostgreSQL is assigned with a username and a password
- (vi) Users may be assigned to groups, and table access may be restricted, for instance, through admin privileges

Furthermore, as mentioned previously, there are many problems with the deployment of the application to Heroku PaaS. Heroku is not known to have any known vulnerabilities itself. However, the server routinely crashes in production mode with no useful error messages when enforcing HTTPS on Heroku. We suspect that this is caused by problems with the TLS Layer, with error messages that stem from Nordea's Open Banking API. Hence, we suspect that the errors stem from how the API handles the TLS Layer in the sandbox version. This imposes a high risk for the packages sent between the application and the API to be intercepted. However, no sufficient documentation explains how to mitigate this issue in Nordea's documentation. This will be an interesting topic for future enhancement.

5.2.8. Insufficient Logging and Monitoring-Threat Agents: 3, Business Impact: 2. As mentioned previously, whenever a user attempts to check off the habit, or perform any other actions in the application, of another user, it is added to the log. The log is handled through a logging package offered by the Python library. We propose also including logging for IP addresses and alarms whenever a user is logged in from a different country.

6. Discussion

Applying the OWASP Top 10 Threats and Risk Modelling Framework to our web application and Nordea's Open Banking API shows that it can mitigate a large part of the most critical threats to the application. The threats posed by Broken Authentication, the most critical in terms of Business Impact, is now largely protected from breaches that could cause the user to lose account funds. The same applies

for Sensitive Data Exposure and Broken Access Control that were also categorized as very critical threats.

However, the OWASP framework also exploited that the components with known vulnerabilities posed a high threat to the application, especially Nordea's APIs. The problems with the TLS Layer in Nordea's Open Banking API force us to use HTTP in production mode to avoid the routinely crashes occurred with HTTPS. This means that the packages sent from the API to the application are encrypted. Packages that can contain access tokens, client IDs, or secret keys might give access to Nordea's infrastructure. This vulnerability is impossible to handle without more documentation of the API, since it does not stem from the application itself. Below are some main challenges on open banking security.

- (i) How to securely share data when transforming the relationship between customers and banks
- (ii) How to transparently manage mutual authentication
- (iii) How to secure data privacy when enabling users to control and share personal data by customizing the Access Control List (ACL) [1]

7. Conclusion and Future Work

Currently, Open Banking has received much attention, which refers to the process of using APIs to open up consumers' financial data to third parties. This concept is believed to be secure by enforcing that only the customer and data owner can authorise any connection between the bank and a regulated third party. However, such openness may also incur some kind of security issue. In this work, we proposed a technical stack and an architectural model that can easily integrate and secure Nordea's Open Banking API. In the evaluation, we applied the OWASP Top 10 threats and threat modelling methodology to identify the most prevalent threats regarding the application data and the functionality of the APIs.

The results showed that many of these security measures were either handled automatically by the components offered by the technical stack or were easily preventable through included packages of the Flask Framework. However, it also shows that the application faces a high risk due to the compromised handling of the TLS Layer in the API, causing the production server to routinely crash when using HTTPS. These risks may propagate upwards in the architecture, resulting in high risks for the user's account data and funds. Since the server loggings show that the errors stem from the API itself, it is most likely not due to the choices of any of the cloud application platform, packages, libraries, database, or frameworks. It is also found that adding an API abstraction layer can facilitate the communication when developing the API, and that it can be implemented as a modification to the MVC for web applications.

For future work, we plan to keep gaining more data and information on the TLS Layer handling by cooperating with the support team from Nordea Bank in Denmark, especially the sandbox documentation. With more Open Banking Team code samples, we believe it can help make more practical contributions to the documentation. In addition, it is another interesting and important direction to apply other

threat models to examine the threats and risk compared with the current results with OWASP.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This is a full version of our paper published in Proc. 13th International Conference on Network and System Security (NSS), pp. 185–198, 2019 [4]. To differentiate our current version from the previous one, we have improved the article in the following aspects. (1) In the conference version, there are mainly two diagrams, while in this version, we have 9 diagrams to describe our approach in more detail. (2) In addition, in this full version, we provide more information on Nordea's Open Banking API, such as sequence diagram, access authorization flow, and account information. (3) In this full version, we also provide more information and details on the OWASP Risk Rating Methodology, which can facilitate readers to understand how this methodology works. The authors would like to thank the help from Nordea Bank in Denmark, and Weizhi Meng was partially supported by H2020 CyberSec4Europe under project no. 830929.

References

- [1] Z. Xu, Q. Wang, Z. Wang, D. Liu, Y. Xiang, and S. Wen, "PPM: a provenance-provided data sharing model for open banking via blockchain," in *Proceeding of ACSW*, pp. 1–8, Melbourne, Australia, 2020.
- [2] S. Kiljan, K. Simoens, D. D. Cock, M. C. J. D. van Eekelen, and H. P. E. Vranken, "A survey of authentication and communications security in online banking," *ACM Computing Surveys (CSUR)*, vol. 49, no. 4, pp. 1–35, 2017.
- [3] N. Provos and D. Mazieres, "A future-adaptable password scheme," *Proceedings of USENIX Annual Technical Conference*, pp. 81–91, FREENIX Track, 1999.
- [4] D. Kellezi, C. Boegelund, and W. Meng, "Towards secure open banking architecture: an evaluation with OWASP," in *Proceedings of the 13th International Conference on Network and System Security (NSS)*, pp. 185–198, Sapporo, Japan, 2019.
- [5] PostgreSQL, *The world's most advanced open source database*<https://www.postgresql.org/>.
- [6] Heroku: *Cloud Application Platform*<https://www.heroku.com/>.
- [7] Pallets Team, *Flask's Documentation*<http://flask.pocoo.org/docs/1.0/>.
- [8] The SQLAlchemy authors and contributors 2019 <https://docs.sqlalchemy.org/en/13/>.
- [9] Pallets Team 2010 <https://flask-sqlalchemy.palletsprojects.com/en/2.x/>.
- [10] OWASP, *Top Ten Web Application Security Risks*<https://owasp.org/www-project-top-ten/>.
- [11] P. Rogaway and T. Shrimpton, "Cryptographic hash-function basics: definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance," *Proceedings of FSE*, pp. 371–388, Springer, 2004.
- [12] W. Meng and Z. Liu, "TMGMap: designing touch movement-based geographical password authentication on smartphones," *Proceedings of the 14th International Conference on Information Security Practice and Experience (ISPEC)*, pp. 373–390, Springer, Cham, 2018.
- [13] W. Meng, L. Zhu, W. Li, J. Han, and Y. Li, "Enhancing the security of FinTech applications with map-based graphical password authentication," *Future Generation Computer Systems*, vol. 101, pp. 1018–1027, 2019.
- [14] M. Burnett, *10,000 Top Passwords*<https://xato.net/10-000-top-passwords-6d6380716fe0>.
- [15] Nordea Open Banking Team, 2019, <https://developer.nordeaopenbanking.com/app/documentation?api=Accounts%20API>.
- [16] A. Sapan, B. Oztekin, E. Unsal, and A. Sen, "Testing OpenAPI banking payment system with model based test approach," in *2020 Turkish National Software Engineering Symposium (UYMS)*, Istanbul, Turkey, 2020.
- [17] S. Rafique, M. Humayun, B. Hamid, A. Abbas, M. Akhtar, and K. Iqbal, "Web application security vulnerabilities detection approaches: a systematic mapping study," in *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. 469–474, Takamatsu, Japan, 2015.
- [18] Post Learning Center https://learning.getpostman.com/docs/postman/api_documentation/intro_to_api_documentation/s.
- [19] C. Dong, Z. Wang, S. Chen, and Y. Xiang, "BBM: a blockchain-based model for open banking via self-sovereign identity," in *International Conference on Blockchain*, pp. 61–75, Springer, Cham, 2020.
- [20] H. Wang, S. Ma, H. N. Dai, M. Imran, and T. Wang, "Blockchain-based data privacy management with nudge theory in open banking," *Future Generation Computer Systems*, vol. 110, pp. 812–823, 2020.
- [21] A. Stranieri, A. N. McInnes, M. Hashmi, and T. Sahama, "Open banking and electronic health records," in *2021 Australasian Computer Science Week Multiconference*, Dunedin, New Zealand, 2021.
- [22] W. Meng, W. Li, and L. Kwok, "EFM: enhancing the performance of signature-based network intrusion detection systems using enhanced filter mechanism," *Computers & Security*, vol. 43, pp. 189–204, 2014.
- [23] Z. Ma, L. Liu, and W. Meng, "Towards multiple-mix-attack detection via consensus-based trust management in IoT networks," *Computers & Security*, vol. 96, article 101898, 2020.
- [24] S. M. Sohi, J. P. Seifert, and F. Ganji, "RNNIDS: enhancing network intrusion detection systems through deep learning," *Computers & Security*, vol. 102, p. 102151, 2021.
- [25] D. P. Pop and A. Altar, "Designing an MVC model for rapid web application development," *Procedia Engineering*, vol. 69, pp. 1172–1179, 2014.
- [26] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, O'Reilly, California, USA, 2014.
- [27] IBM Informix, 2011, https://www.ibm.com/support/knowledgecenter/hu/SSGU8G_11.50.0/com.ibm.gsg.doc/ids_gsg_416.htm.
- [28] The OWASP Foundation, 2017, https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology.

Research Article

Efficient Authentication for Internet of Things Devices in Information Management Systems

Xiaofeng Wu ¹, Fangyuan Ren ^{2,3}, Yiming Li ², Zhenwei Chen ², and Xiaoling Tao ⁴

¹School of Management, Xi'an Jiaotong University, Xi'an 710049, China

²School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

³Du Xiaoman Financial, Beijing 100089, China

⁴Guangxi Cooperative Innovation Center of Cloud Computing and Big Data, Guilin University of Electronic Technology, Guilin 541004, China

Correspondence should be addressed to Fangyuan Ren; rfyren@163.com

Received 22 March 2021; Accepted 7 July 2021; Published 19 July 2021

Academic Editor: Ximeng Liu

Copyright © 2021 Xiaofeng Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of the Internet of Things (IoT) technology, it has been widely used in various fields. IoT device as an information collection unit can be built into an information management system with an information processing and storage unit composed of multiple servers. However, a large amount of sensitive data contained in IoT devices is transmitted in the system under the actual wireless network environment will cause a series of security issues and will become inefficient in the scenario where a large number of devices are concurrently accessed. If each device is individually authenticated, the authentication overhead is huge, and the network burden is excessive. Aiming at these problems, we propose a protocol that is efficient authentication for Internet of Things devices in information management systems. In the proposed scheme, aggregated certificateless signcryption is used to complete mutual authentication and encrypted transmission of data, and a cloud server is introduced to ensure service continuity and stability. This scheme is suitable for scenarios where large-scale IoT terminal devices are simultaneously connected to the information management system. It not only reduces the authentication overhead but also ensures the user privacy and data integrity. Through the experimental results and security analysis, it is indicated that the proposed scheme is suitable for information management systems.

1. Introduction

With the advancement of various wireless mobile network technologies, the field of Internet of Things (IoT) has developed rapidly. IoT is connected by multiple smart physical devices through the Internet. The IoT is used in many different fields, such as smart homes, smart cities, smart health, Internet of Vehicles, and information management systems (IMS). In IMS, IoT devices serve as an information collection and exchange unit. The IoT device plays an important role in connecting users and systems so that they can interact. Furthermore, the IMS requires a large amount of information transmission and management. However, these IoT devices send and receive highly sensitive data regarding the privacy of users or other information regarding the movement of users from one location to another location [1]. Therefore,

the primary problem is to solve the efficiency and security of identity authentication in the system. In the field of information and communication technology, the IMS needs a systematic model that contains multiple information processing units to realize. The current development of Internet and wireless network technology has brought us various convenient network services, but at the same time, it has also brought many new security threats. For example, the intrusion of the Internet system leads to information security leakages and other related incidents, which have caused various enterprises in different fields to attach great importance to the security of IMS. In an IMS, users, IoT devices, computers, and servers make up the various parts of the system. These components are used to complete information processing operations such as access, collection, storage, and transmission of information. The use of IMS

enables information to be systematically carried out in batches and secure operations, thereby improving work efficiency. Since the information is transmitted in the wireless network environment, the user's identity information and the content of the message will be exposed on the network. Therefore, the system also has some security problems. Attackers can use the loopholes in the IMS to illegally invade the system, steal, tamper with, and destroy confidential information. For example, an attack on an enterprise's IMS will cause unpredictable losses to the enterprise. Hence, privacy protection is particularly important. The security requirements of the IMS are listed below.

- (1) Confidentiality: to protect information from eavesdropping by illegal users to prevent passive attacks
- (2) Completeness: to protect information content from being illegally tampered with and ensure that the system is not subject to malicious tampering, sabotage, and other active attacks
- (3) Nonrepudiation: the sender and receiver of the information cannot deny the fact that they have sent or received the information
- (4) Reliability: ensure that the system or server will not be illegally interfered, faked, and affected by other deceptive behaviors for the normal operation of the system
- (5) Availability: ensure that all authorized users can access the information management system normally without denial of service attacks

Therefore, given the information security of the IMS, the mutual authentication between the user and the server must be performed first before the user accesses the system. After both parties have passed the authentication, the access and transmission of information in the system can continue to be allowed. Some elliptic curve cryptography- (ECC-) based certification schemes have been used in the IMS of an enterprise. For example, an authentication protocol based on the elliptic curve discrete logarithm problem (ECDLP) [2] was proposed. However, this scheme has the defect that cannot resist tracking attacks and forgery attacks. Then, Islam et al. [3] proposed an advanced scheme based on ECDLP, which has made improvements to the previous problems, and it can effectively resist tracking attacks. However, this scheme needs to update the database during the identity authentication phase, which increases the cost of the back-end server and does not have the feature of mutual authentication. Therefore, there is an urgent need for a secure data transmission and authentication scheme that can guarantee user privacy in IMS. Users' operations such as accessing data information in the IMS are usually performed by connecting smart terminal devices to the network, such as mobile phones, computers, and other IoT devices. Hou et al. [4] proposed a novel blockchain-based architecture for IoT data sharing systems. For the IoT, user access control becomes crucial because of the characteristics of the IoT. To address this issue, Shobhan et al. [5] proposed a new three-factor

certificateless-signcryption-based user access control for the IoT environment. For different wireless network technologies and application scenarios, the security issues faced are different. In terms of 5G security research, the Third Generation Partnership Project (3GPP), the 5G Infrastructure Public Private Partnership (5G PPP), the Next Generation Mobile Networks (NGMN), the International Telecommunication Union (ITU-2020) promotion group, Ericsson, Nokia, and Huawei also released their own 5G security requirements white papers [6–10]. Today, with the gradual development and popularization of 5G network technology, IMS can also run on 5G networks. In the 5G environment, problems such as the disclosure of user identity information and the exposure of data to relatively open channels due to big data. Thus, secure data transmission under the 5G network has become one of the research hotspots since the development of the fifth-generation communication technology.

With the promotion and commercial application of 5G communication technology by the three major telecommunication operators, people's demand for mobile intelligent devices increases. The computing power and storage capacity of smart mobile devices are limited. When the cost of authentication process is large, they are often unable to calculate the complex authentication process. In the process of authentication, some data such as location data needs stronger protection. Once these data are leaked, it may cause great loss [11]. In some application scenarios, fine-grained access control and the identity-based encryption are urgently needed [12]. In another application scenarios, intelligent mobile devices need to switch authentication frequently. Therefore, a more rapid and secure authentication process is urgently needed. With the development of cloud computing and cloud storage technology, the authentication process of intelligent mobile devices can also be completed by relying on cloud computing technology to improve the authentication efficiency [13]. In addition, the traditional authentication mode is not suitable for equipment to equipment authentication, which can achieve the security of end-to-end authentication and reduce the need of computing cost ripple [14]. In the application scenario of unstable network or no network, offline authentication can improve the reliability of device authentication.

Due to the 3GPP 5G network has the characteristics of high capacity and low transmission delay, it has the advantages of high energy saving level, high efficiency, and relatively low expense. Access to the 5G network environment brings convenient network services, but it also creates more security challenges. These can just meet the user's requirements for transmission message delay and service quality in IMS. Now, 5G has become the focus of more and more researchers [15, 16]. By introducing RUSH, Zhang et al. [17] proposed a robust and universal seamless handover authentication scheme for 5G heterogeneous networks. In RUSH, it introduces the blockchain technology [18] and chameleon hash function to realize an anonymous authentication key protocol for handover in various scenarios.

With the advent of the era of intelligent information society, users' demands are also changing constantly. In order to meet various demands, the IoT technology has been

constantly developed and has become more closely connected with people's life. When each user accesses information in an IMS, one or more IoT devices are usually connected to the network to send or receive messages. It has become a trend that more intelligent terminal devices are designed to provide a range of services that need to be achieved by connecting to the network. The IMS under the 5G network will support simultaneous access by a large number of users and devices without causing the current system crash when multiple users access at the same time. IMS access to 5G will not only greatly increase users' access efficiency but also provide security to protect the user's identity information from being leaked. At the same time, it also prevent illegal attacks during the transmission of massive information. The 5G security mechanism should not only ensure the security of massive access devices but also ensure that the information of users will not be leaked when they interact with the network in the scenario of IoT device access. The function of these IoT terminals is generally to collect sensitive data and usually to transmit it. When users need to access an IMS, these IoT devices serve as a medium for transmitting requests and receiving information. Once the data is leaked, it will not only bring huge losses to users but also seriously affects the 5G network. In addition, if large-scale terminals access the network at the same time and the network authenticates each terminal one by one. It will make the authentication cost too high, the network is difficult to bear, and its authentication efficiency will also be unsatisfactory. The actual identity of the user needs to meet certain anonymity under specific scenario requirements. Hence, data privacy and security are particularly important during access authentication and data transmission. Therefore, in most of the technology research especially those related to 5G security access authentication technology, both communication and security requirements should be considered. On the premise of ensuring communication performance, considering the massive access terminals of 5G network and the diversification of security threats, different security access authentication schemes should be adopted.

2. Related Work

In the Long-Term Evolution-Advanced (LTE-A) networks, many protocols are formulated for access security issues [19–27]. In addition, many researchers are paying attention to security of IoT deployment under the 5G network or some other advanced architecture [28–31]. According to the research findings, the current research on a large number of equipment access authentication process in the network, and these schemes are mainly categorized into the following two types.

(i) Group-based security context transformation

Through this type of scheme, many researchers have proposed some group-based access authentication schemes [19–23]. Based on the problem of a large number of users roaming to the same service network when receiving services, Chen et al. first proposed such a group access

authentication and key agreement scheme [19]. In the IoT scenario, in order to ensure that information is not leaked and safe, authentication is required. In practice, however, we usually need to process information from multiple IoT devices at the same time. Obviously, one-to-one certification has great limitations in terms of timeliness and complexity. Therefore, we need to perform group authentication [20, 21]. The access authentication process of SE-AKA Scheme [22] and EG-AKA Scheme [23] is similar to Lai's scheme, and temporary group key is used to realize local identity authentication. These schemes can reduce the cost of high communication costs between home network and service networks by simplifying the process when dealing with the authentication of other group members. However, they are still unable to avoid signaling congestion since they still need to send multiple access request messages to connect to the network.

(ii) Group-based aggregation authentication

In group-based aggregation authentication scheme, a large number of devices are first combined to create a group, and a group leader is selected at the same time. When multiple members from an IoT group need to access the network at the same time, they all issue an access request message. The group leader then gathers the messages of these group members into an access request message and sends them to the network. The verifier in the network then validates the aggregated signature message, thus validating the entire group of devices or aggregate message authentication code generated by the group leader. In Cao's scheme, a group-based aggregate signature authentication scheme is proposed for the first time [24]. Whereafter, a lightweight packet protection protocol based on aggregated message authentication codes is proposed by Lai et al. [25]. Based on secret sharing technology, Li et al. proposed a new group-based protocol with dynamic policy update [26]. Through aggregation technology, Cao's scheme [24] and Li's scheme [26] made great optimizations in terms of communication and signaling overhead. However, both of these schemes may generate a lot of computational overhead. Basudan et al. proposed a protocol [27]. This protocol is a data security transfer protocol based on fog computing and also has the attribute of privacy protection. This scheme can not only make the signaling cost low but also ensure the authenticity and confidentiality of the design. However, derived from the protocol by introducing bilinear pairing operations, a large amount of computing cost is caused. In the case of limited equipment power, they are not suitable. Lightweight authentication in Lai's scheme can be achieved by using symmetric cryptography. However, due to the existence of internal forgery attack, there are still many security vulnerabilities such as DoS attack and lack of identity privacy protection security issues in LGTH scheme. Aiming at these problems, Zhang et al. proposed a multiparty authentication scheme [28]. This scheme adopts certificateless signcryption authentication technology to solve the problems in the multidevice access scenario. It not only realizes the access authentication of multiple devices but also achieves the characteristics of protecting user

identity privacy and nonrepudiation. But this scheme cannot realize mutual authentication between the user and the server. Moreover, there is the problem of a huge number of messages in the network, which easily causes network congestion. Therefore, the authentication overhead and signaling overhead are relatively large.

These two approaches mentioned above still have some issues with both performance and security issues, although they can reduce signaling overhead to some extent. In addition, these schemes do not address the process of secure data transfer, but simplify the process of access authentication.

2.1. Our Contributions. An efficient and secure authentication protocol for IoT devices in IMS is proposed in this paper in which a CS is introduced for file management. Our main contributions are summarized as follows:

- (i) Considering the current development status of wireless communication technology and the mobility and efficiency of most IoT devices, we decided to connect IMS to the 5G network. We will perform our protocol between IoT devices and AMF to achieve mutual authentication. After the mutual authentication process, data can be transmitted in a secure manner under 5G-based IMS
- (ii) Our scheme is to build an IoT group. The leader of the group will aggregate the messages of the legitimate group members and send them to the network, which greatly reduces the number of messages sent to the network and effectively avoids network congestion in IMS. All IoT devices need to be registered on the network before the device is connected to the system. In this stage, the proposed scheme introduces a group leader. It can also realize that AMF communicates to each IoT member device in the group through the group leader device (GLD)
- (iii) Data transmission and authentication are carried out under the premise of ensuring the security and integrity of data. In our scheme, the method of certificateless aggregation and signcryption is adopted. And the group session key (GSK) is used to encrypt messages between the network and the IoT group
- (iv) Mutual authentication between terminal and network will be implemented in our scheme. It can ensure not only the legality of the terminal accessing the network but also the authentication of the network, and the server is realized
- (v) The network inspects the legitimacy of the entire IoT group and the integrity of the transmitted data through aggregate signcryption, which significantly improves the authentication efficiency. The security analysis shows that the scheme can resist security threats such as replay attack and forgery attack. The performance analysis indicates that the scheme is better than the existing schemes in signaling cost, computing cost, and communication cost when fac-

ing the massive IoT devices and can take into account the security and efficiency

2.2. Organization. The following arrangement of the paper is shown below. In Section 3, we elaborated an overview of the system model and relevant requirements. In Section 4, we give a comprehensive overview of the scheme proposed in this paper. The security analysis part and the performance evaluation part are, respectively, described in detail in Section 5 and Section 6. Finally, the conclusion and future work are given in Section 7.

3. Model and Security Requirements

3.1. System Model. When designing the system model, it is necessary to consider the actual needs of communication, user terminal, and network communication. In addition, timeliness is also critical for communication. An IMS usually consists of an authentication server, a confidential server, a file server, and a client. When a user logs in and accesses the file system through a client running on a personal computer, it must first pass the authentication of the authentication server. In a system with high confidentiality, file information also needs to be encrypted by a confidential system. The most important thing in the process of the system running in the 5G environment is security and efficient mutual authentication and data transmission.

Generally, a large enterprise or organization needs to handle a huge amount of data and the number of users for information management. Therefore, a management entity with a large storage capacity is required, and the server must not be interrupted. Then, a management entity that can operate continuously is required. The important issue that the information management department must face is to ensure the data security and stable operation of the information management system. In our scheme, we overcome the problem of large storage and computing overhead by introducing cloud servers and the service interruption will not occur. In addition, the introduction of cloud servers can also avoid data loss or system crashes caused by hardware damage. There are four types of entities in an IMS system model: the Key Generation Center (KGC), the Access and Mobility Management (AMF), the Cloud Server (CS), and IoT groups as shown in Figure 1. In this system, AMF is used as the authentication server, the security server is assumed by KGC, a CS is introduced to complete the work related to the file server, and there are multiple groups (i.e. group_{*i*}, $i = 1, 2, 3, \dots, n$) of IoT devices that make up the IMS client. These entities can be roughly divided into three parts: information access unit, information transmission unit, and information processing unit.

- (1) The information access unit is composed of multiple IoT devices of the user, these IoT devices are divided into multiple IoT groups according to specific attributes. And this unit mainly forms human-computer interaction with the user, allowing the user to access the IMS through the IoT device and perform related operations on the information stored in the IMS

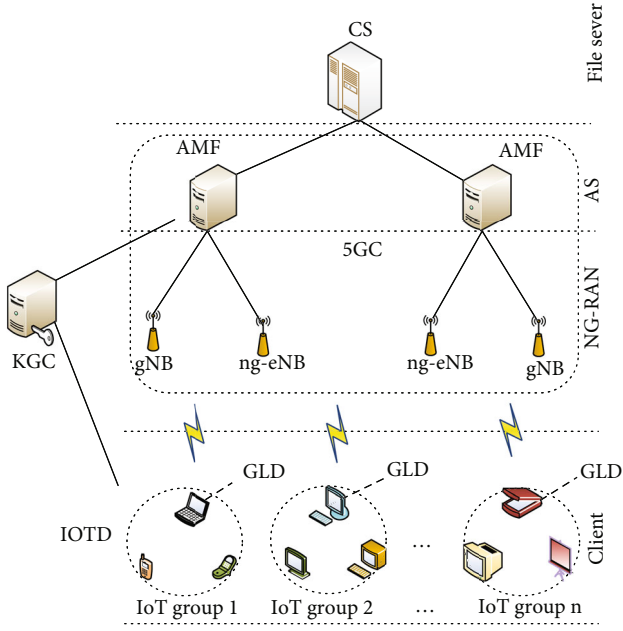


FIGURE 1: The system model of 5G-enabled information management system.

- (2) The information transmission unit is composed of two types of access points in NG-RAN, namely, gNB and ng-eNB. This part, like the base station, is mainly responsible for user access to the network. It is also a medium for sending and receiving information and communicating
- (3) The information processing unit is composed of three types of servers, KGC, AMF, and CS. This part mainly completes the authentication of user identity information, the encryption of data messages, and the function of processing the information stored in the server

The communication of the whole system includes communication between IoT device (IOTD) and KGC, AMF and KGC, IOTD and group leader of its IoT group, each IoT group and AMF, and AMF and CS. In our scheme, KGC is an incomplete trusted entity. It generates partial key during the interaction with IOTD and verifies whether the registered IOTD is legitimate and whether it is a corresponding group member. IOTD encrypts the communication data and sends it to the GLD of the IoT group for verification and aggregation. After that, the GLD sends the aggregate data of the whole group to the network through ng-ran, and AMF verifies the legitimacy of the entire IoT group. Various information of IMS is stored in CS, and users access data information in CS indirectly through AMF, because the communication between the AMF and the CS can be regarded as a completely trusted transmission, and mutual authentication can be performed between the AMF and the IOTD. The specific process is as described later. First, AMF selects a third-party cloud service operator to register and configure the cloud server and then establish the session

key after passing the mutual authentication between IOTD and AMF. Finally, AMF accesses the information in CS and sends it to each IOTD.

3.2. Security and Privacy Requirements. In IMS, users access the data in the file server through the IOTD accessing the system network. In this scheme, IMS is based on the 5G wireless network, so IoT devices access the system network through the nodes gNB and ng-eNB of the 5G access network. Since this process is carried out in a wireless network environment, there are some insecure elements of the connected node between IoT devices and networks can be derived from the system model presented above. And the external adversaries want to interfere with wireless transmission via control and disrupt the medium between IoT devices and networks. On the one hand, attackers can attack in a range of insecure means including replay attacks, man-in-the-middle attacks, and simulation attacks to simulate IoT devices or networks to launch various protocol attacks. On the other hand, privacy protection is indispensable for the sender. Therefore, the identity of the IOTD and the IoT group must have good concealment during the access of authentication. Even if the attacker is threatened, the real identity of the IOTD cannot be obtained.

Specifically, the following safety requirements should be met in the design proposal.

- (1) Mutual authentication: when the network is sent an access request by a group of IoT devices and needs to be accessed, AMF also authenticates the group of devices. In addition, each IOTD needs to confirm the legitimacy of AMF
- (2) Identity privacy protection: in the process of data transmission of IoT group, mutual authentication of network is usually accompanied. In order to ensure that the attacker will not steal the identity information and group identity information of the IOTD, the actual identity and group identity information of each IOTD need to be hidden in the message
- (3) Resistance to protocol attacks: typically, the scheme needs to resist various existing protocol attacks, such as replay, eavesdropping, and man-in-the-middle attacks
- (4) Data confidentiality and integrity: in general, the confidentiality and integrity of data transmission between the IoT group and the AMF should be guaranteed. Based on this, scheme can be designed
- (5) Efficient and feasible: the proposed scheme needs to reduce all kinds of costs in the process of authentication, including calculation cost, signaling cost, and communication cost

4. The Proposed Authentication Scheme

The efficient authentication for Internet of Things devices in information management systems consists of seven

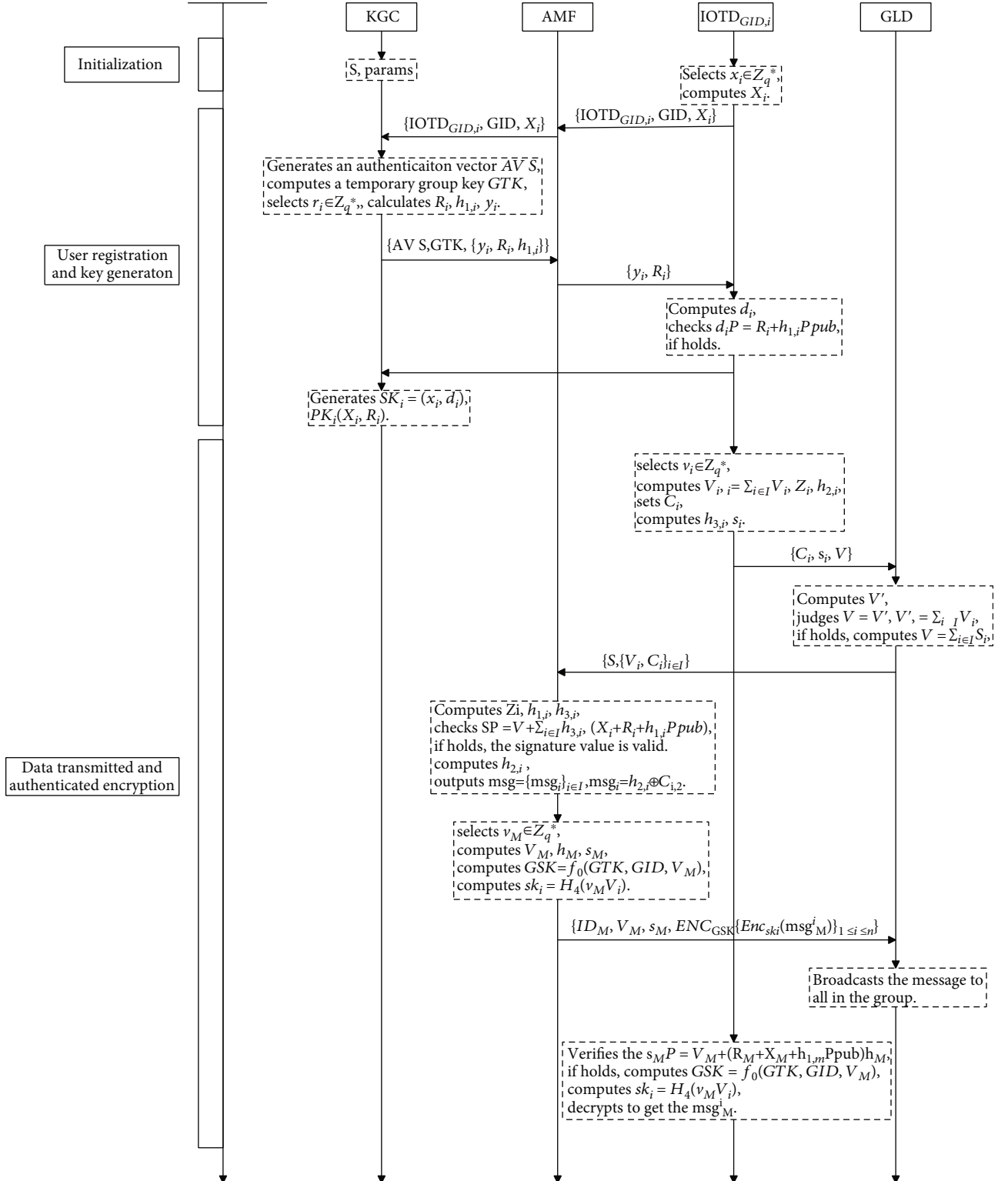


FIGURE 2: The procedures of the proposed scheme.

algorithms: system initialization, client key extraction, private key extraction, signcryption, aggregate signcryption, authentication, and aggregate authentication. The detailed process is shown in Figure 2.

- (1) System Initialization: input security parameters λ , the algorithm can return a series of system public parameters, and the master private key from the input value
- (2) Client Key Extraction: the user ID_i first chooses a random number x_i and then computes the common parameters X_i
- (3) Private Key Extraction: after KGC receives (ID_i, X_i) , it randomly selects r_i and calculates R_i, Y_i, y_i and set the private key (x_i, y_i) and public key (X_i, Y_i)
- (4) Signcryption: ID_i signcrypts the message m_i and send the signcryption to the receiver B as the identity ID_B
- (5) Aggregate Signcryption: after receiving signcrypts, B aggregates signcrypts and generates aggregate signcryption and sends them to verifier A
- (6) Authentication: B authenticates the signcryption after receiving the signcryption of the message m_i
- (7) Aggregate authentication: verifier A authenticates the aggregate signcryption after receiving the aggregate signcryption sent by B

The process of this scheme can be divided into the following three stages: initialization phase, user registration and key generation phase, and data transmitted and authenticated encryption phase.

4.1. Initialization Phase. During the system initialization phase, KGC executes the system initialization algorithm to generate the system common parameters params and master key. The detailed process is as follows:

- (1) KGC selects a cyclic additive group G of prime order q when it receives a security parameter λ . Suppose P is the generator of G
- (2) Then, KGC chooses four hash functions $H_0 = \{0, 1\}^{\ell_1} \rightarrow Z_q^*, H_1 = \{0, 1\}^{\ell_2} \rightarrow Z_q^*, H_2 = \{0, 1\}^* \times G$, and $H_3 = G \rightarrow Z_q^*$, where ℓ_1 is the bit length of the user and ℓ_2 is the bit length of the plain text message
- (3) KGC chooses $s \in_R Z_q^*$ as the master key and computes $P_{\text{pub}} = sP$
- (4) Finally, the $\{G, P, q, P_{\text{pub}}, H_{i\{0 \leq i \leq 3\}}\}$ is used as the public parameter, and for KGC, the master key remains his private secret

4.2. User Registration and Key Generation Phase. In this stage, each IOTD and AMF start to register and provide some of the private keys to obtain another part of the private key generated by KGC. Then, KGC sends a message

to the IOTD and AMF, respectively; the content is their corresponding private key. Each user legally has a distinctive ID , and each user has one or more terminal devices. Thus, multiple different devices constructed into an Internet of Things group should have common attributes. These common attributes are user attribution consistency, location consistency, functional similarity, or other similar characteristics. A GLD can be selected, which is based on the corresponding capabilities (such as the communication capabilities of each device, storage status, and battery status). In the 5G network, GLD will be activated at the same time when data is sent and received between the network and the user equipment. There is a dedicated group identity (GID) and a group key (GK) between each device and KGC that is prestored in the IoT group. And there are many IoT groups, one of these groups is denoted as group i ($i = 1, 2, 3 \dots n$). Each IOTD has an identity IOTD $D_{\text{GID},i}$, let IOTD $D_{\text{GID},1}, D_{\text{GID},2}, \dots, D_{\text{GID},n}$ be a member of the group i . This stage is illustrated as follows.

- (1) IOTD $D_{\text{GID},i}$ randomly selects $x_i \in_R Z_q^*$ and computes $X_i = x_i P$. Then, a message containing the terminal identification IOTD $D_{\text{GID},i}$, the group identity GID, and X_i is sent to AMF
- (2) Upon receiving the message, AMF transmits the identity verification request message to KGC, which contains the terminal identification IOTD $D_{\text{GID},i}$, the group identity GID, and X_i
- (3) When a message is received from the sender, KGC begins to validate the received terminal identification IOTD $D_{\text{GID},i}$ and GID validate the terminal IOTD $D_{\text{GID},i}$ as a member and also validate whether it is a member of group. Then, the KGC generates an authentication vector AVS and defines the GTK = $f_0(\text{GK}, \text{GID})$ as a temporary group key. Then, select a secure hash function f_0 safely, which is confidential between the IoT group, AMF, and KGC. Almost simultaneously, the KGC randomly selects $r_i \in_R Z_q^*$ and calculates $R_i = r_i P, h_{1,i} = H_1(\text{IOTD}_{\text{GID},i} \| X_i \| R_i \| \text{GID})$ and $y_i = r_i + sh_{1,i} + H_0(sX_i)$. Finally, the KGC embeds AVS, GTK, and $(y_i, R_i, h_{1,i})$ in the authentication response message sent to AMF
- (4) When the AMF receives the response message, (y_i, R_i) will be sent to IOTD $D_{\text{GID},i}$
- (5) When a message is received from the AMF, IOTD $D_{\text{GID},i}$ computes $d_i = y_i - H_0(x_i P_{\text{pub}})$ and checks the equation $d_i P = R_i + h_{1,i} P_{\text{pub}}$. If the equation hold, the KGC generates the complete key $SK_i = (x_i, d_i)$, $PK_i = (X_i, R_i)$.

The following details show that AMF generates key pairs in a similar way to the IOTD $D_{\text{GID},i}$.

- (1) Assume that the ID_M as an identity of the AMF, it selects a random number $x_M \in_R Z_q^*$ and calculates

$X_M = x_M P$. After the above calculation is completed, the request message is sent to KGC. The message includes its identity ID_M and X_M

- (2) After receiving the message. The KGC validates the ID_M by validating the messages it receives that contain ID_M and X_M . Then, the KGC generates an authentication vector AVS_M . At the same time, the KGC randomly selects $r_M \in_R Z_q^*$ and computes $R_M = r_M P$, $h_{1,M} = H_1(sn_M \| X_M \| R_M)$ and $y_M = r_M + sh_{1,M} + H_0(sX_M)$. Finally, the KGC embeds AVS_M and $(y_M, R_M, h_{1,M})$ in the authentication response message sent to AMF
- (3) When AMF receives a message from the KGC, it computes $d_M = y_M - H_0(x_M P_{pub})$ and checks the equation $d_M P = R_M + h_{1,M} P_{pub}$. If the equation hold, KGC generates the complete key $SK_M = (x_M, d_M)$, $PK_M = (X_M, R_M)$.

4.3. Data Transmitted and Authenticated Encryption Phase.

In this part, the IoT groups and the AMF perform data encryption and transmission operations while encrypting and transmitting data. And the CS we introduced is through a third-party cloud-computing technology operator such as Amazon, Alibaba Cloud, and Google. Then, each IoT group and AMF can perform mutual authentication. When the IOTD is connected to the network, GLD will aggregate the encrypted data and verification information of each member in the group. And the GLD of each group generates an aggregate signcryption. Then, AMF will send aggregated information and other public parameters by GLD. Based on the aggregated signcryption information, AMF can verify IoT members in each group. A key will be established between each terminal device and the AMF to ensure the security of the data. When the IoT group and the AMF interact, and the group session key GSK will be obtained. Subsequently, AMF uses its private key to generate a signature and send the encrypted data. After the authentication is passed, the user can access the data in the CS. The process is described in detail as follows; we assume that the following steps are executed in a certain group (i.e.group_{*i*}). And other groups are similar.

- (1) In a group_{*i*} ($i = 1, 2, 3 \dots n$), each IOTD_{GID,*i*} will select an element $v_i \in_R Z_q^*$. Then, five steps will be performed in proper order
 - (a) Computes $V_i = v_i P$ and sends it to other $n - 1$ group members
 - (b) Computes $V = \sum_{i \in I} V_i$, $Z_i = v_i (R_M + h_{1,M} P_{pub})$, $h_{2,i} = H_2(ID_M \| V_i \| V \| v_i X_M)$
 - (c) Sets $C_i = C_{i,1} \| C_{i,2} = \text{IOTD}_{\text{GID},i} \| \text{GID} \| (h_{2,i} \oplus \text{msg}_i)$
 - (d) Computes $h_{3,i} = H_3(V_i \| C_i \| X_i \| R_i \| Z_i)$
 - (e) Computes $s_i = v_i + (d_i + x_i) \cdot h_{3,i}$

- (2) IOTD_{GID,*i*} sends the above ciphertext C_i , the signcryption s_i , and V embedded access request message to GLD in the group_{*i*}
- (3) After receiving messages from other group members in group_{*i*}, the GLD judges whether V and V' are equal, where $V' = \sum_{i \in I} V_i$ and if $V = V'$, computes $S = \sum_{i \in I} s_i$, and sends the aggregated message $\{S, \{V_i, C_i\}_{i \in I}\}$ to AMF
- (4) The GLD of each group sends an aggregate message to the AMF. And then for the group_{*i*}, AMF begins to execute the following six steps. Similarly, it performs the same operation for each IoT group
 - (a) Computes $Z_i = d_M V_i$
 - (b) Sets $h_{1,i} = H_1(\text{IOTD}_{\text{GID},i} \| X_i \| R_i \| \text{GID})$.
 - (c) Sets $h_{3,i} = H_3(V_i \| C_i \| X_i \| R_i \| Z_i)$.
 - (d) Then, AMF can check whether the formula $SP = V + \sum_{i \in I} h_{3,i} (X_i + R_i + h_{1,i} P_{pub})$ is equal. The detailed calculation process is as follows

$$\begin{aligned}
 SP &= \sum_{i \in I} (v_i + (d_i + x_i) \cdot h_{3,i}) P \\
 &= \sum_{i \in I} (v_i + (y_i - H_0(x_i P_{pub})) + x_i \cdot h_{3,i}) P \\
 &= \sum_{i \in I} (v_i + (r_i + s \cdot h_{1,i} + x_i) \cdot h_{3,i}) P \\
 &= \sum_{i \in I} V_i + (R_i + P_{pub} \cdot h_{1,i} + X_i) \cdot h_{3,i} \\
 &= V + \sum_{i \in I} (X_i + R_i + P_{pub} \cdot h_{1,i}) \cdot h_{3,i}.
 \end{aligned} \tag{1}$$

We say that the signature value is valid if the equation holds. The AMF can ensure that the received ciphertext C_i is not only valid but also belongs to a legal IOTD_{GID,*i*} in the group_{*i*}.

- (e) Computes $h_{2,i} = H_2(ID_M \| V_i \| V \| x_M V_i)$
 - (f) AMF produces an output $\text{msg} = \{\text{msg}_i\}_{i \in I}$, where $\text{msg}_i = h_{2,i} \oplus C_{i,2}$
- (5) If the data in the CS needs to be sent to an IOTD, it needs to be sent through AMF. AMF reads the data directly from the preregistered and configured CS and then sends the read data to IOTD_{-(GID,*i*)} in the group_{*i*}. After that, AMF performs the following steps
 - (a) Selects an element $v_M \in_R Z_q^*$, then three values will be calculated, and they are $V_M = v_M P$, $h_M = H_2(ID_M \| V_M \| \text{GID})$, $s_M = v_M + (d_M + x_M) \cdot h_M$
 - (b) Computes $\text{GSK} = f_0(\text{GTK}, \text{GID}, V_M)$ as the session key with GLD to encrypt the message msg_M
 - (c) Computes $sk_i = H_4(v_M \cdot V_i)$ as the session key with IOTD_{GID,*i*} to encrypt msg_M^i

TABLE 1: The symbol of notation.

Symbol	Definition
N_g	A quantity of groups
N_t	A quantity of terminals

TABLE 2: The signaling overhead.

Protocol	The number of signaling
Cao's scheme	$7N_t + 3N_g$
Sultan's scheme	$N_t + N_g$
Our scheme	$N_t + 3N_g$

- (d) Generates an aggregate message $(ID_M, V_M, s_M, \text{ENC}_{\text{GSK}}\{\{\text{ENC}_{\text{sk}_i}(\text{msg}_M^i)\}\}_{1 \leq i \leq n})$ and send the aggregate message to GLD
- (e) After receiving the message, GLD broadcasts the message to all in the group
- (f) $\text{IOTD}_{\text{GID},i}$ verifies that the following equation is true: $s_M P = V_M + (R_M + X_M + h_{1,M} \cdot P_{\text{pub}}) \cdot h_M$. The detailed process is as follows

$$\begin{aligned}
 s_M P &= (v_M + (d_M + x_M) \cdot h_M) P \\
 &= v_M P + (d_M P + x_M P) \cdot h_M \\
 &= v_M + ((r_M + s \cdot h_{1,M}) P + x_M P) \\
 &= V_M + (R_M + X_M + h_{1,M} \cdot P_{\text{pub}}) \cdot h_M.
 \end{aligned} \tag{2}$$

If the equation holds, computes $\text{GSK} = f_0(\text{GTK}, \text{GID}, V_M)$, $\text{sk}_i = H_4(v_M \cdot V_i)$, and decrypts to get the message msg_M^i .

5. Security Analysis

In this part, the security of the protocol has been analyzed. And we have defined six security goals.

- (1) Mutual authentication: in the IoT group and AMF, mutual authentication can be implemented in our scheme. In the process of AMF's identity authentication for each $\text{IOTD}_{\text{GID},i}$, the legal signcryption s_i is generated only by the convincing IOTD, and GLD calculates the valid aggregate signature. If the adversary does not have a correct private key, it is impossible to obtain a valid aggregate value. In addition, a private key can be used to generate a signcryption to authenticate the AMF
- (2) Data privacy and integrity: in order to strengthen data security, our scheme uses certificateless aggregation and signcryption technology. When data is transferred from $\text{IOTD}_{\text{GID},i}$ to AMF, only legitimate users have a valid private key. And the legal public

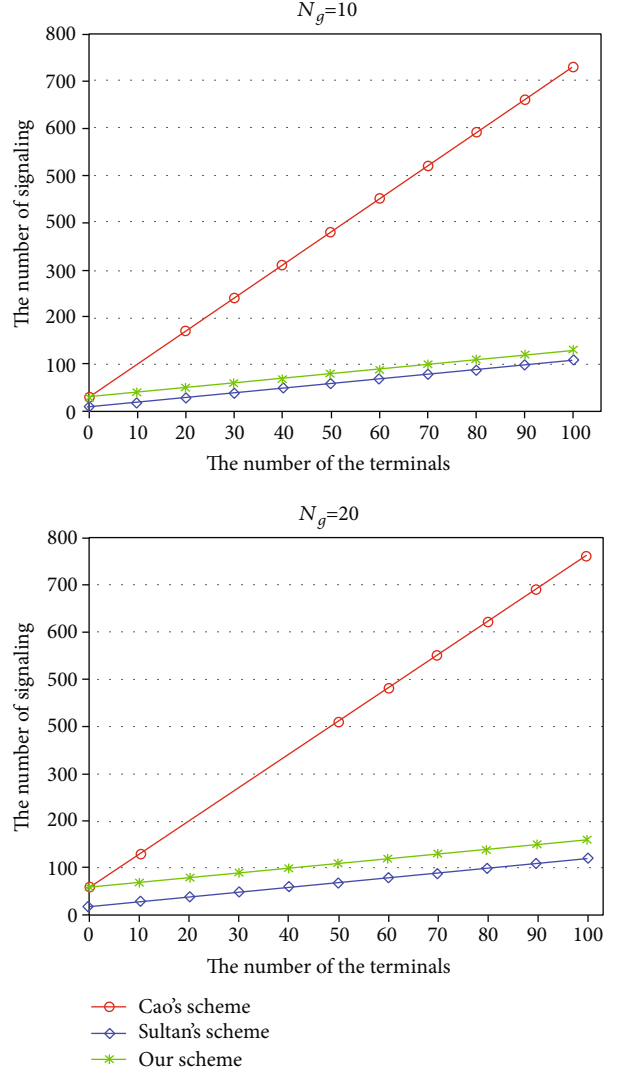


FIGURE 3: The comparison of signaling cost.

key of AMF is jointly used to signcrypt the data. This operation is run by GLD. And only legal AMF can verify the aggregate signcryption and decrypt it. In addition, when the data is transmitted from AMF to $\text{IOTD}_{\text{GID},i}$, use the session key of each $\text{IOTD}_{\text{GID},i}$ and AMF (IoT group and AMF) to ensure the privacy and integrity of the data

- (3) Identity privacy protection: after the $\text{IOTD}_{\text{GID},i}$'s relevant information are encrypted in this scheme, it can protect the user's identity information from being leaked. According to the proposed scheme, we use AMF's public key to encrypt the $\text{IOTD}_{\text{GID},i}$ and GID. If an adversary wants to decrypt the information of interest, he must know the valid AMF private key. So, they cannot use the legal identity to further implement the replay attack
- (4) Attack resistance: there are some attacks that can be resisted in our scheme, such as replay attacks,

TABLE 3: The symbol of notation.

Symbol	Definition
T_M	Time of a point multiplication
T_H	Time of a hash function operation
T_P	A pairing operation time

TABLE 4: The time required for the encryption operation.

UE	T_M	T_H	T_P
IOTD	4.312	0.514	31.812
AMF	1.048	0.036	8.671

modification attacks, impersonation attacks, eavesdropping, and man-in-middle attacks

- (i) Replay attacks: since a random value is introduced to generate the signcryption in the construction of our scheme, which can resist replay attacks. In detail, we ensure the randomness of the message by selecting a random value v_i during the data transmission phase. Thus, the adversary cannot perform a replay attack without obtaining the value v_i
- (ii) Modification attacks: in this proposal, a valid triple $(S, \{V_i, C_i\}_{1 \leq i \leq l})$, S is the signature valid. We can check whether the message has been modified by the adversary through the formula $SP = V + \sum_{i \in I} h_{3,i}(X_i + R_i + h_{1,i}P_{\text{pub}})$
- (iii) Impersonation attacks: when an adversary wants to send a forged message to AMF, it needs to be simulated as a legitimate device $\text{IOTD}_{\text{GID},i}$. At this time, AMF will test the formula $SP = V + \sum_{i \in I} h_{3,i}(X_i + R_i + h_{1,i}P_{\text{pub}})$, if it is established, it will pass the verification; otherwise, stop it
- (iv) Man-in-the-middle attacks: our scheme can resist an attack such as an man-in-the-middle attacks. The prerequisite for the adversary to generate the correct signcryption or signature information is to know part of the private partial-key of AMF, which is related to the generation of the session key. And the generation of the session key requires the adversary to break the Computational Diffie-Hellman (CDH) problem. Specifically, in the data transmission and authentication encryption stage, we set multiple points $(V_i, X_i, R_i, Z_i, \text{etc.})$ on the elliptic curve in the content of the transmission message to ensure certain security. The adversary needs to break through the points we set on the curve based on CDH problem to obtain the corresponding private key and session key parameters
- (v) Eavesdropping: no adversary can obtain the session key by eavesdropping. If an adversary can

TABLE 5: The computation overhead.

Protocol	IOTD_i	AMF
Cao's scheme	$T_H + 2T_M$	$2N_t(T_H + T_P)$
Sultan's scheme	$3T_H + 6T_M$	$N_t(T_H + 2T_M + 4T_P)$
Our scheme	$3T_H + 8T_M$	$(3N_t + 1)T_H + (5N_t + 1)T_M$

forge a signature or aggregate signature information, a private key needs to be forged to make entities AMF or $\text{IOTD}_{\text{GID},i}$ believe. In summary, the scheme is secure

Based on the above analysis, the IMS in our scheme can resist the above-mentioned attacks to ensure the information security of the entire system. It prevents illegal users from entering the IMS by resisting replay attacks and impersonation attacks. And to ensure that the security of the information stored in the system by each entity in the IMS is not tampered with and eavesdropped through the other security features.

- (5) Signaling Congestion Avoidance: we used the idea of certificateless signcryption technology to construct the scheme. A large number of IoT devices send access requirements to GLD, and GLD aggregates this information to generate messages. It can reduce the amount of signaling and effectively improving the efficiency of access authentication. The authentication process includes data transmission, which reduces the communication overhead of the scheme and reduces the pressure on the communication network

6. Performance Analysis

Compared with some similar schemes, this scheme has greater advantages in performance. In this part, we compare the signaling overhead, computing overhead, and communication overhead separately with Cao's scheme [24] and Sultan's scheme [27]. In Table 1, we describe the symbol definition where N_g and N_t represent the number of two different entities.

6.1. Signaling Overhead. In this section, we analyze our scheme, Cao's scheme [24], and Sultan's scheme [27]. And we take the number of signaling messages as a parameter.

In Cao's scheme, the communication between IOTD and AMF needs $7N_t + 3N_g$ signaling messages to realize authentication. In the scheme of Sultan, the communication between IOTD and AMF needs $N_t + N_g$ signaling messages to realize authentication. In this scheme, IoT devices and AMF achieve multiparty authentication need $N_t + 3N_g$ signaling messages. We can see the theoretical comparison results in Table 2.

In Cao's scheme, when the terminal communicates with the network, a great quantity access request messages can be integrated. Then, the aggregated message is sent to the network, which can be verified by AMF. After the

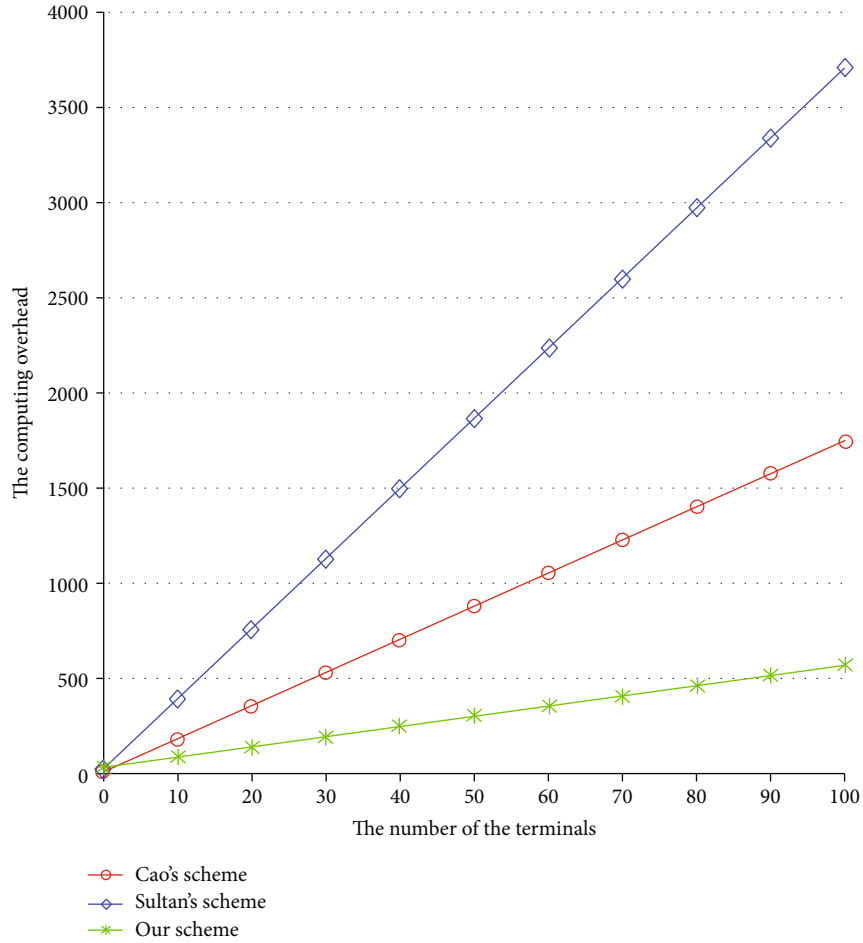


FIGURE 4: The comparison of computation cost.

authentication is successful, AMF sends messages to the terminal in broadcast mode. Due to the broadcast mechanism, Cao's scheme has higher signaling overhead compared with Sultan's scheme and our scheme. Based on aggregate sign-encryption technology, our scheme embeds data from different devices into authentication request messages. Then, they will be sent to AMF for authentication after aggregation by group leaders. In addition, the user terminal to authenticate the AMF generated signature authentication network through GLD. This method does not require each member to authenticate the message one by one, thereby greatly reducing the signaling overhead.

Figure 3 shows the change of the total number of signaling messages with the increasing number of terminal devices when $N_g = 10$ and $N_g = 20$, respectively. When the number of terminals increases from 1 to 100, the signaling cost in this scheme is similar to Sultan's scheme but is significantly better than Cao's scheme. It can be concluded that this scheme has good performance in signaling overhead.

6.2. Computational Overhead. In our scheme, we mainly consider three relatively time-consuming calculations (as shown in Table 3). T_M stands for dot multiplication operation, T_P stands for pair operation, and T_H stands for a hash operation. These calculations were tested on a laptop

TABLE 6: The communication overhead.

Protocol	Communication overhead
Cao's scheme	$(a + 2 + 2c)N_t + (a + 2)N_g$
Sultan's scheme	$N_t + N_g$
Our scheme	$aN_t + (a + 2)N_g$

computer (Computer brand: Lenovo, processor: I5-3320 M 2.6 GHZ, memory: 4 G bytes, operating system: window7) and realized by calling the JPBC library. The running time of each operation is shown in Table 4.

In Cao's scheme, the computational overhead of each IOTD and AMF is $T_H + 2T_M$ and $2N_t(T_H + T_P)$, respectively. In Sultan's scheme, the computational overhead of each IOTD and AMF are $3T_H + 6T_M$ and $N_t(T_H + 2T_M + 4T_P)$, respectively. In our proposed scheme, the computational overhead of each IOTD and AMF are $3T_H + 8T_M$ and $(3N_t + 1)T_H + (5N_t + 1)T_M$, respectively. We can see the computational overhead in each scheme from Table 5 and the relationship between them in Figure 4.

Due to the large number of pairing operations in Cao's scheme and Sultan's scheme, the computation cost of the two schemes is high. In Cao's scheme, the computational

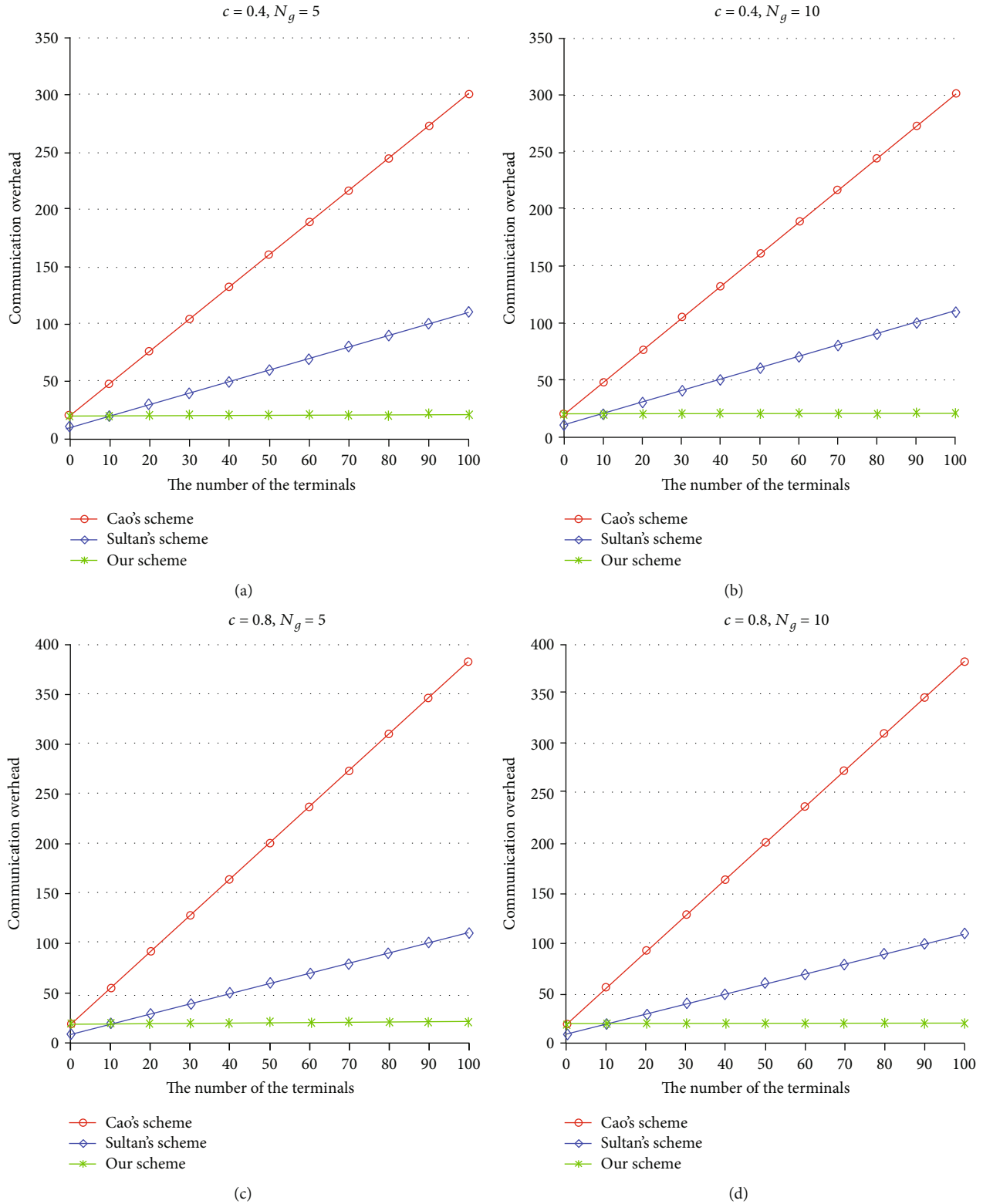


FIGURE 5: The comparison of communication cost.

cost of the protocol is the largest because it performs time-consuming mapping hash, bilinear pairing, and point multiplication. Our scheme realizes message aggregation authentication without bilinear pairing operation, so the computation cost of this scheme is less than that of the other

two schemes. Figure 4 shows the comparison between the scheme in our scheme and the other two schemes. When the number of terminals increases from 0 to 100, the computational cost of the scheme in this chapter is significantly lower than that of the other two schemes.

6.3. Communication Overhead. We think that the transmission between AMF and IOTD_i is a unit. There are a units between IOTD_i and GLD, and c units are between eNB and IOTD_i. Since the distance between IOTD_i and IOTD_j is less than 100 meters, the cost of a units is much less than that of one unit. Due to different eNB locations, the distance between eNB and IOTD_i is also various. Also, the distance between IOTD_i and entities connected by wire is relatively fixed. In order to facilitate our analysis of the proposed scheme, we assume $a = 0.01$. Because of using the control plane to optimize the transmission mechanism, AKA scheme generates additional transmission overhead. During the establishment of the data holder, the consumption of AMF and IOTD_i is two units. And the consumption of eNB and IOTD_i transmission is $2c$ units.

After analysis, the communication cost of Cao's scheme is $(a + 2 + 2c)N_t + (a + 2)N_g$, that of Sultan is $N_t + N_g$, and that of our scheme is $aN_t + (a + 2)N_g$. In Table 6, the total communication overhead of Cao's scheme, Sultan's scheme, and our scheme are compared. Figure 5 shows the comparison of communication consumption between the scheme in this chapter and the other two schemes in four cases: $c = 0.4, N_g = 5, c = 0.4, N_g = 10, c = 0.8, N_g = 5, c = 0.8, N_g = 10$. It can be clearly seen from Figure 5 that when the number of user terminals increases from 0 to 100, the communication overhead of the scheme in this chapter is significantly lower than that of the other two schemes.

7. Conclusions and Future Work

In order to perform authentication and data transmission safely and efficiently in IMS, we propose an efficient and secure authentication for IoT device in information management systems. By screening the specific attributes of the device, an IOTD in the IoT group is selected as the group leader to perform message aggregation, signature, encryption, and transmission in our scheme. Therefore, while ensuring user identity privacy and data integrity, it greatly improves the efficiency of mutual authentication and data transmission between the user and the server in IMS. And it solves the large signaling overhead caused by multiple IoT devices simultaneously accessing the IMS, low authentication efficiency, and network congestion caused by processing multiple messages at the same time. Then, security analysis shows that the protocol can resist various malicious attacks. Performance analysis also shows that this scheme is effective in terms of signaling overhead, computing overhead, and communication overhead. In future research, it will be interesting to design a secure, efficient, and meet the needs of more intelligent scenarios in a IoT device authentication scheme.

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 have declared that no conflict of interest exists.

Acknowledgments

This work is supported by the National Key R&D Program of China (2017YFB0802000), the Key Research and Development Program of Shaanxi (2019KW-053, 2020ZDLGY08-04), the Innovation Capability Support Program of Shaanxi (2020KJXX-052), Guangxi Cooperative Innovation Center of Cloud Computing and Big Data (No. YD1903), and the Basic Research Program of Qinghai Province (No. 2020-ZJ-701).

References

- [1] K. B. Jalbani, A. H. Jalbani, and S. S. Soomro, *IoT Security: To Secure IoT Devices With Two-Factor Authentication by Using a Secure Protocol*, Industrial Internet of Things and Cyber-Physical Systems: Transforming the Conventional to Digital, 2020.
- [2] R. Arshad and N. Ikram, "Elliptic curve cryptography based mutual authentication scheme for session initiation protocol," *Multimedia Tools and Applications*, vol. 66, no. 2, pp. 165–178, 2013.
- [3] S. K. Hafizul Islam and G. P. Biswas, "Design of improved password authentication and update scheme based on elliptic curve cryptography," *Mathematical and Computer Modelling*, vol. 57, no. 11-12, pp. 2703–2717, 2013.
- [4] M. Hou, T. Kang, and L. Guo, "A blockchain based architecture for IoT data sharing system," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Austin, TX, USA, March 2020.
- [5] S. Mandal, B. Bera, A. Kunar, K. Kwang, and Y. Park, "Certificateless-signcryption-based three-factor user access control scheme for IoT environment," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 1–1, 2020.
- [6] Huawei Technologies Co, "5G Opening Up New Business Opportunities.," White Paper, 2016.
- [7] 3GPP, "Study on The Security Aspects of the Next Generation System," Technical Report, 2017.
- [8] 5G PPP, "5G PPP Phase 1 Security Landscape," 2017.
- [9] NGMN, "5G Security Recommendations (Networking Slicing, Mobile Edge Computing)," White Paper, 2016.
- [10] Nokia, "Security Challenge and Opportunities for 5G Mobile Networks," White Paper, 2017.
- [11] D. Wang, J. Shen, J. Liu, and K. K. R. Choo, "Rethinking authentication on smart mobile devices," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7079037, 4 pages, 2018.
- [12] T.-Y. Teh, Y.-S. Lee, Z.-Y. Cheah, and J.-J. Chin, "IBI-mobile authentication: a prototype to facilitate access control using identity-based identification on mobile smart devices," *Wireless Personal Communications*, vol. 94, no. 1, pp. 127–144, 2017.
- [13] W. I. Khedr, K. M. Hosny, M. M. Khashaba, and F. A. Amer, "Prediction-based secured handover authentication for mobile cloud computing," *Wireless Networks*, vol. 26, no. 6, pp. 4657–4675, 2020.

- [14] A. P. G. Lopes and P. R. L. Gondim, "Group authentication protocol based on aggregated signatures for d2d communication," *Computer Networks*, vol. 178, article 107192, 2020.
- [15] H. Yang, Y. Wu, J. Zhang, H. Zheng, Y. Ji, and Y. Lee, "Blockchain: blockchain-based trusted cloud radio over optical fiber network for 5G fronthaul," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pp. 1–3, San Diego, CA, USA, March 2018.
- [16] H. Yang, Y. Li, S. Guo, J. Ding, Y. Lee, and J. Zhang, "Distributed blockchain-based trusted control with multi-controller collaboration for software defined data center optical networks in 5G and beyond," in *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, San Diego, CA, USA, March 2019.
- [17] Y. Zhang, R. H. Deng, E. Bertino, and D. Zheng, "Robust and universal seamless handover authentication in 5G HetNets," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 858–874, 2021.
- [18] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008, <http://bitcoin.org/bitcoin.pdf>.
- [19] Y. Chen, J. Wang, K. Chi, and C. Tseng, "Group-based authentication and key agreement," *Wireless Personal Communications*, vol. 62, no. 4, pp. 965–979, 2012.
- [20] S. Basudan, "LEGA: a lightweight and efficient group authentication protocol for massive machine type communication in 5G networks," *Journal of Communications and Information Networks*, vol. 5, no. 4, pp. 457–466, 2020.
- [21] Y. Aydin, G. Karabulut, and H. Yanikomeroglu, "A flexible and lightweight group authentication scheme," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10277–10287, 2020.
- [22] C. Lai, H. Li, R. Lu, and X. S. Shen, "Se-aka: a secure and efficient group authentication and key agreement protocol for LTE networks," *Computer Networks*, vol. 57, no. 17, pp. 3492–3510, 2013.
- [23] R. Jiang, C. Lai, J. Luo, X. Wang, and H. Wang, "EAP-based group authentication and key agreement protocol for machine-type communications," *International Journal of Distributed Sensor Networks*, vol. 11, 318 pages, 2013.
- [24] J. Cao, M. Ma, and H. Li, "Gbaam: group-based access authentication for MTC in LTE networks," *Security and Communication Networks*, vol. 8, no. 17, pp. 3282–3299, 2015.
- [25] C. Lai, H. Li, R. Lu, R. Jiang, and X. Shen, "Lgth: a lightweight group authentication protocol for machine-type communication in LTE networks," in *2013 IEEE Global Communications Conference (GLOBECOM)*, pp. 832–837, Atlanta, GA, USA, December 2013.
- [26] J. Li, M. Wen, and T. Zhang, "Group-based authentication and key agreement with dynamic policy updating for MTC in LTE-a networks," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 408–417, 2016.
- [27] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 772–782, 2017.
- [28] Y. Zhang, F. Ren, A. Wu, T. Zhang, J. Cao, and D. Zheng, "Certificateless multi-party authenticated encryption for NB-IoT terminals in 5G networks," *IEEE Access*, vol. 7, pp. 114721–114730, 2019.
- [29] S. Behrad, E. Bertin, S. Tuffin, and N. Crespi, "A new scalable authentication and access control mechanism for 5G-based IoT," *Future Generation Computer Systems*, vol. 108, pp. 46–61, 2020.
- [30] B. Seok, J. Sicato, T. Erzhen, C. Xuan, and J. Park, "Secure D2D communication for 5G IoT network based on lightweight cryptography," *Applied Sciences*, vol. 10, no. 1, p. 217, 2019.
- [31] J. Sanchez-Gomez, D. Garcia-Carrillo, R. Marin-Perez, A. Skarmeta, and A. Skarmeta, "Secure authentication and credential establishment in narrowband IoT and 5G," *Sensors*, vol. 20, no. 3, p. 882, 2020.

Research Article

Differential Privacy Location Protection Method Based on the Markov Model

Hongtao Li ¹, Yue Wang ¹, Feng Guo,² Jie Wang,¹ Bo Wang,¹ and Chuankun Wu²

¹College of Mathematics and Computer Science, Shanxi Normal University, 041000 Linfen, China

²School of Information Science and Engineering, Linyi University, Linyi 276002, China

Correspondence should be addressed to Yue Wang; 951727247@qq.com

Received 4 May 2021; Accepted 20 June 2021; Published 1 July 2021

Academic Editor: Ximeng Liu

Copyright © 2021 Hongtao Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Location-based services (LBS) have become an important research area with the rapid development of mobile Internet technology, GPS positioning technology, and the widespread application of smart phones and social networks. LBS can provide convenience and flexibility for the users' daily life, but at the same time, it also brings security risks to the users' privacy. Untrusted or malicious LBS servers can collect users' location data through various ways and disclose it to the third party, thus causing users' privacy leakage. In this paper, a differential privacy location protection method based on the Markov model for user's location privacy is proposed. Firstly, the transition probability matrix between states of the n -order Markov model is used to predict the occurrence state and development trend of events; thereby, the user's location is predicted, and then a location prediction algorithm based on the Markov model (LPAM) is proposed. Secondly, a location protection algorithm based on differential privacy (LPADP) is proposed, in which location privacy tree (LPT) is constructed according to the location data and the difficulty of retrieval, the two nodes with the largest predicted value of LPT are allocated with a reasonable privacy budget, and Laplace noise is added to protect location privacy. Theoretical analysis and experimental results show that the proposed method not only meets the requirements of differential privacy and protects location privacy effectively but also has high data availability and low time complexity.

1. Introduction

In recent years, the rapid development of mobile Internet technology, Internet of things technology, and GPS positioning technology has promoted the rapid development of various smart devices and social networks, making location-based services (LBS) widely applied in people's lives [1–4]. Users can send their identity, location, interests, and other information to the LBS server through the LBS application, so as to query and obtain the required information, such as the nearest shopping center, supermarket, and restaurant. The LBS service provider can also predict the next location of the user according to the current location of the user and provide the user with relevant information of the area before the user enters the next area. For example, in the aspect of traffic, vehicle positioning and prediction can enable users to get a faster and more convenient path. However, while users enjoy the convenience brought by LBS service, it will

also lead to the risk of sensitive information leakage. When users query information from the LBS server, they need to send personal identity, location, interests, and other information to the LBS server. If this information is leaked by untrusted or malicious LBS servers, the attackers can not only link the user's identity with location and interests but can also infer more user's private information. Therefore, location privacy protection in LBS is becoming more and more important and has been attached great importance to by relevant fields.

At present, domestic and foreign researchers have conducted a large number of studies on location privacy protection and proposed a variety of solutions to the privacy protection problems in LBS. The dominating location privacy protection technologies include cryptography, k -anonymity, and differential privacy.

Cryptography was proposed by Diffie and Hellman with the idea of public key cryptography in 1976 [5]. The main

idea of location privacy protection technology using cryptography is to encrypt the user's query information. Because the users' query information is not visible to the server, the attacker cannot infer the true data of the user even after obtaining the encrypted data. Although cryptography can effectively protect the privacy of users, it costs a lot in computing and communication and suffers insufficient data availability.

k -anonymity was proposed by Samarati and Sweeney in 1998 [6], which can ensure that each individual record stored in the publication dataset cannot be distinguished from other $k-1$ individuals for sensitive attributes. k -anonymity mechanism requires that the same quasiidentifier must have at least k records; so, the attackers cannot link the records through the quasiidentifier. Although k -anonymity technology can prevent identity disclosure, it cannot prevent attribute disclosure nor can it resist homogeneous attacks and background knowledge attacks.

Differential privacy was proposed by Dwork et al. in 2006 [7], which can protect the privacy information effectively even if the attacker gets the user's background knowledge. Differential privacy has a rigorous statistical model that facilitates the use of mathematical tools and quantitative analysis and proof.

At present, location privacy protection faces great challenges. In this paper, a differential privacy location protection method based on the Markov model is proposed. The main contributions of this paper as follows:

- (1) In this paper, the Markov model is used to predict the location information, and the probability transfer matrix between the states of the N -order Markov model is used to predict the state of occurrence of events and their development trend, so as to predict the user's location. Then, a location prediction algorithm based on the Markov model (LPAM) is proposed
- (2) A location protection algorithm based on differential privacy (LPADP) is proposed, in which location privacy tree (LPT) is constructed according to the location data and the difficulty of retrieval, the two nodes with the largest predicted value of LPT are allocated a reasonable privacy budget, and Laplace noise is added to protect location privacy
- (3) A comprehensive theoretical and experimental analysis has been done between the proposed method and the related works. Results show that our method meets the requirements of differential privacy and protects user location privacy effectively

The rest of this paper is organized as follows: Section 2 introduces the related works; Section 3 introduces the definition, transition probability matrix, system model, and attack model; Section 4 introduces the LPAM algorithm and LPADP proposed in this paper; Section 5 conducts experiments on data availability, privacy protection degree, and algorithm run-time of algorithm proposed in this paper; Section 6 is the conclusion of this paper.

2. Preliminaries

2.1. Definitions

Definition 1. (Markov model) [8, 9]. Let E be the discrete state space of random sequence $\{X(n), n = 0, 1, 2, \dots\}$. If for any m nonnegative integers n_1, n_2, \dots, n_m ($0 \leq n_1 < n_2 < \dots < n_m$) and any natural number k , and any $i_1, i_2, \dots, i_m, j \in E$ satisfies the following conditions:

$$P\{X(n_m + k) | X(n_1), X(n_2), \dots, X(n_m)\} = P\{X(n_m + k) | X(n_m)\}. \quad (1)$$

Then, $\{X(n), n = 0, 1, 2, \dots\}$ is called the one-order Markov model. This equation shows that the state of the next moment only depends on the present moment and has nothing to do with the past moment. This property is the Markov model with no aftereffect.

The n -order Markov model means that the state of the next moment is not only related to the present moment but also related to the past moment; so, the prediction is more comprehensive and effective.

Definition 2. (Neighboring dataset). Let the data set D and D' have the same attribute structure, and the symmetric difference between the D and D' is recorded as $D\Delta D'$, $|D\Delta D'|$ represents the number of symmetry differences. If $|D\Delta D'| = 1$, then D and D' are called neighboring dataset (also known as brothers data sets).

Definition 3. (Differential privacy) [10, 11]. There is a random algorithm M and all possible outputs of M are SM . For any two neighboring datasets D and D' , if algorithm M satisfies the following conditions:

$$\Pr [M(D) \in SM] \leq e^\epsilon \times \Pr [M(D') \in SM], \quad (2)$$

then algorithm M provides ϵ -differential privacy protection, where parameter ϵ is called privacy protection budget. The larger the ϵ is, the higher the data availability is, and the lower the degree of privacy protection is; on the contrary, the lower the data availability is, the higher the degree of privacy protection is.

Definition 4. (Sensitivity). Let d be a positive integer, D is a set of data sets, and $f : D \rightarrow R^d$ is a function. The function sensitivity represented by Δf has the following definition: $\Delta f = \max \|f(D) - f(D')\|_1$, where $\|\cdot\|_1$ is the Manhattan distance.

Definition 5. (Laplace mechanism) [12, 13]. Given dataset D , there is a function $f : D \rightarrow R^d$, the sensitivity is Δf , and then the random algorithm $M(D) = f(D) + Y$ provides ϵ -differential privacy protection, where $Y \sim \text{Lap}(\Delta f/\epsilon)$ is the random noise and obeys the Laplace distribution with the scale parameter $\Delta f/\epsilon$.

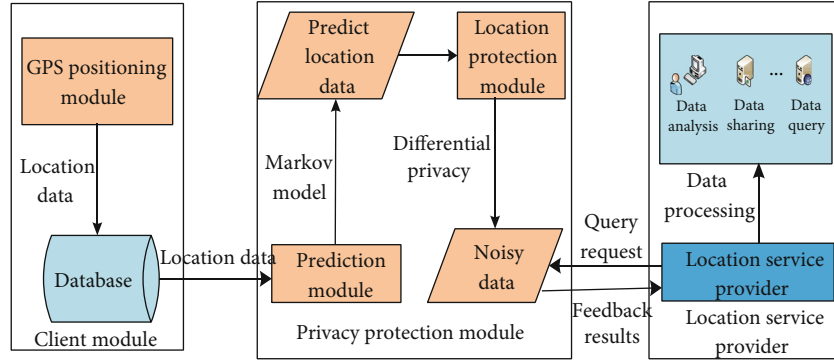


FIGURE 1: System structure.

2.2. Transition Probability. In this paper, the n -order Markov model is used to predict the location. The basic method of Markov prediction is to use the transition probability matrix between states to predict the occurrence and development probability of events.

The transition probability is derived by using the one-order Markov model [14, 15].

$$P = \frac{N_{ij}}{\sum_{j=1}^n N_{ij}}, \quad (3)$$

where N_{ij} is the number of times that location i turns to location j , and P is called one-step transition probability.

The recurrence relation can be obtained by C-K equation.

$$P(n) = PP(n-1) = P(n-1)P, \quad (4)$$

$$P(n) = P^n, \quad (5)$$

where P^n is called the n -step transition probability matrix of the Markov model.

2.3. System Structure and Threat Model. The system structure of this paper is shown in Figure 1, which is mainly composed of client module, privacy protection module, and location service provider module. The client module acquires the user's location information mainly through the GPS positioning module and stores the location data in the database. The privacy protection module is composed of prediction module and location protection module. The prediction module predicts users' location by the Markov model, while the location protection module protects users' location by differential privacy. Location service providers can respond to users' query requests, feedback the query results to users, and use the feedback results for data analysis, data sharing and data query, and other services.

In this paper, a differential privacy location protection method based on the Markov model is proposed to solve the problem of users' location privacy disclosure. The user's location is acquired through the GPS positioning module and stored in the database. In the prediction module, the n -order Markov model is used to predict the user's location, and the LPAM algorithm is proposed. In the location

protection module, differential privacy technology is used to protect location data, and LPADP algorithm is proposed. Location service providers can respond to users' query requests, feedback the query results to users, and use the feedback results for data analysis, data sharing, and data query and other services.

Almost all LBS providers collect users' personal data, such as identity, location, and interests. Many LBS providers provide different security guarantees, such as Google, Twitter, and Youtube. Once these LBS providers are attacked, users' privacy information will be leaked. The threat model of this paper is shown in Figure 2. The users' location data is acquired through the smart mobile devices equipped with positioning technology, such as mobile phones, portable computers, and cars, and the obtained location data is uploaded to the database. Then, the location data is transferred to the LBS servers for further intelligent data processing, which allows users to get convenient services from the LBS providers, such as in the aspect of traffic, vehicle positioning, and prediction that can enable users to get a faster and more convenient path; in terms of travel, location positioning and prediction can help users obtain nearby scenic spots and accommodations with better evaluations. The intelligent data processing of LBS servers mainly includes two parts: location prediction and location protection. The attackers can obtain the user's personal data by attacking the user's smart terminals, LBS servers, or location service providers, which will result in the users' privacy being breached.

3. Differential Privacy Location Protection Method Based on the Markov Model

To solve the problem of users' location privacy leakage, a differential privacy location protection method based on the Markov model is proposed in this paper. Firstly, the transition probability matrix between states of the n -order Markov model is used to predict the location information, and LPAM algorithm is proposed. Secondly, LPT is constructed according to the characteristics of location data and the difficulty of retrieval. Finally, the LPADP algorithm is proposed to protect users' location information by using differential privacy technology.

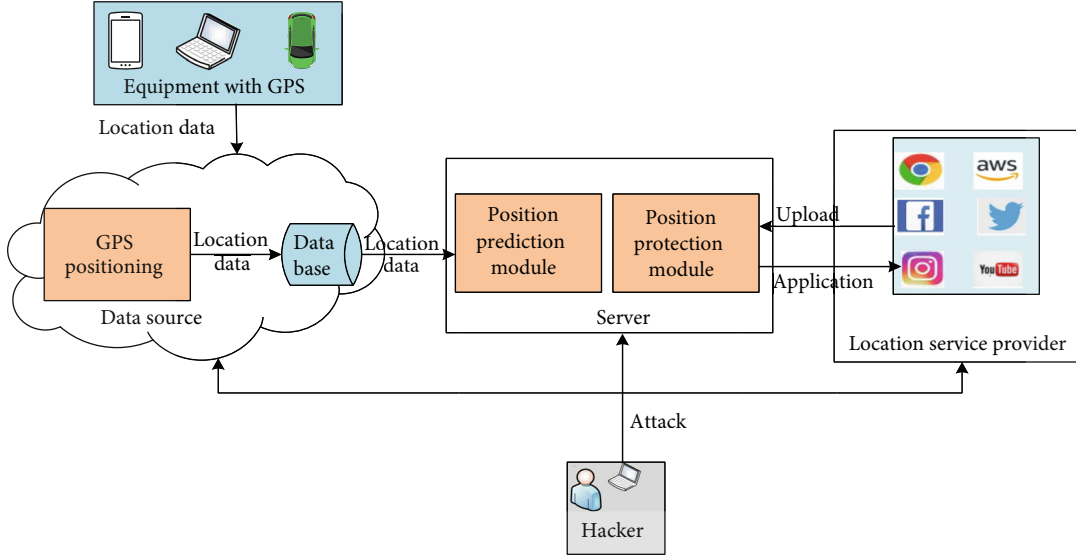


FIGURE 2: Threat model.

3.1. Location Prediction Algorithm Based on the Markov Model. Location prediction enriches and expands LBS, which is of great significance to LBS. The location prediction methods can be mainly divided into three categories: the location prediction method based on linear or nonlinear mathematical model [16], the location prediction method based on frequent track pattern mining [17], and the location prediction method based on the Markov model.

The location prediction method based on the linear or nonlinear mathematical model is to establish a mathematical model according to the current running speed and time to simulate the trajectory of moving objects, thereby predicting the location. The location prediction method based on frequent trajectory pattern mining is to find the frequent trajectory pattern from the user's historical trajectory and then match the current query trajectory with the frequent trajectory pattern to predict the location. The location prediction method based on the Markov model uses the transition probability matrix between states to predict the state of the event and its development trend, so as to predict the user's location.

The n -order Markov model is used to predict the user's next location in this paper. The basic method of Markov prediction is to predict the occurrence and development trend of events by using the transfer probability matrix between states. The Markov model has the advantages of low time complexity and high prediction accuracy, which not only avoids the problem that the user's moving speed and direction are affected by the road network in the first method but also avoids the problem that the query time is too long in the second method, which affects the prediction efficiency and the redundant noise affects the trajectory prediction accuracy.

Location prediction is fundamentally determined by the current location and historical location. Obviously, the historical location that is closer to the current location has the greatest impact on the next location. Therefore, this paper obtains the predicted value of each location based on the Markov model weighting method.

$$X(t) = a_1 S(t-1)P + a_2 S(t-2)P^2 + \dots + a_n S(t-n)P^n. \quad (6)$$

In equation (6), t is the time of the next location, and $t-1$ is the time of the current location. $X(t)$ is a $1 \times n$ matrix that represents the predicted value of each location. $S(i), 1 \leq i \leq n$, is a $1 \times n$ matrix, the value of column i is 1, and the rest is 0. P is an $n \times n$ probability transition matrix. a_1, a_2, \dots, a_k are weights, representing the influence degree of the $1, 2, 3, \dots, n$ locations on the next location decision. Based on the Markov model, this paper proposes a location prediction algorithm. The specific content of the algorithm is as follows:

Analysis shows that Algorithm 1 is a location prediction algorithm based on the Markov model, which contains four modules. First, step 1 to step 5, the one-step transition probability matrix M_1 is obtained according to equation (3). The one-step transition probability matrix indicates that the next predicted position is only related to the current position; secondly, step 6 to step 8, the n -step transition probability matrix is obtained according to equation (5). The n -step transition probability matrix indicates that the predicted next position is related to all historical positions and is comprehensive; thirdly, step 9 to step 12, the predicted value of each position is calculated according to equation (6). Because the closer the historical position has the greater influence on the next position, the weight a is set for each historical position; finally, step 14 outputs the predicted probabilities of all positions.

3.2. Location Protection Algorithm Based on Differential Privacy. In the location protection module, this paper proposes the LPADP algorithm. The basic principle is as follows: Firstly, LPT is constructed for all locations predicted by the LPAM algorithm; secondly, the two nodes with the largest prediction value on LPT are protected by adding Laplacian noise. The algorithm is as follows:

The analysis shows that Algorithm 2 is a location protection algorithm based on differential privacy, which contains three modules. The main purpose of Algorithm 2 is to add

```

Input:  $N = \{N_{ij}\}_{n \times n}$ ; // degree transition matrix
           $S = \{S_i\}$ ; // location at time  $k-1$ 
          now; // current location
           $X = \{X_i\}_{1 \times n}$ ; // estimate each location
           $a = \{a_i\}_{1 \times n}$ ; // weight array
Output: result // output all predicted locations
1. FOREACH  $N_i \in N$ 
2.   sum = cumulate  $N_{ij} \in N_i$ ; //cumulate is an accumulation process
3.   FOREACH  $N_{ij} \in N_i$ 
4.      $P_{ij} = N_{ij}/sum$ ;
5.    $M_1 = P$  // one step transition probability matrix  $P$  is obtained
6. FOR  $i=2$  to  $n$  Do
7.    $M_i = \text{matrixMul}(M_{i-1}, P)$ ; // calculate  $n$ -step transition probability matrix
8. ENDFOR
9. setZero( $X$ ); // clear  $X$  and calculate the estimate
10. FOR  $i=1$  to  $n$  Do
11.   Tepmatrix =  $\text{matrixMul}(S_i, M_i, a_i)$ ; // multiplication of weight and matrix
12.    $X = \text{matrixAdd}(X, \text{Tepmatrix})$ ; // calculate  $X$ 
13. ENDFOR
14.   result = put( $X_1, X_2, \dots, X_n$ ); // output all predicted locations
15. RETURN result;

```

ALGORITHM 1: Location prediction based on the Markov model (LPAM).

```

Input:  $X = (X_i)_{1 \times n}$ ; // location from Algorithm 2
Output: The two locations with the largest prediction probability are protected by adding noise
1. Constructing LPT;
2. void fun(int *X, int *X1, int *X2) // select the two nodes with the largest prediction probability on LPT( $X_1, X_2$ )
3. {
4.   int i;
5.   *max = X[0];
6.   for( $i=1$ ;  $i < \text{strlen}(X)$ ;  $i++$ )
7.     if(*max < *(X + i)) *X1 = *(X + i); // select the nodes with the highest prediction probability  $X_1$ 
8.   *X2 = X[0];
9.   for( $i=1$ ;  $i < \text{strlen}(X)$ ;  $i++$ )
10.    {
11.     if(*X2 < *(X + i) && *X2 < *X1)
12.       *X2 = *(X + i); // select the next largest value node  $X_2$ 
13.    }
14.   result = put( $X_1, X_2$ ); //output  $X_1, X_2$ 
15. }
16.  $\epsilon = \epsilon_1 + \epsilon_2$ ; //  $\epsilon_1 < \epsilon_2$ 
17.  $X'_i = X_i + \text{Lap}(\epsilon_i)$  // Laplacian noise is added to the two location nodes with the largest prediction value

```

ALGORITHM 2: Location protection based on Differential privacy (LPADP).

Laplacian noise to the two position nodes with the largest predicted value for protection. Firstly, the first step is to construct LPT for all positions predicted by Algorithm 1, which LPT is constructed according to the location data and the difficulty of retrieval; secondly, the function of the second step to the fifteenth step is to traverse all positions on the LPT to obtain the node with the largest predicted value and the node with the second largest predicted value. There are two loop functions in the second step to the fifteenth step. Among them, the first position node on the LPT is defaulted to the maximum value

node, and then the nodes on the LPT are traversed in turn, the function of the first loop is to obtain the node X_1 with the largest predicted value (that is, the fourth to seventh steps), and the function of the second loop is to obtain the node X_2 with the second largest predicted value (that is, the eighth to thirteenth steps); finally, the sixteenth to seventeenth steps are to protect the two locations X_1 and X_2 . The sixteenth step is to allocate a reasonable privacy budget to the two locations, and the seventeenth step is to add Laplace noise to X_1 and X_2 according to the privacy budget, so as to protect the two positions.

3.3. Algorithm Analysis

3.3.1. Safety Analysis. The basic principle of differential privacy technology is as follows: when the user submits a query request to the data provider, if the user directly publishes the accurate query results, it may lead to privacy leakage, because the attacker may use the query result to deduct private information. In order to avoid this problem, the differential privacy technology requires a middleware to be extracted from the database, and a specially designed random algorithm is used to inject an appropriate amount of noise into the middleware to obtain a noisy middleware; then, a noisy query result is derived from the noisy middleware and returned to the user. In this way, even if the attacker can deduce the noisy middleware from the noisy result, it is impossible for him to infer the noiseless middleware accurately, let alone infer the original database, so as to achieve the purpose of protecting the user's privacy.

This paper uses differential privacy technology to protect the user's location privacy. The main reason is that differential privacy technology has three major advantages: (1) differential privacy strictly defines the background knowledge of the attacker: except for a certain record, the attacker knows all the information in the original data. Such an attacker is almost the most powerful. In this case, differential privacy can still effectively protect private information; (2) differential privacy has a rigorous statistical model, which greatly facilitates the use of mathematical tools and quantitative analysis and verification; and (3) differential privacy does not require special attack assumptions, does not care about the background knowledge of the attacker, and quantitatively analyzes the risk of privacy leakage.

The main implementation mechanism of differential privacy technology is to add random noise to input or output to protect the privacy of users', such as Laplace mechanism, Gaussian mechanism, and Exponential mechanism. In this paper, Laplacian mechanism is used to protect the user's location by adding Laplacian noise.

Laplacian noise is essentially a group of random values satisfying the Laplacian distribution, and the basic principle is to add noise that obeys Lap (b) to the original data and statistical results, so that the query results after adding the noise meet the differential privacy constraint effect. Laplacian noise is added in the LPADP algorithm, which conforms to ϵ -differential privacy. The proof process is as follows:

It can be known from the probability density function of the laplace mechanism:

$$\begin{aligned} \frac{P_x(z)}{P_y(z)} &= \prod_{i=1}^k \frac{e^{-\epsilon|f(x)_i - z_i|/\Delta f}}{e^{-\epsilon|f(y)_i - z_i|/\Delta f}} = \prod_{i=1}^k e^{\frac{\epsilon(|f(y)_i - z_i| - |f(x)_i - z_i|)}{\Delta f}} \\ &\leq \prod_{i=1}^k e^{\frac{\epsilon(|f(x)_i - f(y)_i|)}{\Delta f}} \leq e^{\frac{\epsilon\|f(x) - f(y)\|_1}{\Delta f}} = e^\epsilon. \end{aligned} \quad (7)$$

According to the definition of differential privacy, the LPADP algorithm proposed in this paper satisfies ϵ -differential privacy.

3.3.2. Complexity Analysis. Assuming that the location data table contains n pieces of records data. The privacy protection module in Figure 1 mainly contains two modules: prediction module and location protection module. So, the complexity of the algorithm in this paper mainly includes two aspects: the time complexity of the LPAM algorithm in the prediction module and the LPADP algorithm in the location protection module. The LPAM algorithm mainly uses the n -order Markov model to predict the position.

The realization of the LPAM algorithm mainly includes three parts: first, calculate the one-step transition probability matrix according to formula (3), and its time complexity is $O(n^2)$, reflected in the first to fifth steps of Algorithm 1; secondly, calculate the n -step transition probability matrix according to formula (5), and its time complexity is $O(n)$, reflected in the sixth to eighth steps of Algorithm 1; finally, calculate and output the predicted value of each position according to formula (6), and its time complexity is $O(n)$, reflected in the tenth to fourteenth steps of Algorithm 1.

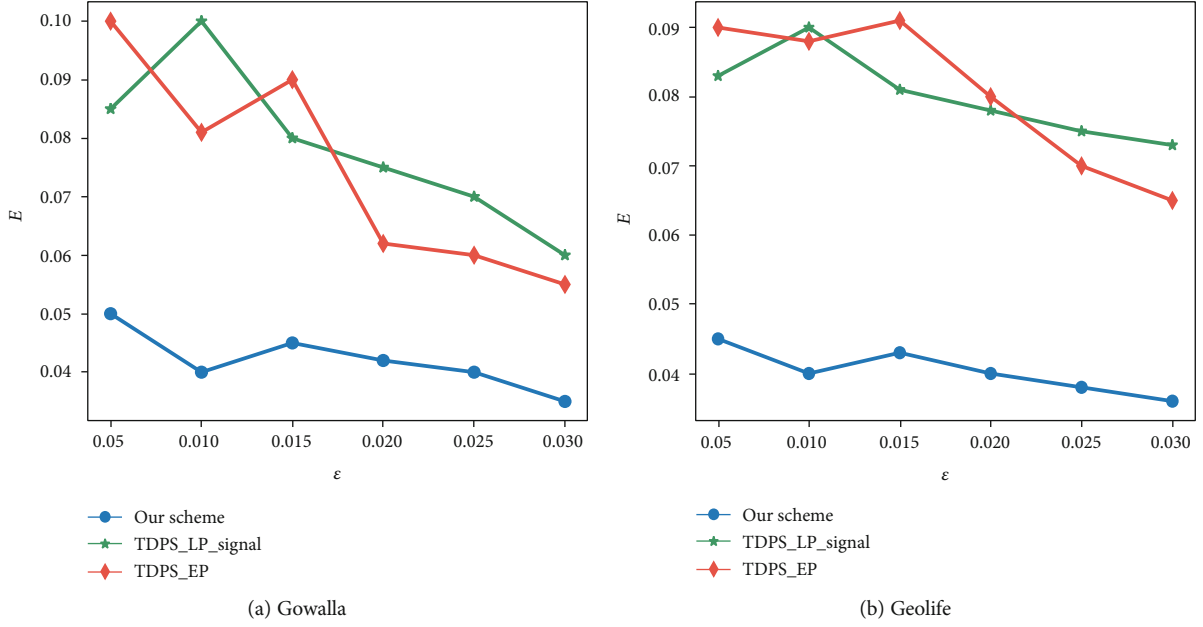
The LPADP algorithm mainly allocates a reasonable privacy budget to the two locations with the larger predicted value on the LPT and then adds Laplacian noise to protect the location privacy. The realization of the LPADP algorithm mainly includes three parts: first, construct LPT for all positions predicted by Algorithm 1, and its time complexity is $O(n)$, reflected in the first step of Algorithm 2; secondly, traverse all the position nodes on the LPT and then select the two nodes with the largest predicted value, and the time complexity is $O(n)$, reflected in the second to the fifteenth steps of Algorithm 2; finally, a reasonable privacy budget is allocated to the two nodes with the largest predicted value, Laplacian noise is added for protection, and the time complexity is $O(1)$, reflected in the sixteenth to seventeenth steps of Algorithm 2. In general, the time complexity required in this article is

$$O(n^2) + O(n) + O(n) + O(n) + O(n) + O(1) \approx O(n^2). \quad (8)$$

4. Experimental Results and Analysis

4.1. Environment Configuration. In order to test the performance of the location privacy protection method proposed in this paper, the algorithm has been fully experimented in terms of data availability, privacy protection degree, and algorithm running time. The experiment is implemented using Python, and the data sets are Gowalla data set and Geolife data set [18, 19]. The experimental environment of this article is PyCharm. The hardware environment is 2.60GHz i7 CPU, 8.00RAM, Win10 system 64-bit.

4.2. Data Availability Analysis. For the same query function Q , the similarity of the output query results before and after the noise is added to the data that reflects the influence of the privacy protection algorithm on the availability of the data. Let $G(Q)$ be the query result of the data before adding noise, and $G'(Q)$ be the query result of the data after adding noise, and then the degree of approximation S_Q can be

FIGURE 3: The effect of ϵ on data availability.

defined as the first-order normal form distance between the two output results $S_Q = \|G(Q) - G'(Q)\|_1$.

For continuous query $Q_i \in \{Q_1, Q_2, \dots, Q_k\}$, the availability of data published by location services is defined as

$$E = \frac{1}{Q_i} \sum \left(\frac{S_{Q_i}}{e^{\epsilon/2}} \right). \quad (9)$$

Comparing the method proposed in this paper with TDPS_LP_Signal and TDPS_EP [20] in terms of data availability, the results are shown in Figure 3. The X-axis represents the value of ϵ , and the Y-axis represents the data availability. The E value of the three algorithms will decrease with the increase of ϵ , because the larger the ϵ , the smaller the noise addition and the better the data availability. When $\epsilon > 0.015$, the data availability of the method proposed in this paper tends to be stable. Therefore, the method proposed in this paper has better data availability compared with TDPS_LP_Signal and TDPS_EP. The data availability of the TDPS_LP_Signal algorithm is between the algorithm proposed in this paper and TDPS_EP, and the data availability of TDPS_EP is relatively poor.

Comparing the Markov model with the trajectory mining model and linear or nonlinear mathematical model in data availability, the results are shown in Figure 4. The X-axis represents the number of historical locations, and the Y-axis represents data availability. The E value of the three algorithms will decrease with the increase in the number of historical locations, because the increase in the number of historical locations, the more accurate the prediction and the better the data availability. The n -order Markov model is more accurate and comprehensive in location prediction; so, it has better data availability. The trajectory mining model is between the Markov model and linear or nonlinear mathematical model in terms of data availabil-

ity, and the data availability of the linear or nonlinear mathematical model is poor.

4.3. Analysis of the Degree of Privacy Protection. Comparing the method proposed in this paper with TDPS_LP_Signal and TDPS_EP in terms of privacy protection degree, the result is shown in Figure 5. The X-axis represents the value of ϵ , and the Y-axis represents the degree of privacy protection. The degree of privacy protection of the three algorithms will decrease with the increase of ϵ , because the larger the ϵ , the smaller the noise addition and the worse the degree of privacy protection. The algorithm proposed in this paper uses differential privacy technology to protect the location and has better security. The degree of privacy protection of the TDPS_LP_Signal algorithm is between the algorithm proposed in this paper and the TDPS_EP algorithm, and the degree of privacy protection of the TDPS_EP algorithm is relatively low.

Comparing the Markov model with the trajectory mining model and linear or nonlinear mathematical model in the degree of privacy protection, the result is shown in Figure 6. The X-axis represents the number of historical locations, and the Y-axis represents the degree of privacy protection. The degree of privacy protection of the three algorithms will increase with the increase in the number of historical locations, because the increase in the number of historical locations, the more accurate the prediction and the better the degree of privacy protection. The n -order Markov model is more accurate and comprehensive in location prediction and has better security. The privacy protection degree of the trajectory mining mode algorithm is between the Markov model and the linear and nonlinear mathematical model. The privacy protection degree of the linear or nonlinear mathematical model algorithm is relatively low.

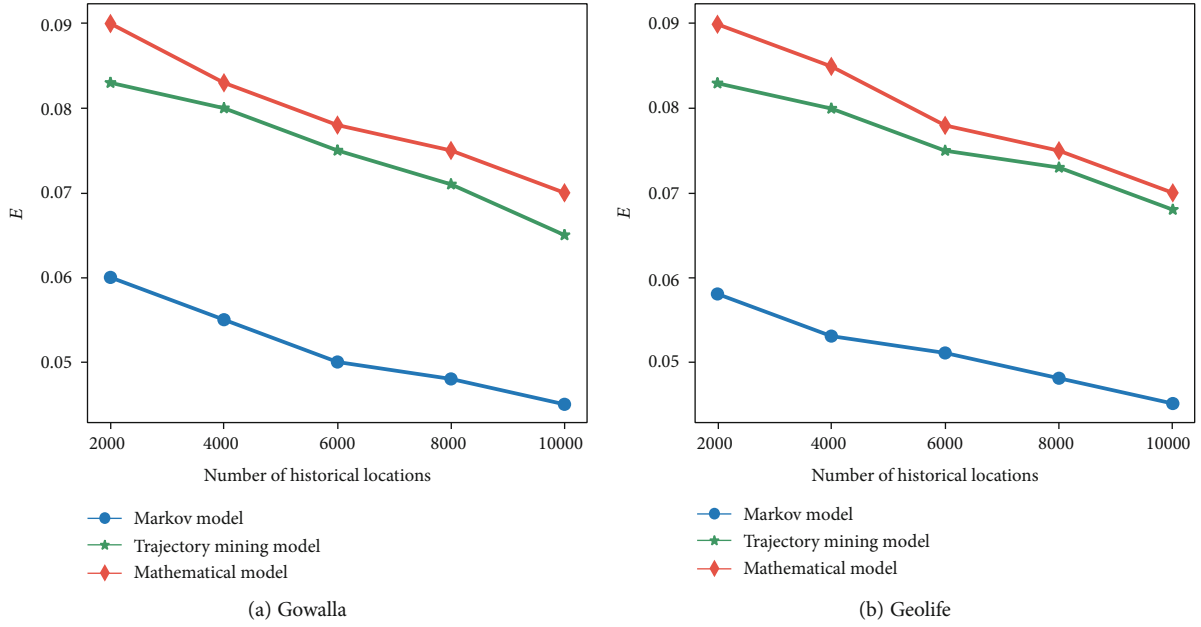
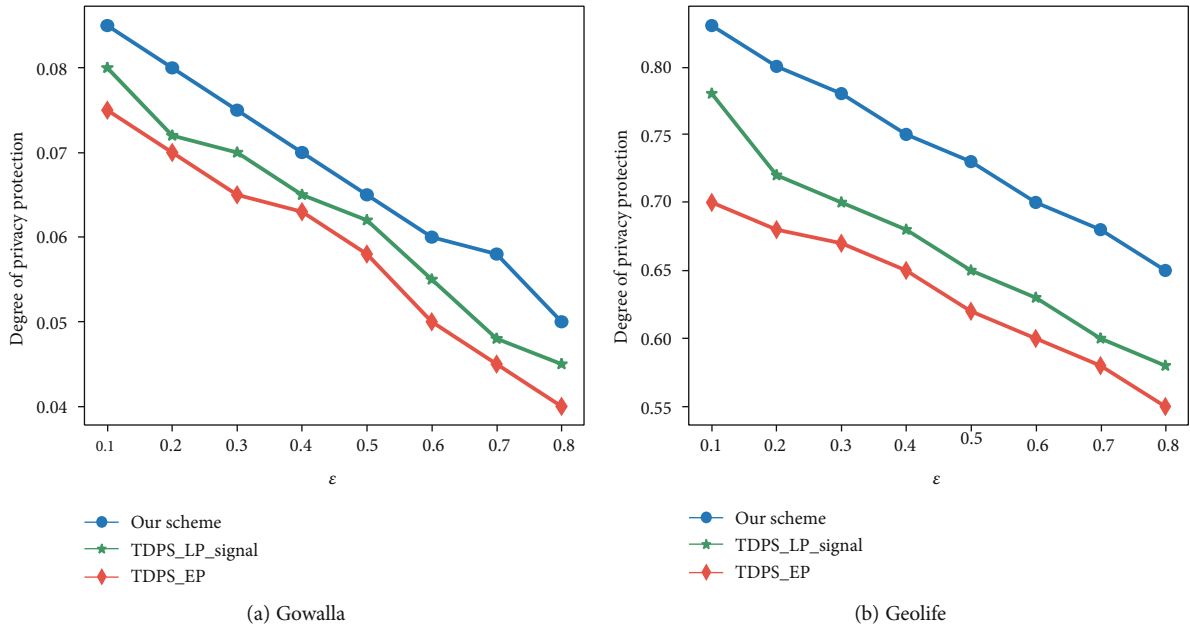


FIGURE 4: The impact of the number of historical locations on data availability.

FIGURE 5: The effect of ϵ on the degree of privacy protection.

4.4. Analysis of Algorithm Running Time. Comparing the method proposed in this paper with TDPS_LP_Signal and TDPS_EP in terms of algorithm running time, the result is shown in Figure 7. The X-axis represents the value of ϵ , and the Y-axis represents the running time of the algorithm. The running time of the three algorithms will decrease with the increase of ϵ , because the larger the ϵ , the smaller the noise addition and the shorter the running time. The method proposed in this paper only protects the two locations with the largest predicted value and has less algorithm running time. The running time of the TDPS_LP_Signal algorithm is between the algorithm proposed in this paper and the

TDPS_EP algorithm, and the TDPS_EP algorithm requires relatively more time.

Comparing the Markov model with the trajectory mining model and linear or nonlinear mathematical model in terms of algorithm running time, the results are shown in Figure 8. The X-axis represents the number of historical locations, and the Y-axis represents the running time of the algorithm. The running time of the three algorithms will increase as the number of historical locations increases, because the number of historical locations increases, the prediction time increases, thereby increasing the running time. Because the Markov model has the advantage of low time complexity, it

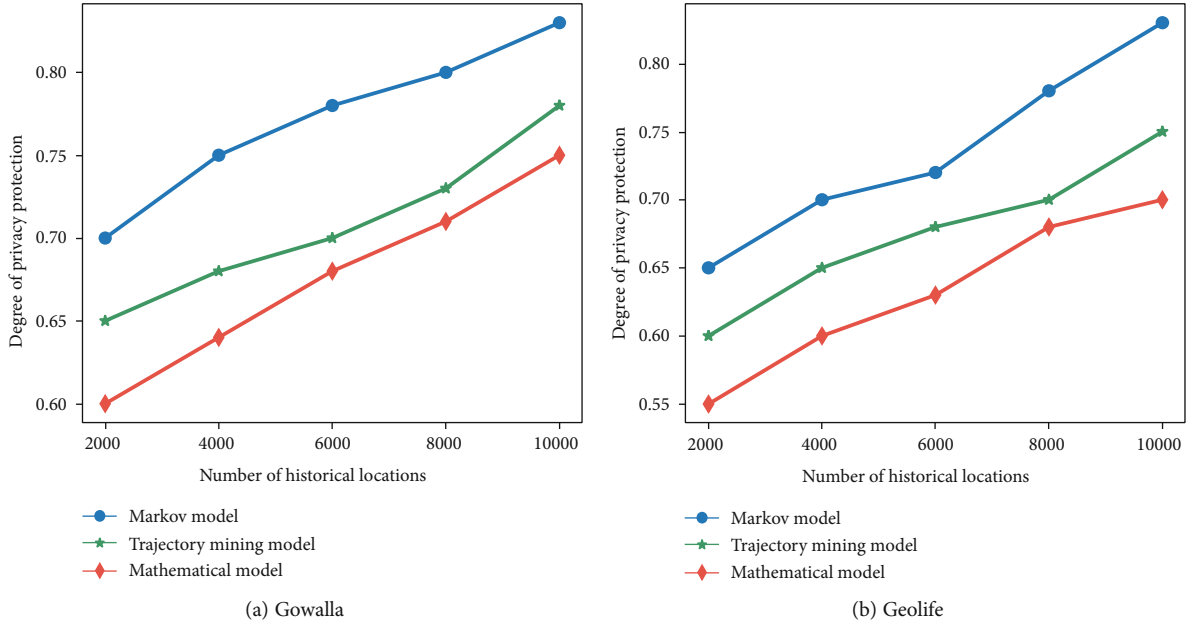


FIGURE 6: The influence of the number of historical locations on the degree of privacy protection.

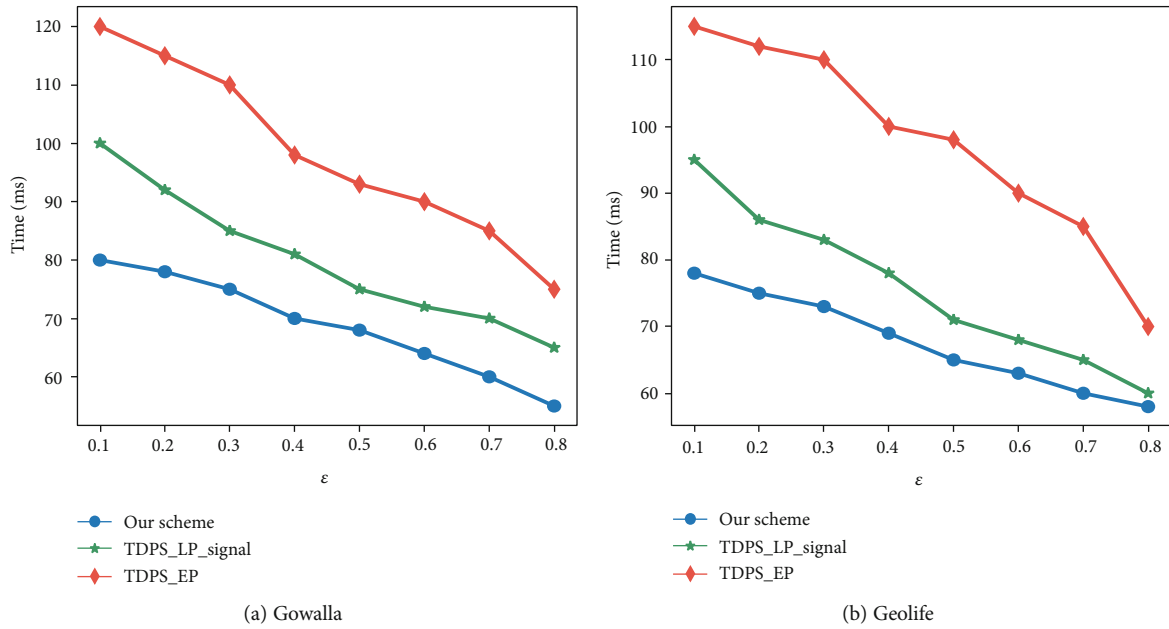


FIGURE 7: The effect of ϵ on the running time of the algorithm.

has less algorithm running time. The running time of the trajectory mining pattern algorithm is between the Markov model and linear and nonlinear mathematical model, and linear and nonlinear mathematical model algorithm takes more time.

5. Related Work

As LBS has privacy that becomes the focus of research, more and more scholars have paid close attention to LBS privacy protection methods. At present, the main methods of loca-

tion privacy protection include cryptography, k -anonymity, and differential privacy.

Cryptography is a privacy protection method based on encryption and signature, which realizes privacy protection by encrypting users' information [21–23]. Liang et al. proposed a privacy protection method based on POI query in the road network environment by combining Hilbert curve with anonymous technology, which effectively avoided inference attack against location information [24]. While it is known that unconditionally secure position-based cryptography is impossible both in the classical and the quantum setting, it

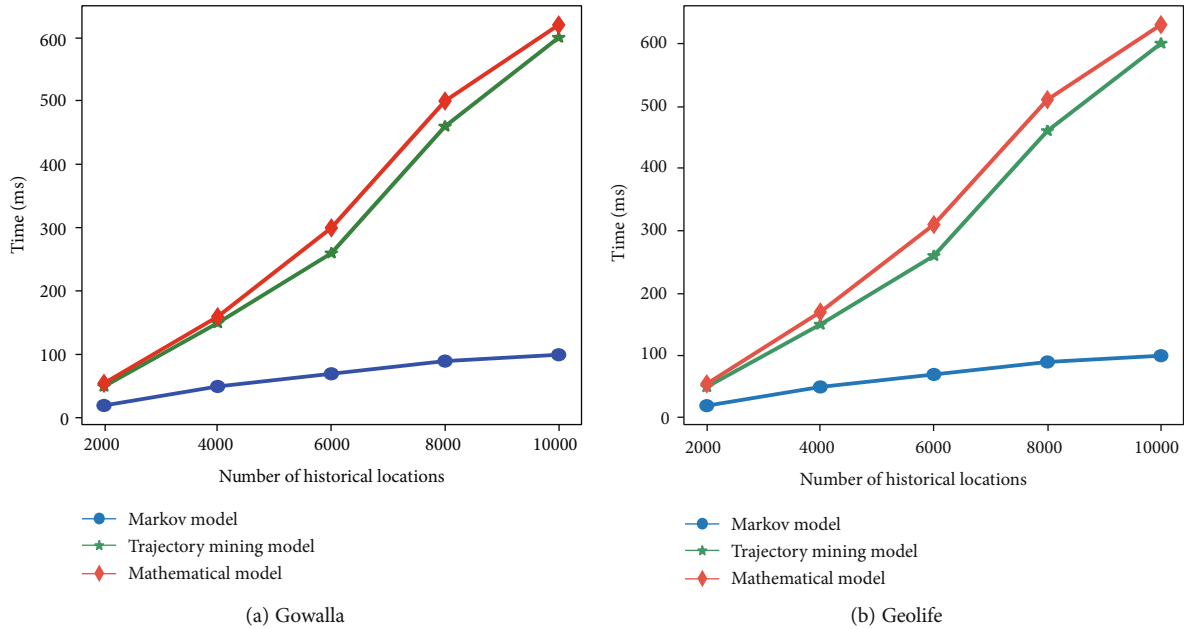


FIGURE 8: The effect of the number of historical locations on the running time of the algorithm.

has been shown that some quantum protocols for position verification are secure against attackers which share a quantum state of bounded dimension. Bluhm et al. considered the security of the qubit routing protocol. The protocol has the advantage that an honest prover only has to manipulate a single qubit and a classical string of length $2n$ and shows that the protocol is secure if each of the attackers holds at most $n/2 - 3$ qubits [25]. However, cryptography is difficult to implement because of huge computing and communication costs.

k -anonymity requires that the same quasiidentifier must have at least k records, and each individual record cannot be distinguished from other $k-1$ individuals for sensitive attributes; so, the attackers cannot link the records through the quasiidentifier [26–28]. In reference [29], the user’s real location was replaced by the anonymous users’ area; so, the attacker could not identify the user’s real location. In reference [30], the users used historical information to process real information anonymously, so as to protect users’ location privacy. In reference [31], the users cooperated with each other, shared part of the location information, and formed an anonymous space to achieve the effect of k -anonymity. Mingyan et al. [32] proposed a location anonymity algorithm based on the mobile P2P structure, which avoided the risk of information leakage caused by single point failure. Xingyou et al. [33] selected the location anonymous set in the grid that published the request according to the real service request data and sent the location anonymous set to the server instead of the user’s real location. Although k -anonymity technology can prevent the disclosure of identity, it cannot resist homogeneous attacks and background knowledge attacks.

Differential privacy can protect privacy effectively and has a rigorous statistical model [34–36]. Zhiqiang et al. [37] proposed a location data acquisition scheme based on local differential privacy, which used the random response mech-

anism to obtain location data, and the data collector used direct statistics and expectation maximum method to analyze the location data to ensure that the normal analysis can be carried out. In order to solve the problem of privacy leakage in crowdsourcing, Zheng et al. [38] proposed a crowdsourcing location data acquisition scheme that satisfied the localized differential privacy. In this scheme, the road network space was divided into Voronoi diagram, and a method of spatial range query on disturbed data set was designed. Fuzzy C-means clustering algorithm is one of the typical clustering algorithms in data mining applications. However, due to the sensitive information in the dataset, there is a risk of user privacy being leaked during the clustering process. Zhang et al. [39] aimed at the problem that the algorithm accuracy is reduced by randomly initializing the membership matrix of fuzzy C-means; in this paper, the maximum distance method is firstly used to determine the initial center point. Then, the Gaussian value of the cluster center point is used to calculate the privacy budget allocation ratio. Additionally, Laplace noise is added to complete differential privacy protection. Wei et al. [40] proposed a differential privacy-based location protection (DPLP) scheme, and DPLP splits the exact locations of both workers and tasks into noisy multilevel grids by using adaptive three-level grid decomposition (ATGD) algorithm and DP-based adaptive complete pyramid grid (DPACPG) algorithm, respectively, thereby considering the grid granularity and location privacy. Furthermore, DPLP adopts an optimal greedy algorithm to calculate a geocast region around the task grid, which achieves the trade-off between acceptance rate and system overhead, which protects the location privacy of both workers and tasks, and achieves task allocation with high data utility.

In view of the problem of location privacy protection, this paper uses differential privacy technology to protect the location privacy of users’. Differential privacy can not only resist the background knowledge attack and homogeneous

attacks but also can effectively protect the user's privacy when adding or deleting a record without affecting the query result.

6. Conclusions

The continuous use of LBS will expose the user's location information, which results in the disclosure of user's privacy. In order to solve issues of user privacy disclosure in LBS, a differential privacy location protection method based on the Markov model is proposed in this paper. Experiments show that this method can protect location privacy effectively and has high data availability and low time complexity. In the future research, the research mainly focuses on two aspects. On the one hand, the location prediction of the Markov model does not consider the situation of new users; so, the future research direction is to predict the location of new users and protect the predicted location information. On the other hand, the Markov model predicts and protects the position, which realizes the direct protection of the position, but ignores the spatiotemporal correlation between the predicted positions. Therefore, the future research direction is to protect the position indirectly according to the spatiotemporal correlation between the predicted positions.

Data Availability

All data, models, and codes generated or used during the study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Key Research and Development Project of Shandong Province under grant no. 2019JZZY010134, the Natural Science Foundation of Shanxi Province under grant no. 201901D111280, and the Scientific and Technological Innovation Project in Colleges and Universities of Shanxi Province under grant no. 2019L0459.

References

- [1] H. Huang, G. Gartner, J. M. Krisp, M. Raubal, and N. van de Weghe, "Location based services: ongoing evolution and research agenda," *Journal of Location Based Services*, vol. 12, no. 2, pp. 63–93, 2018.
- [2] M. Yu, G. Fan, H. Yu, and L. Chen, "Location-based and time-aware service recommendation in mobile edge computing," *International Journal of Parallel Programming*, vol. 2, 2021.
- [3] A. Aloui, O. Kazar, S. Bourekkache, and F. Omary, "An efficient approach for privacy-preserving of the client's location and query in M-business supplying LBS services," *International Journal of Wireless Information Networks*, vol. 27, no. 3, pp. 433–454, 2020.
- [4] A. Sz and C. B. Xin, "Multiple-user closest keyword-set querying in road networks," *Information Sciences*, vol. 509, pp. 133–149, 2020.
- [5] Q. Zeng, X. Han, and Y. M. Cao, "Integrated public key encryption and public key encryption with keyword search," *Computer and Modernization*, vol. 284, no. 4, pp. 107–111, 2019.
- [6] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," in *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems - PODS '98*, pp. 188–202, Washington, 1998.
- [7] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography. TCC 2006*, S. Halevi and T. Rabin, Eds., vol. 3876 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2006.
- [8] E. Akinola and S. S. Daodu, "Location prediction in the Long term evolution network using ST-RNN and Markov model," *International Journal of Computer Applications*, vol. 176, no. 30, pp. 14–17, 2020.
- [9] A. Rahimifar, "Predicting the energy consumption in software defined wireless sensor networks: a probabilistic Markov model approach," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2020.
- [10] C. Xia, J. Hua, W. Tong, and S. Zhong, "Distributed K-Means clustering guaranteeing local differential privacy," *Computers & Security*, vol. 90, p. 101699, 2020.
- [11] Z.-q. Gao, X.-l. Cui, S. Zhou, and C. Yuan, "Local differential privacy protection and its application," *Journal of Computer Engineering and Science*, vol. 40, no. 6, pp. 1029–1036, 2018.
- [12] J. Sharma, D. Kim, A. Lee, and D. Seo, "On differential privacy-based framework for enhancing user data privacy in mobile edge computing environment," *IEEE Access*, vol. 9, pp. 38107–38118, 2021.
- [13] S. Chen, A. Fu, S. Yu, H. Ke, and M. Su, "DP-QIC: a differential privacy scheme based on quasi-identifier classification for big data publication," *Soft Computing*, vol. 25, pp. 7325–7339, 2021.
- [14] M. N. Cakir, M. Saleemi, and K. H. Zimmermann, "On the theory of stochastic automata," 2021, <https://arxiv.org/abs/2103.14423>.
- [15] A. Niessl, A. Allignol, C. Mueller, and J. Beyersmann, "Estimating state occupation and transition probabilities in non-Markov multi-state models subject to both random left-truncation and right-censoring," 2020, <https://arxiv.org/abs/2004.06514>.
- [16] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2019.
- [17] X. Pan, Q. Zhao, and P. Zhao, "Frequent trajectory of pattern mining with spatio-temporal attribute and relationship label," *Computer Engineering and Applications*, vol. 55, no. 10, pp. 83–89, 2019.
- [18] K. Cao, Q. Sun, H. Liu, Y. Liu, G. Meng, and J. Guo, "Social space keyword query based on semantic trajectory," *Neurocomputing*, vol. 428, pp. 340–351, 2020.
- [19] H. Luo, H. Zhang, S. Long, and Y. Lin, "Enhancing frequent location privacy-preserving strategy based on geo-Indistinguishability," *Multimedia Tools and Applications*, vol. 80, no. 14, pp. 21823–21841, 2021.
- [20] H. Kang, S. Zhang, and Q. Jia, "A method for time-series location data publication based on differential privacy," *Wuhan*

- University Journal of Natural Sciences*, vol. 24, no. 2, pp. 107–115, 2019.
- [21] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, “Enabling efficient and geometric range query with access control over encrypted spatial data,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 870–885, 2019.
- [22] M. Etemad, A. Küpçü, C. Papamanthou, and D. Evans, “Efficient dynamic searchable encryption with forward privacy,” *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 1, pp. 5–20, 2018.
- [23] C. Cui, F. Li, T. Li, J. Yu, R. Ge, and H. Liu, “Research on direct anonymous attestation mechanism in enterprise information management,” *Enterprise Information Systems*, vol. 15, no. 4, pp. 513–529, 2021.
- [24] H. C. Liang, B. Wang, N. N. Cui, K. Yang, and X. C. Yang, “Privacy preserving method for point-of-interest query on road network,” *Journal of Software*, vol. 29, no. 3, pp. 703–720, 2018.
- [25] A. Bluhm, M. Christandl, and F. Speelman, “Position-based cryptography: single-qubit protocol secure against multi-qubit attacks,” 2021, <https://arxiv.org/abs/2104.06301>.
- [26] A. T. Truong, “Privacy preserving spatio-temporal databases based on k-anonymity,” *Science & Technology Development Journal - Engineering and Technology*, vol. 3, no. SI1, pp. SI82–SI94, 2020.
- [27] B. S. Kumar, T. Daniya, N. Sathya et al., “Investigation on privacy preserving using K-anonymity techniques,” in *2020 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2020.
- [28] T. K. Esmeel, M. M. Hasan, M. N. Kabir, and A. Firdaus, “Balancing data utility versus information loss in data-privacy protection using k-anonymity,” in *2020 IEEE 8th Conference on Systems, Process and Control (ICSPC)*, Melaka, Malaysia, 2020.
- [29] S. Zhang, X. Li, Z. Tan, T. Peng, and G. Wang, “A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services,” *Future Generation Computer Systems*, vol. 94, pp. 40–50, 2019.
- [30] Z. Wu, G. Li, S. Shen, X. Lian, E. Chen, and G. Xu, “Constructing dummy query sequences to protect location privacy and query privacy in location-based services,” *World Wide Web*, vol. 24, pp. 25–49, 2020.
- [31] D.-d. Wu and L. Xin, “Location anonymous algorithm based on user collaboration under distributed structure,” *Computer Science*, vol. 46, no. 4, pp. 158–163, 2019.
- [32] M.-y. Xu, Z. Hua, J. Xinsheng, and S. Juan, “Distribution-perceptive-based spatial cloaking algorithm for location privacy in mobile peer-to-peer environments,” *Journal of Software*, vol. 29, no. 7, pp. 1852–1862, 2018.
- [33] X.-Y. Xia, Z. -H. Bai, J. Li, and R. -Y. Yu, “A location cloaking algorithm based on dummy and Stackelberg game,” *Chinese Journal of Computers*, vol. 42, no. 10, pp. 2216–2232, 2019.
- [34] X. Zhao, D. Pi, and J. Chen, “Novel trajectory privacy-preserving method based on prefix tree using differential privacy,” *Knowledge-Based Systems*, vol. 198, p. 105940, 2020.
- [35] X. Niu, H. Huang, and Y. Li, “A real-time data collection mechanism with trajectory privacy in mobile crowd-sensing,” *IEEE Communications Letters*, vol. 24, no. 10, pp. 2114–2118, 2020.
- [36] F. O. Olowononi, D. B. Rawat, and C. Liu, “Federated learning with differential privacy for resilient vehicular cyber physical systems,” in *2021 IEEE 18th Annual Consumer Communica-*
- tions & Networking Conference (CCNC)*, Las Vegas, NV, USA, 2021.
- [37] G. A. Zhiqiang, C. U. Xiaolong, D. U. Bo, Z. H. Sha, Y. U. Chen, and L. I. Ai, “Collection scheme of location data based on local differential privacy,” *Journal of Tsinghua University: Science and Technology*, vol. 59, no. 1, pp. 23–27, 2019.
- [38] Z. Huo, K. Zhang, P. He, and Y. Wu, “Crowdsourcing location data collection for local differential privacy,” *Journal of Computer Applications*, vol. 39, no. 3, pp. 763–768, 2019.
- [39] Y. Zhang and J. Han, “Differential privacy fuzzy C-means clustering algorithm based on gaussian kernel function,” *PLoS One*, vol. 16, no. 3, article e0248737, 2021.
- [40] J. Wei, Y. Lin, X. Yao, and J. Zhang, “Differential privacy-based location protection in spatial crowdsourcing,” *IEEE Transactions on Services Computing*, 2019.