Wireless Communications and Mobile Computing

Intelligent and Flexible Security of Next-Generation Wireless Networks

Lead Guest Editor: Zengpeng Li
Guest Editors: Chenglu Jin and Nan Li

Intelligent and Flexible Security of Next-Generation Wireless Networks

Wireless Communications and Mobile Computing

Intelligent and Flexible Security of Next-Generation Wireless Networks

Lead Guest Editor: Zengpeng Li
Guest Editors: Chenglu Jin and Nan Li

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 6039034, 26 pages
https://doi.org/10.1155/2023/6039034

Research Article

Efficient Secure Computation from SM Series Cryptography

Yibiao Lu,[1,2] **Zecheng Wu**,[1,2] **Bingsheng Zhang**,[1,2] **and Kui Ren**[1,2]

[1]*ZJU-Hangzhou Global Scientific and Technological Innovation Center, China*
[2]*Zhejiang University, China*

Correspondence should be addressed to Bingsheng Zhang; bingsheng@zju.edu.cn

Received 26 July 2022; Revised 22 November 2022; Accepted 15 April 2023; Published 17 May 2023

Academic Editor: Nan Li

The wireless network suffers from many security problems, and computation in a wireless network environment may fail to preserve privacy as well as correctness when the adversaries conduct attacks through backdoors, steganography, kleptography, etc. Secure computation ensures the execution security in such an environment, and compared with computation on the plaintext, the performance of secure computation is bounded by the underlying cryptographic algorithms and the network environment between the involved parties. Besides, the Chinese cryptography laws require the cryptographic algorithms that appeared in the commercial market to be authorized. In this work, we show how to implement oblivious transfer (OT), an important primitive in secure multiparty computation (MPC), using the Chinese government-approved SM2 and SM3 algorithms. The SM2 algorithm is based on the elliptic curve cryptography and is much faster than the discrete logarithm-based solutions. Moreover, by adopting the standard OT extension technique, we can extend the number of OTs efficiently with one more round of communication and invocations to the SM3 and SM4 algorithms. The OT primitive can be used in the Beaver multiplication triple generation and other MPC protocols, e.g., private set intersection. Therefore, we can utilize the SM series cryptography, specifically, the SM2, SM3, and SM4 algorithms, to build highly efficient secure computation frameworks which are suitable for the wireless network environment and for commercial applications in China. The experimental evaluation results show that our protocols have comparable performance to existing protocols; specifically, our protocols are quite suitable for bad network environments.

1. Introduction

Wireless network (WLN) enables devices to communicate with each other without cable connections, and it is a major component of the modern Internet. With the advancement of the Internet of things (IoT) technique, a vast amount of wireless networks are being deployed [1]. However, the wireless network can be quite vulnerable [2, 3], an adversary may eavesdrop on or alter the communication in the network. When several parties want to perform a joint computation in a wireless network with potential adversaries, the correctness of the computation and the privacy of inputs can be easily breakdown. Although efforts have been made to avoid or mitigate the security threat in the wireless network [4–6], there are still a lot of security issues. Therefore, we need privacy-enhancing technologies to ensure security in such a network environment.

In this work, we leverage a cryptographic technique called secure multiparty computation (MPC) [7, 8] to construct an efficient and provably secure computation framework that protects parties' privacy in a network potentially controlled by the adversary. The goal of MPC is to design protocols that enable several mutually untrusted parties to jointly compute a function on their private inputs without revealing anything except for the function output. Typically, the computational security of a MPC protocol relies on some computational or setup assumptions. Therefore, if there is an adversary that can break the security of the protocol, either the adversary has unbounded computation power or the computational assumption does not hold.

We focus on the oblivious transfer (OT) primitive, which is complete for the MPC computation [9] and is a fundamental building block for many MPC protocols. Naor and Pinkas [10] provide the first efficient OT protocol, and Chou and Orlandi [11] propose the simplest OT construction. Specifically, in [12], Masny and Rindal initiate the study of a new OT primitive called endemic OT. The endemic OT is weaker than commonly used OT in the sense that a corrupted party is able to control its output message, and the same definition has also been considered by Garg et al. [13]. As demonstrated by Masny and Rindal [12], endemic OT can be constructed using proper key agreement protocols in the programmable random oracle model. Later, McQuoid et al. [14] improve the efficiency of 1-out-of-N endemic OT protocol using a programmable-once public function (POPF). They also notice some security issues in the batch setting and provide a proper treatment in [15].

In the implementation, the main factor affecting the performance of a MPC protocol is the involved cryptographic primitives, whose performance depends on the underlying cryptographic assumptions. A wide range of computational MPC protocols are built on asymmetric-key cryptosystems, and as for asymmetric-key cryptography, the Rivest-Shamir-Adleman (RSA) based on integer factorization and elliptic curve cryptography (ECC) based on discrete logarithm are two of the most important algorithms being used. Evaluation results show that ECC has great advantages over RSA in both computation time [16] and resource consumption [17]. In 2010, the Chinese State Cryptography Administration announced the public key cryptographic algorithm SM2 [18] and several other SM series algorithms, which are based on the elliptic curve cryptography. According to the Chinese cryptography laws [19], it is mandatory to adopt cryptography algorithms which have been authorized for commercial use in China, e.g., the SM algorithms.

1.1. Our Contribution. The contributions of this work can be summarized as follows. First, we construct an endemic OT protocol based on the SM2 key agreement protocol. Moreover, we build several MPC protocols on the top of the endemic OT protocol, including a two-party secure computation protocol on the Boolean circuits, a multiparty party secure computation protocol on arithmetic circuits, and a two-party private set intersection (PSI) protocol. Our constructions consider both efficiency and availability and only use the SM series cryptography. The security of our protocols can be proved in the random oracle model and in the public key infrastructure (PKI) setting. Since a PKI is used, the parties can communicate without the secure channel functionality, and in implementation, the parties can transmit messages without the TLS protocol. To the best of our knowledge, we are the first to propose a secure computation framework that complies with the Chinese national standards and regulations.

2. Related Work

There have been some secure computation solutions for the wireless network. We can categorize them into low-communication MPC and hardware-based computation.

Garay et al. [20] investigated the feasibility of designing protocols with sublinear communication complexity. Their work enables large-scale secure computation in a communication-restricted environment. Moreover, Gentry et al. proposed a communication model called YOSO [21], in which each party only sends one message to others. In YOSO MPC, only a small fraction of parties execute and communicate in each round; therefore, its communication complexity is also sublinear to the total amount of involved parties. Fully homomorphic encryption is a widely used primitive in low-communication MPC protocol design. Asharov et al. [22] constructed a MPC protocol using threshold FHE. The proposed protocol only needs two or three communication rounds depending on the underlying assumption, and its communication cost is independent of the function to be computed. López-Alt et al. [23] and Mukherjee and Wichs [24] considered to use multikey FHE and a third-party untrusted server in their construction, the proposed protocol achieves a minimal communication complexity which is independent of the function and the number of parties. All these FHE-based protocols have the property that the communication size only depends on the input/output size; however, using FHE dramatically increases the computation burden. Another way to reduce communication in the protocol execution is to offload the major communication workload to a preprocessing phase. Damgård et al. proposed the celebrated SPDZ computation framework [25], in which a bunch of authenticated triples are generated in the preprocessing phase and consumed in the online phase. The authenticated garbling protocols proposed by Wang et al. [26, 27] also use authenticated triples to speed up the online computation. Specifically, these garbled circuit-based protocols have constant round complexity, which makes them more suitable for the wireless network. Carter et al. [28] noticed that a remote server can be used to instantiate the preprocessing phase. However, in such a server-aided setting, the adversaries are allowed to corrupt the cloud server.

Hardware-based computation has many things in common with the server-aided computation, and the efficiency of the protocols varies with the parties' trust to the hardware. Since the hardware can have a fast connection with the computing parties, hardware-based computation is typically much faster than other solutions [29–31]. Generally, hardware-based computation uses a hardware token or trusted hardware issued by the parties or a hardware manufacturer, and the parties can use the hardware to generate preprocessing information or even directly compute the function. These hardware-based protocols assume that the hardware is tamper-resistant or tamper-proof, while in practice, there have been works that successfully break the security of some commonly used hardware [32, 33].

3. Preliminaries

3.1. Notations. Throughout this paper, we use the following notations and terminologies. Let λ be the computational security parameter, and μ be the statistical security parameter. Denote a binary matrix with a rows and b columns as $\{0,1\}^{a\times b}$. When A and B are two-bit strings, $A\|B$ is the concatenation of them. When A is a bit string, a vector, or an array, $A[i]$ is its i-th element. Denote the set $\{a, a+1, \cdots, b\}$ by $[a, b]$, let $[b]$ denote $[1, b]$, and let $\varnothing$ denote the empty set. When a set $A = \{a_i\}_{i\in[n]}$ is used, we assume the elements are arranged by the indexes as $a_1, \cdots, a_n$. When A is a set, $a \longleftarrow A$ stands for sampling a uniformly at random from A, and $|A|$ stands for the size of A in terms of the number of elements. When A is a matrix, A^i denotes its i-th column, and A_j denotes its j-th row. When A is a randomized algorithm, $y \longleftarrow A(x)$ stands for running A on input x with a fresh random coin r; when needed, we denote $y \coloneqq A(x\,;r)$ as running A on input x with the explicit random coin r. Let $\mathrm{poly}(\cdot)$ and $\mathrm{negl}(\cdot)$ be a polynomially bounded function and negligible function, respectively. We assume each party has a unique PID. For readability, we refer P_i as the PID for the party P_i. We abbreviate "probabilistic polynomial time" as PPT and "interactive Turing machine" as ITM.

3.1.1. Elliptic Curve Cryptography Notation. In this work, we work on the finite field $\mathbb{F}_p$, and the elliptic curve E is defined by two elements $a, b \in \mathbb{F}_p$. The set of all the points on E is denoted as $E_p(a, b)$. $g \coloneqq (x_g, y_g)$ is the base point of E with order n, and $h \coloneqq |E_p(a, b)|/n$ is the cofactor.

3.2. Security Definition. Our security model is based on the universal composability (UC) framework [34], which lays down a solid foundation for designing and analyzing protocol secure against attacks in an arbitrary network execution environment (therefore, it is also known as a network-aware security model). We refer to the original work [34] for details.

Roughly speaking, in the UC framework, protocols are carried out over multiple interconnected machines; to capture attacks, a network adversary $\mathscr{A}$ is introduced, which is allowed to partially control the communication network and corrupt some machines (i.e., have full control of all physical parts of some machines). Then, a protocol Π is a UC-secure implementation of a functionality $\mathscr{F}$, if it satisfies that for every network adversary $\mathscr{A}$ attacking an execution of Π, there is another adversary $\mathscr{S}$—known as the simulator—-attacking the ideal process that uses $\mathscr{F}$ (by corrupting the same set of machines) such that the executions of Π with $\mathscr{A}$ and that of $\mathscr{F}$ with $\mathscr{S}$ make no difference to any network execution environment $\mathscr{Z}$.

3.2.1. The Ideal World. In the ideal world, $P_1, \cdots, P_N$ only communicate with an ideal functionality $\mathscr{F}^f_{\mathrm{mpc}}$ during the execution. As depicted in Figure 1, $\mathscr{F}^f_{\mathrm{mpc}}$ waits for each party to provide input, and when all parties' inputs have been received, it computes the function $(y_1, \cdots, y_N) \longleftarrow f(x_1, \cdots, x_N)$ and sends the output y_i to the party P_i, for $i \in [N]$. Besides, the functionality $\mathscr{F}^f_{\mathrm{mpc}}$ interacts with the simulator $\mathscr{S}$. When a party P_i sends its input x_i to $\mathscr{F}^f_{\mathrm{mpc}}$, $\mathscr{S}$ receives a notification $(\mathrm{ComputeNotify}, \mathrm{sid}, P_i)$. Before $\mathscr{F}^f_{\mathrm{mpc}}$ outputs, it sends $(\mathrm{Output}, \mathrm{sid})$ to ask for permission of $\mathscr{S}$, and it only sends y_i to P_i if a $(\mathrm{Deliver}, \mathrm{sid}, P_i)$ is received.

3.2.2. Adversary Models. There are two main adversary models. A semihonest adversary follows the protocol description and a malicious adversary can deviate from the protocol description arbitrarily. Both adversaries try to help the environment distinguish between the ideal world and the real world by learning more information from the protocol execution.

3.2.3. Model of Protocol Execution. In the protocol execution, an environment $\mathscr{Z}$ provides inputs to the parties and receives outputs from them. Moreover, it can interact with the adversary $\mathscr{A}$ freely. At the end of the protocol, $\mathscr{Z}$ outputs a binary variable. Let $\mathrm{exec}_{\Pi,\mathscr{A},\mathscr{Z}}(i)$ denote the output variable of $\mathscr{Z}$ in an execution of protocol Π with environment $\mathscr{Z}$ and adversary $\mathscr{A}$ on input i, and $\mathrm{exec}_{\Pi,\mathscr{A},\mathscr{Z}}$ denote the ensemble $\{\mathrm{exec}_{\Pi,\mathscr{A},\mathscr{Z}}(i)\}_{i\in\{0,1\}^*}$. We use $\mathrm{exec}^{\mathscr{F}}_{\Pi,\mathscr{A},\mathscr{Z}}$ when protocol Π is in the $\mathscr{F}$-hybrid model, i.e., $\mathscr{F}$ can be invoked in Π. We slightly abuse notation and use $\mathrm{exec}_{\mathscr{F},\mathscr{S},\mathscr{Z}}$ for the ideal execution.

3.2.4. Random Oracle. A random oracle [35] is an idealized hash function that can be publicly accessed. In the random oracle model, the random oracle maintains a table of the previous queries. For a query with input x, the random oracle first checks if x is recorded. For an unrecorded x, the random oracle chooses an element from its output domain uniformly at random and responds with this element, and it then records x and the corresponding response; for a recorded x, the random oracle simply responds with the recorded response.

3.2.5. Public Key Infrastructure. A public key infrastructure (PKI) links a party's public identity with its public key. In this work, we use a PKI to guarantee the authenticity and validity of a party's public key and further ensure the security of the communication. In such a PKI setting, the parties can communicate with each other without the requirement of an underlying secure channel functionality [36]. In implementation, we allow the parties to send messages in an insecure network environment that may be eavesdropped on or tampered with without the TLS protocol.

3.3. One-Round Key Agreement Protocol. Key agreement (KA) protocols allow two parties A and B to jointly establish a key known to no one else. We use a similar notation as in [14], which considers two-round key agreement protocols, while we focus on one-round key agreement protocols. We first provide an illustrative example that is provided in Figure 2, and the protocol involves the following parameters:

(i) $\mathrm{KA}.\mathscr{R}$ is the set of randomness used by the parties

(ii) $\mathrm{KA.Msg}_1$ and $\mathrm{KA.Msg}_2$ are A's message generation function and B's message generation function, respectively

(iii) $\mathrm{KA}.\mathscr{M}_1$ and $\mathrm{KA}.\mathscr{M}_2$ are the set of A's protocol message and the set of B's protocol message, respectively

(iv) $\mathrm{KA.Key}_1$ and $\mathrm{KA.Key}_2$ are A's key generation function and B's key generation function, respectively

(v) $\mathrm{KA}.\mathscr{K}$ is the set of output keys

Functionality $\mathcal{F}^f_{\mathsf{mpc}}$

It interacts with players $\mathcal{P} := \{P_1, \ldots, P_N\}$ and the adversary $\mathcal{S}$. Let $\mathcal{P}_c$ be the set of corrupted parties. Initially, set $\mathcal{P}_c = \emptyset$.

Compute:

(i) Upon receiving $(\textsc{Compute}, \mathsf{sid}, x_i)$ from party $P_i \in \mathcal{P}$:
  (i) Send a notification $(\textsc{ComputeNotify}, \mathsf{sid}, P_i)$ to $\mathcal{S}$;
(ii) If it has received x_i from P_i for $i \in [N]$:
  (i) Compute $(y_1, \ldots, y_N) \leftarrow f(x_1, \ldots, x_N)$;
  (ii) Send $(\textsc{Output}, \mathsf{sid})$ to $\mathcal{S}$;
  (iii) For $i \in [N]$, upon receiving $(\textsc{Deliver}, \mathsf{sid}, P_i)$ from $\mathcal{S}$ and send $(\textsc{Compute}, \mathsf{sid}, y_i)$ to P_i;

Corruption handling:

(i) Upon receiving $(\textsc{Corrupt}, \mathsf{sid}, P_i)$ from the adversary $\mathcal{S}$, if $P_i \in \mathcal{P}$:
  (i) Set $\mathcal{P}_c := \mathcal{P}_c \cup \{P_i\}$;

Figure 1: Secure multiparty computation functionality $\mathscr{F}^f_{\text{mpc}}$.

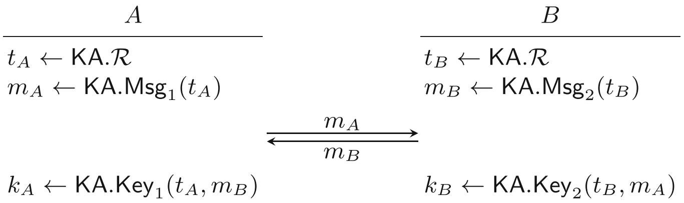

A
$t_A \leftarrow \mathsf{KA}.\mathcal{R}$
$m_A \leftarrow \mathsf{KA.Msg}_1(t_A)$
B
$t_B \leftarrow \mathsf{KA}.\mathcal{R}$
$m_B \leftarrow \mathsf{KA.Msg}_2(t_B)$
m_A
m_B
$k_A \leftarrow \mathsf{KA.Key}_1(t_A, m_B)$
$k_B \leftarrow \mathsf{KA.Key}_2(t_B, m_A)$

Figure 2: An illustrative example of one-round key agreement protocol.

In a one-round key agreement protocol Π, party A picks random $t_A \longleftarrow \mathrm{KA}.\mathscr{R}$ and computes $m_A \longleftarrow \mathrm{KA.Msg}_1(t_A)$, and it sends m_A to B; in the meanwhile, party B also picks random $t_B \longleftarrow \mathrm{KA}.\mathscr{R}$ and computes $m_B \longleftarrow \mathrm{KA.Msg}_2(t_B)$, and it sends m_B to A. At the end, A and B establish the same key by computing $k_A \longleftarrow \mathrm{KA.Key}_1(t_A, m_B)$ and $k_B \longleftarrow \mathrm{KA.Key}_2(t_B, m_A)$, respectively. Throughout the protocol, the computational security parameter λ is used implicitly as a parameter of the algorithms.

We require the protocol to have the following properties:

Definition 1 (Correctness). A one-round KA protocol Π is correct if for any $t_A, t_B \longleftarrow \mathrm{KA}.\mathscr{R}$, and $m_A \longleftarrow \mathrm{KA.Msg}_1(t_A)$, $m_B \longleftarrow \mathrm{KA.Msg}_2(t_B)$,

$$\Pr[k_A = \mathrm{KA.Key}_1(t_A, m_B) = \mathrm{KA.Key}_2(t_B, m_A) = k_B] = 1 - \mathrm{negl}(\lambda). \tag{1}$$

Definition 2 (Security). A one-round KA protocol Π is secure if for any PPT distinguisher $\mathscr{D}$,

$$\left| \Pr \left[\begin{array}{c} t_A, t_B \longleftarrow \mathrm{KA}.\mathscr{R}\,; \\ m_A \longleftarrow \mathrm{KA.Msg}_1(t_A)\,; m_B \longleftarrow \mathrm{KA.Msg}_2(t_B)\,; \\ k_1 \longleftarrow \mathrm{KA.Key}_1(t_A, m_B)\,; k_2 \longleftarrow \mathrm{KA}.\mathscr{K}\,; \\ b \longleftarrow \{0,1\}\,; b^* \longleftarrow \mathscr{D}(k_b, m_A, m_B)\colon b = b^* \end{array} \right] - \frac{1}{2} \right| = \mathrm{negl}(\lambda). \tag{2}$$

Definition 3 (Uniformity). A one-round KA protocol Π is Msg_1-uniform if for any PPT distinguisher $\mathscr{D}$,

$$\left| \Pr \left[\begin{array}{c} t_A \longleftarrow \mathrm{KA}.\mathscr{R}\,; m_1 \longleftarrow \mathrm{KA.Msg}_1(t_A)\,; m_2 \longleftarrow \mathrm{KA}.\mathscr{M}_1\,; \\ b \longleftarrow \{0,1\}\,; b^* \longleftarrow \mathscr{D}(m_b)\colon b = b^* \end{array} \right] - \frac{1}{2} \right| = \mathrm{negl}(\lambda). \tag{3}$$

Likewise, we can define Msg_2 uniformity.

Definition 4 (Robustness). A one-round KA protocol Π is robust if for any PPT distinguisher $\mathscr{D}$,

$$\left| \Pr \left[\begin{array}{c} t_A, t_B \longleftarrow \mathrm{KA}.\mathscr{R}\,; m_A \longleftarrow \mathrm{KA.Msg}_1(t_A)\,; m_B \longleftarrow \mathrm{KA.Msg}_2(t_B)\,; \\ (m_B^*, \mathrm{state}) \longleftarrow \mathscr{D}(m_A)\,; k^* \longleftarrow \mathrm{KA.Key}_1(t_A, m_B^*)\,; \\ k_1 \longleftarrow \mathrm{KA.Key}_2(t_A, m_B)\,; k_2 \longleftarrow \mathrm{KA}.\mathscr{K}\,; \\ b \longleftarrow \{0,1\}\,; b^* \longleftarrow \mathscr{D}(\mathrm{state}, k_b, k^*, m_B)\colon b = b^* \end{array} \right] - \frac{1}{2} \right| = \mathrm{negl}(\lambda). \tag{4}$$

3.4. Programmable-Once Public Function. The primitive programmable-once public function (POPF) is proposed by McQuoid et al. [14]. Later in [15], they fixed some issues in the definition and formally defined a batch 2-POPF. In our endemic OT protocol, we use a N-POPF, and in its multi-instance variant, we use a batch N-POPF; therefore, we provide a formal definition of a batch N-POPF here.

A batch N-POPF consists of two algorithms: Program $: [N] \times \mathcal{N} \longrightarrow \mathcal{M}$ and Eval $: \mathcal{M} \times [N] \longrightarrow \mathcal{N}$. Programmable-once means that one can compute $\phi = \text{Program}(x, y)$ for an $x \in [N]$ and $y \in \mathcal{N}$, but for any other $x' \neq x$, the value of $y' \in \mathcal{N}$ such that $\text{Program}(x', y') = \phi$ should be unpredictable, i.e., y' looks like random. This unpredictability is defined with respect to a 1-weak random oracle $F : \mathcal{N} \longrightarrow \mathcal{O}$ which produces a pseudorandom $y \coloneqq F(x)$ when F is only allowed to be accessed once.

Definition 5 (1-weak random oracle). A function $F : \mathcal{N} \longrightarrow \mathcal{O}$ is a 1-weak random oracle if for any PPT distinguisher $\mathcal{D}$,

$$\left|\Pr\left[x \longleftarrow \mathcal{N} ; y_0 \coloneqq F(x) ; y_1 \longleftarrow \mathcal{O} ; b \longleftarrow \{0,1\} ; b^* \longleftarrow \mathcal{D}(x, y_b) \colon b = b^*\right] - \frac{1}{2}\right| = \text{negl}(\lambda). \tag{5}$$

$\mathcal{D}$ can only access F through this experiment.

Now we can formally define a batch N-POPF. Generally, a batch N-POPF makes use of some local setups $\mathcal{H}$, which can consist of random oracles, common reference strings, etc. We use $\text{Program}^{\mathcal{H}}$ and $\text{Eval}^{\mathcal{H}}$ to denote the algorithms when they access $\mathcal{H}$. Besides, a batch N-POPF should include two alternative local setups:

$\mathcal{H}_{\text{Sim}}$: this setup provides the same interface as $\mathcal{H}$ and an additional method Sim $: \mathcal{N}^N \longrightarrow \mathcal{M}$.

$\mathcal{H}_{\text{Extract}}$: this setup provides the same interface as $\mathcal{H}$ and an additional method Extract $: \mathcal{M} \longrightarrow [N]$.

We require the batch N-POPF to have the following properties:

Definition 6 (Correctness). A batch N-POPF is correct if for any $x \in [N], y \in \mathcal{N}$,

$$\Pr[\text{Eval}(\text{Program}(x, y), x) = y] = 1 - \text{negl}(\lambda). \tag{6}$$

Definition 7 (Honest simulation). A batch N-POPF has honest simulation if for any PPT distinguisher $\mathcal{D}$ and PPT adversary $\mathcal{A}$,

$$\left|\Pr\begin{bmatrix} x \longleftarrow [N] ; (s, y) \longleftarrow \mathcal{A}(\quad) ; \phi^0 \longleftarrow \text{Program}(x, y), \text{for} i \in [N], r_i^0 \coloneqq \text{Eval}\left(\phi^0, i\right) ; \\ r_x^1 \coloneqq y, \text{for} i \neq x, r_i^1 \longleftarrow \mathcal{N}, \phi^1 \longleftarrow \text{Sim}(r_1, \cdots, r_N) ; \\ b \longleftarrow \{0,1\} ; b^* \longleftarrow \mathcal{D}^{\mathcal{H}_{\text{Sim}}}\left(s, \phi^b, \left\{r_i^b\right\}_{i \in [N]}\right) \colon b = b^* \end{bmatrix} - \frac{1}{2}\right| = \text{negl}(\lambda). \tag{7}$$

The honest simulation property captures the batch N-POPF's ability of hiding x: when $b = 0$, ϕ^0 is generated using x and when $b = 1$, ϕ^1 is generated from random $\{r_i^1\}_{i \in [N]}$. If ϕ^0 and ϕ^1 are indistinguishable even when $\{r_i^1\}_{i \in [N]}$ is given, we can say that ϕ does not leak the information of x.

Definition 8 (Uncontrollable outputs). A batch N-POPF has uncontrollable outputs if for any 1-weak random oracle F any PPT distinguisher $\mathcal{D}$ and PPT adversary $\mathcal{A}$,

$$\left|\Pr\begin{bmatrix} (\phi, \text{state}) \longleftarrow \mathcal{A}^{\mathcal{H}_{\text{Extract}}} ; \\ x \coloneqq \text{Extract}(\phi), \text{for} i \neq x, r_i^0 \coloneqq F(\text{Eval}(\phi, i)) ; \\ \text{for} i \neq x, r_i^1 \longleftarrow \mathcal{N} ; \\ b \longleftarrow \{0,1\} ; b^* \longleftarrow \mathcal{D}\left(\text{state}, \left\{r_i^b\right\}_{i \neq x}\right) \colon b = b^* \end{bmatrix} - \frac{1}{2}\right| = \text{negl}(\lambda). \tag{8}$$

The uncontrollable output property restricts the adversary to only be able to program once. Given any ϕ produced by the adversary $\mathcal{A}$, the Extract method finds an x such that for $i \neq x$, the value of $\text{Eval}(\phi, i)$ is unpredictable.

Moreover, when the batch N-POPF has an honest simulation and uncontrollable outputs, and the interface of $\mathcal{H}_{\text{Sim}}$ and $\mathcal{H}_{\text{Extract}}$ looks indistinguishable from the adversary, we can say that the batch N-POPF is secure.

In this work, we use a correct and secure batch N-POPF which is constructed by McQuoid et al. in [15]. For simplicity, we denote $E_p(a, b)$ as $\mathbb{G}$. The hash functions $\{\text{hash}_i^{\mathbb{G}}\}_{i \in [N]}$ are defined as $\text{hash}_i^{\mathbb{G}} : \mathbb{G}^{N-1} \longrightarrow \mathbb{G}$ and are modeled as random oracles. We provide the details of the batch N-POPF in Figure 3, and we have the following theorem from [15]:

Theorem 9 (See [15]). *Figure 3 defines a correct and secure batch N-POPF.*

3.5. Oblivious Transfer. Oblivious transfer (OT) is a cryptographic primitive that allows a receiver Rec to choose and to obtain several messages from a bunch of messages held by a sender Sen, while Sen is not aware of Rec's choice, and Rec will not learn anything about the unchosen messages. The messages held by Sen can be contributed by itself or generated by the OT functionality. We denote an OT functionality where Sen decides the messages as sender OT $\mathcal{F}_{S-\text{OT}}$ and an OT functionality which generates messages for Sen as random OT $\mathcal{F}_{U-\text{OT}}$. In [10], Naor and Pinkas first provide an efficient implementation of $\mathcal{F}_{S-\text{OT}}$, and in [11], Chou and Orlandi propose the simplest OT protocol for

```
$\mathcal{N} := \mathbb{G}, \mathcal{M} := \mathbb{G}^N$

$\mathcal{H}$
    Record the history calls in the transcript $\mathcal{T}$
$\mathsf{hash}_i^{\mathbb{G}}(u)$ :
    If there exists a $v$ s.t. $(v = \mathsf{hash}_i^{\mathbb{G}}(u)) \in \mathcal{T}$:
        Output $v$
    Else:
        Output random $v \leftarrow \mathbb{G}$

$\mathsf{Program}(x, y)$ :
    For $i \neq x$, pick random $r_i \leftarrow \mathbb{G}$
    Compute $r_x := y - \mathsf{hash}_x^{\mathbb{G}}(\{r_i\}_{i \neq x})$
    Output $\phi := \{r_i\}_{i \in [N]}$

$\mathsf{Eval}(\phi = \{r_i\}_{i \in [N]}, x)$ :
    Output $r_x + \mathsf{hash}_x^{\mathbb{G}}(\{r_i\}_{i \neq x})$
```

```
$\mathcal{H}_{\textsc{Sim}}$
    Record the history calls in the transcript $\mathcal{T}$
    Allocate an empty array $U$
$\mathsf{hash}_i^{\mathbb{G}}(u)$ :
    If there exists a $v$ s.t. $(v = \mathsf{hash}_i^{\mathbb{G}}(u)) \in \mathcal{T}$:
        Output $v$
    Else if $U[i, u]$ has been defined:
        Output $U[i, u]$
    Else:
        Output random $v \leftarrow \mathbb{G}$
$\textsc{Sim}(r_1, \ldots, r_N)$ :
    Pick random $\phi = (s_1, \ldots, s_N) \leftarrow \mathcal{M}$
    For $i \in [N]$, set $U[i, \{s_j\}_{j \neq i}] := r_i + s_i$
    Output $\phi$

$\mathcal{H}_{\textsc{Extract}}$
    Record the history calls in the transcript $\mathcal{T}$
    The interfaces of $\{\mathsf{hash}_i^{\mathbb{G}}\}_{i \in [N]}$ are the same as $\mathcal{H}$
$\textsc{Extract}(\phi = \{r_i\}_{i \in [N]})$
    Find the first query in $\mathcal{T}$ s.t. $\mathsf{hash}_x^{\mathbb{G}}(\{r_i\}_{i \neq x})$ is in $\mathcal{T}$
        Output $x$
    If there are no such queries:
        Output 1
```

Figure 3: Batch N-way programmable-once public function from [15].

1-out-of-N Endemic Oblivious Transfer Functionality $\mathcal{F}_{\text{E-OT}}^{1,N}$

It interacts with players $\mathcal{P} := \{\mathsf{Sen}, \mathsf{Rec}\}$ and the adversary $\mathcal{S}$. It is parameterized by the length of the messages length. Let $\tilde{\mathcal{P}}$ be the set of corrupted parties. Initially, set $\tilde{\mathcal{P}} = \emptyset$.

Transfer:

(i) Upon receiving $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ from Sen:
- (i) Send a notification $(\textsc{SendNotify}, \mathsf{sid}, \mathsf{ssid})$ to $\mathcal{S}$;
- (ii) Store $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$;
- (iii) Ignore future $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ messages with the same $\mathsf{sid}, \mathsf{ssid}$;

(ii) Upon receiving $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, c)$ from Rec, where c in $[N]$:
- (i) Send a notification $(\textsc{ReceiveNotify}, \mathsf{sid}, \mathsf{ssid})$ to $\mathcal{S}$;
- (ii) Store $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, c)$;
- (iii) Ignore future $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, \ldots)$ messages with the same $\mathsf{sid}, \mathsf{ssid}$;

(iii) If both $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ and $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, c)$ are stored:
- (i) For $i \in [N]$, pick random $m_i \leftarrow \{0,1\}^{\mathsf{length}}$;
- (ii) If $\mathsf{Sen} \in \tilde{\mathcal{P}}$, wait for $(\textsc{FixMessage}, \mathsf{sid}, \mathsf{ssid}, \{\tilde{m}_i\}_{i \in [N]})$ from $\mathcal{S}$, set $m_i := \tilde{m}_i$, for $i \in [N]$;
- (iii) If $\mathsf{Rec} \in \tilde{\mathcal{P}}$, wait for $(\textsc{FixMessage}, \mathsf{sid}, \mathsf{ssid}, \tilde{m}_c)$ from $\mathcal{S}$, set $m_c := \tilde{m}_c$;
- (iv) Send $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid}, \{m_i\}_{i \in [N]})$ to Sen and $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, m_c)$ to Rec;

Corruption handling:

(i) Upon receiving $(\textsc{Corrupt}, \mathsf{sid}, \mathsf{ssid}, P)$ from the adversary $\mathcal{S}$, if $P \in \mathcal{P}$:
- (i) Set $\tilde{\mathcal{P}} := \tilde{\mathcal{P}} \cup \{P\}$;
- (ii) Send $(\textsc{Input}, \mathsf{sid}, \mathsf{ssid}, P, x)$ to $\mathcal{S}$ if party P's input x is already defined;

Figure 4: 1-out-of-N endemic oblivious transfer functionality $\mathscr{F}_{E\text{-}\mathrm{OT}}^{1,N}$.

$\mathscr{F}_{U\text{-}\mathrm{OT}}$, and they show how to transform it into a $\mathscr{F}_{S\text{-}\mathrm{OT}}$ protocol. However, both [10] and the random OT protocol in [11] do not provide full simulation-based security.

In this work, we make extensive use of a special notion of OT called 1-out-of-N endemic OT, which is proposed in [12]. Endemic OT is essentially the same as random OT, except that the endemic OT functionality allows the adversary $\mathscr{S}$ to determine the corrupted party's messages. As depicted in Figure 4, the 1-out-of-N endemic OT functionality $\mathscr{F}_{E\text{-}\mathrm{OT}}^{1,N}$ waits for (Send, sid, ssid) from Sen

Private Set Intersection Functionality $\mathcal{F}_{\text{PSI}}$

It interacts with players $\mathcal{P} := \{\text{Sen}, \text{Rec}\}$ and the adversary $\mathcal{S}$. It is parameterized by the set size of Sen and Rec, n_1 and n_2. Let $\tilde{\mathcal{P}}$ be the set of corrupted parties. Initially, set $\tilde{\mathcal{P}} = \emptyset$.

Compute:

(i) Upon receiving $(\textsc{Compute}, \text{sid}, \text{Sen}, X = (x_1, \ldots, x_{n_1}))$ from Sen:
  (i) Send a notification $(\textsc{ComputeNotify}, \text{sid}, \text{Sen})$ to $\mathcal{S}$;
  (ii) Store $(\textsc{Compute}, \text{sid}, \text{Sen}, X)$;
  (iii) Ignore future $(\textsc{Compute}, \text{sid}, \text{Sen}, \ldots)$ messages with the same sid;
(ii) Upon receiving $(\textsc{Compute}, \text{sid}, \text{Rec}, Y = (y_1, \ldots, y_{n_2}))$ from Rec:
  (i) Send a notification $(\textsc{ComputeNotify}, \text{sid}, \text{Rec})$ to $\mathcal{S}$;
  (ii) Store $(\textsc{Compute}, \text{sid}, \text{Rec}, Y)$;
  (iii) Ignore future $(\textsc{Compute}, \text{sid}, \text{Rec}, \ldots)$ messages with the same sid;
(iii) If both $(\textsc{Compute}, \text{sid}, \text{Sen}, X)$ and $(\textsc{Compute}, \text{sid}, \text{Rec}, Y)$ are stored:
  (i) Compute $\text{Res} := X \cap Y$;
  (ii) Send $(\textsc{Compute}, \text{sid})$ to Sen and $(\textsc{Compute}, \text{sid}, \text{Res})$ to Rec;

Corruption handling:

(i) Upon receiving $(\textsc{Corrupt}, \text{sid}, P)$ from the adversary $\mathcal{S}$, if $P \in \mathcal{P}$:
  (i) Set $\tilde{\mathcal{P}} := \tilde{\mathcal{P}} \cup \{P\}$;
  (ii) Send $(\textsc{Input}, \text{sid}, P, x)$ to $\mathcal{S}$ if party P's input x is already defined;

Figure 5: Private set intersection functionality $\mathcal{F}_{\text{PSI}}$.

and (Receive, sid, ssid, c) from Rec, where $c \in [N]$ denotes Rec's choices. After both messages are obtained, $\mathcal{F}_{E\text{-OT}}^{1,N}$ picks n uniformly random messages $\{m_i\}_{i\in[N]}$. However, when the adversary $\mathcal{S}$ corrupts Sen, it is allowed to determine all the messages by sending (FixMessage, sid, ssid, $\{\tilde{m}_i\}_{i\in[N]}$), and when the adversary $\mathcal{S}$ corrupts Rec, it is allowed to determine the message m_c by sending (FixMessage, sid, ssid, m_c). At the end, $\mathcal{F}_{E\text{-OT}}^{1,N}$ sends $\{m_i\}_{i\in[N]}$ to Sen and m_c to Rec. We also provide the standard random OT functionality $\mathcal{F}_{U\text{-OT}}^{1,N}$ in Appendix A.1.

3.6. Private Set Intersection. Private set intersection (PSI) is a specialized MPC problem. In PSI, two parties want to compute the intersection of their input sets, without revealing the content of their inputs. As described in Figure 5, the party Sen and Rec send their input sets X and Y to the functionality $\mathcal{F}_{\text{PSI}}$, and $\mathcal{F}_{\text{PSI}}$ computes the intersection $\text{Res} \coloneqq X \cap Y$ and sends Res to Rec. PSI can be solved using generic MPC techniques, like GMW protocol [37] and Yao's garbled circuit protocol [38], while there are also custom protocols for this problem that are more efficient.

4. SM Series Cryptography

4.1. SM3 Hash Function and Key Derivation Function. Let SM3$(\cdot)$ denote the SM3 hash function that can map an arbitrary length string to a ℓ-bit hash digest, i.e., $\text{SM3} : \{0,1\}^* \longrightarrow \{0,1\}^\ell$. Let KDF($m$, length) denote the key derivation function that takes input as the string $m \in \{0,1\}^*$ and the key length length $\in \mathbb{N}$, and it outputs length-bit key string k. In this work, we implement KDF with SM3 hash function SM3, and the details are shown in Algorithm 1.

The security of KDF directly follows the security of SM3.

4.2. SM4 Block Cipher Algorithm. Let $F : \{0,1\}^\lambda \times \{0,1\}^\ell \longrightarrow \{0,1\}^\ell$ denote the block cipher that takes a λ-bit seed and ℓ-bit plaintext as input and output an ℓ-bit ciphertext. The SM4 block cipher algorithm has an electronic codebook (ECB) mode and a cipher block chaining (CBC) mode. In ECB mode, SM4 is instantiated by repeatedly invoking F, in other words, $\text{SM4}_k(m_1, \cdots, m_n) = (F_k(m_1), \cdots, F_k(m_n))$; in CBC mode, each block of ciphertext depends on the previous ciphertext block and an initialization vector iv is used, more specifically, $\text{SM4}_k(m_1, \cdots, m_n) = (c_1, \cdots, c_n)$, where $c_1 \coloneqq F_k(m_1 \oplus \text{iv})$ and $c_i \coloneqq F_k(m_i \oplus c_{i-1})$ for $i \neq 1$.

4.3. SM2 Key Agreement Protocol. The SM2 key agreement protocol is defined in the PKI setting, and it works on elliptic curve E defined over the field $\mathbb{F}_p$. The parties A and B first agree on the output key length and an elliptic curve system where the elliptic curve discrete logarithm problem (ECDLP) is hard. The system parameters include p, a, b, $g = (x_g, y_g), n, h$, and $w \coloneqq \lceil(\lceil\log_2(n)\rceil/2)\rceil - 1$ is used in the computation.

We assume each party knows the other party's distinguishing identifier ID and a PKI distributes the public key $\text{pk} = (x_{\text{pk}}, y_{\text{pk}})$ computed from $\text{pk} = [\text{sk}]g$. Therefore, the parties can compute the identifier hash value $Z \coloneqq \text{SM3}(\text{ENTL}\|\text{ID}\|a\|b\|x_g\|y_g\|x_{\text{pk}}\|y_{\text{pk}})$, where ENTL is the length of ID and SM3 outputs a 256-bit string. Therefore, A has s $\text{k}_A, \text{pk}_A, \text{pk}_B, Z_A, Z_B$, and B has $\text{sk}_B, \text{pk}_A, \text{pk}_B, Z_A, Z_B$. The public key/secret key pairs are used to prevent the man-in-the-middle attack, and the identifier hash values are used to identify the parties executing the protocol. In a nutshell, the SM2 key agreement protocol consists of the following PPT algorithms which use elliptic curve system parameters implicitly:

```
1. Set ctr = 1;
2. For $i \in [\lceil \text{length}/\ell \rceil]$:
   (a) Compute $h_{\text{ctr}} \longleftarrow \text{SM3}(m \| ctr)$;
   (b) Set ctr = ctr + 1;
3. If $\lceil \text{length}/\ell \rceil \neq \text{length}/\ell$, set $\bar{h}$ as the top length $-$ ($\ell * \lfloor \text{length}/\ell \rfloor$) bits of $h_{\lceil \text{length}/\ell \rceil}$; otherwise, set $\bar{h} := h_{\lceil \text{length}/\ell \rceil}$;
4. Set $k := h_1 \| \cdots \| h_{\lceil \text{length}/\ell \rceil - 1} \| \bar{h}$;
5. Output $k$.
```

Algorithm 1

(i) $(m, t) \longleftarrow \text{MsgGen}()$ is the message generation algorithm that outputs a fresh random private message t and the corresponding public message m

(ii) $p \longleftarrow \text{PointGen}(t, m, m', \text{sk}, \text{pk}')$ is the point generation algorithm that takes input as the party's private message t, both parties' public messages m, m', the party's secret key sk, and the other party's public key pk$'$, and it outputs a point that can be used to derive the shared key

(iii) $k \longleftarrow \text{KeyGen}(p, Z_A, Z_B)$ is the key generation algorithm that takes input as the point p, A and B's identification message Z_A, Z_B, and it outputs a shared key k

We slightly modify the order of message delivery in the original SM2 key agreement protocol to achieve the one-round property. In the original protocol, party B sends the messages m_B to A after it obtains the key k_B, which indicates B sends m_B only after m_A is received, while we notice that this step can be done right after m_B is generated, and it does not raise any security issues. In our modified protocol, A first invokes MsgGen to generate $(m_A, t_A) \longleftarrow \text{MsgGen}()$, and it sends m_A to B. After receiving m_B from B, A computes the point $p_A := \text{PointGen}(t_A, m_A, m_B, \text{sk}_A, \text{pk}_B)$ and derives $k_A := \text{KeyGen}(p_A, Z_A, Z_B)$. The execution of B is exactly symmetric. The process of the protocol is illustrated in Figure 6, and the details of the algorithms can be found in Figure 7.

When key confirmation is needed, an augmented SM2 key agreement protocol can be used which contains one more PPT algorithm Verify and several more steps.

$h \longleftarrow \text{Verify}(s, p, m_A, m_B, Z_A, Z_B)$ is the verification algorithm that takes input as a string s, a point p, both parties' public messages m_A, m_B, and both parties' identifier hash values Z_A, Z_B, and it outputs a hash value h.

As in Figure 6, after generating p_A and k_A, A invokes $h_A := \text{Verify}(\text{``}A\text{''}, p_A, m_A, m_B, Z_A, Z_B)$ as a proof that it obtains a correct point p_A and is able to derive a correct key, and it sends h_A to B. When it receives h_B from B, it checks if B obtains the correct point p_B. The process of B is likewise.

Claim 10. If the key derivation function KDF is modeled as a random oracle, and the ECDLP is hard in $E_p(a, b)$, then the SM2 key agreement protocol is a one-round KA protocol with perfect correctness, security, perfect Msg_1 uniformity, perfect Msg_2 uniformity, and robustness.

Proof. It has been shown in [39] that the SM2 key agreement protocol is secure in the well-known Bellare-Rogaway model [40, 41] when the key derivation function is modeled as a random oracle and the ECDLP is hard in $E_p(a, b)$. Security in the Bellare-Rogaway model means perfect correctness and security of the KA protocol. Moreover, as illustrated in Figure 7, $m_A = [t_A]g$, where t_A is a uniformly random element from $[n-1]$; since m_A is generated by a bijective function, it should be perfectly indistinguishable from a random element in $\mathbb{G}$, which indicates the perfect Msg_1 uniformity of the protocol, and the perfect Msg_2 uniformity can be obtained in the same way. At the end, the parties A and B already hold $p_A = p_B$ after PointGen is invoked, which can be used as a shared key, and the KeyGen function only converts the point to a bit string. As proved in [14], this gives the robustness of the SM2 key agreement protocol when KDF is modeled as a random oracle, since a random oracle outputs uniformly random strings. □

5. Construct Oblivious Transfer Using SM

In this section, we show how to construct an oblivious transfer protocol using the SM series cryptography and the batch N-POPF. Moreover, we illustrate how to extend the number of OT instances using OT extension protocols. Before presenting the constructions, we first provide the descriptions of the symbols used in Table 1.

5.1. Oblivious Transfer from SM2 Key Agreement. Our one-round 1-out-of-N endemic oblivious transfer protocol $\Pi_{E\text{-OT}}^{1,N}$ is constructed from the SM2 key agreement protocol and the batch N-POPF defined in Figure 3. As depicted in Figure 8, the sender Sen and the receiver Rec first run a setup phase to determine the protocol parameters and to exchange the public keys along with the distinguishable identifiers. This setup phase only needs to be run once between these two parties Sen and Rec, and the parameters can be used in multiple instances.

When Sen receives the instruction $(\text{Send}, \text{sid}, \text{ssid})$ from the environment $\mathcal{Z}$, it invokes MsgGen to generate m_A, t_A, and it sends m_A to Rec. Meanwhile, Rec receives the instruction $(\text{Receive}, \text{sid}, \text{ssid}, c)$ from $\mathcal{Z}$, and it invokes $(m_B, t_B) \longleftarrow \text{MsgGen}()$. After that, Rec generates $\{r_i\}_{i\in[N]}$

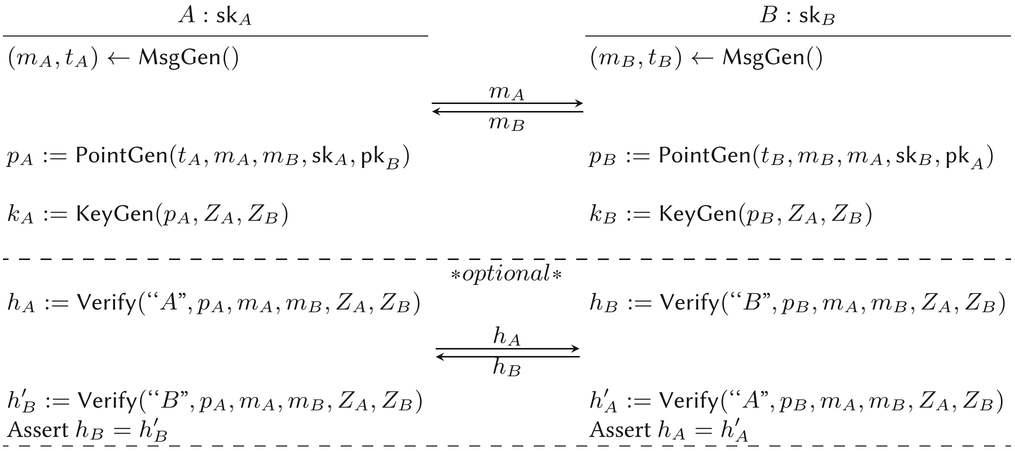

$A : \mathsf{sk}_A$
$B : \mathsf{sk}_B$
$(m_A, t_A) \leftarrow \mathsf{MsgGen}()$
$(m_B, t_B) \leftarrow \mathsf{MsgGen}()$
m_A
m_B
$p_A := \mathsf{PointGen}(t_A, m_A, m_B, \mathsf{sk}_A, \mathsf{pk}_B)$
$p_B := \mathsf{PointGen}(t_B, m_B, m_A, \mathsf{sk}_B, \mathsf{pk}_A)$
$k_A := \mathsf{KeyGen}(p_A, Z_A, Z_B)$
$k_B := \mathsf{KeyGen}(p_B, Z_A, Z_B)$
$*optional*$
$h_A := \mathsf{Verify}(``A", p_A, m_A, m_B, Z_A, Z_B)$
$h_B := \mathsf{Verify}(``B", p_B, m_A, m_B, Z_A, Z_B)$
h_A
h_B
$h'_B := \mathsf{Verify}(``B", p_A, m_A, m_B, Z_A, Z_B)$
Assert $h_B = h'_B$
$h'_A := \mathsf{Verify}(``A", p_B, m_A, m_B, Z_A, Z_B)$
Assert $h_A = h'_A$

FIGURE 6: SM2 key agreement protocol with optional key confirmation. The parties A and B share the elliptic curve system parameters $p, a, b, g, n, h, w := \lceil(\lceil\log_2(n)\rceil/2)\rceil - 1$; the public keys $\mathrm{pk}_A, \mathrm{pk}_B$; the identifier hash values Z_A, Z_B; and the output key length.

```
MsgGen() :
    Pick random $t \leftarrow [n-1]$
    Set $m := [t]g$
    Output $(m, t)$

PointGen($t, m, m'$, sk, pk') :
    $m = (x, y)$ and $m' = (x', y')$
    Set $\bar{x} := 2^w + (x \& (2^w - 1))$
    Set $\bar{x}' := 2^w + (x' \& (2^w - 1))$
    Set $r := (\mathsf{sk} + \bar{x} \cdot t) \mod n$
    Set $p := [h \cdot r](\mathsf{pk}' + [\bar{x}']m')$
    Output $p$

KeyGen($p, Z_A, Z_B$) :
    $p = (x^*, y^*)$
    Compute $k := \mathsf{KDF}(x^* || y^* || Z_A || Z_B, \mathsf{length})$
    Output $k$

Verify($s, p, m_A, m_B, Z_A, Z_B$)
    $p = (x^*, y^*)$
    $m_A = (x_A, y_A)$ and $m_B = (x_B, y_B)$
    Compute $r := \mathsf{SM3}(x^* || Z_A || Z_B || x_A || y_A || x_B || y_B)$
    Compute $h := \mathsf{SM3}(s || y^* || r)$
    Output $h$
```

FIGURE 7: Algorithms used in SM2 key agreement protocol.

TABLE 1: Symbols used in the OT protocol and the OT extension protocol.

Symbol	Description
sk_A, sk_B	The private key of the users
pk_A, pk_B	The public key of the users
Z_A, Z_B	The hash value of the user's ID, the system's parameter, and the user's public key
MsgGen()	The message generation algorithm in the SM2 key agreement protocol
PointGen()	The point generation algorithm in the SM2 key agreement protocol
KeyGen()	The key generation algorithm in the SM2 key agreement protocol
Program()	The program algorithm in the batch N-POPF
Eval()	The evaluation algorithm in the batch N-POPF

by $\{r_i\}_{i\in[N]} := \mathrm{Program}(c, m_B)$ and sends $\{r_i\}_{i\in[N]}$ to Sen. Upon receiving $\{r_i\}_{i\in[N]}$ from Rec, for $i \in [N]$, Sen sets $m_{B,i} := \mathrm{Eval}(\{r_j\}_{j\in[N]}, i)$, computes $p_i := \mathrm{PointGen}(t_A, m_A, m_{B,i}, \mathrm{sk}_A, \mathrm{pk}_B)$, and sets $k_i := \mathrm{KeyGen}(p_i, Z_A, Z_B)$. Sen then returns (Send, sid, ssid, $\{k_i\}_{i\in[N]}$) to $\mathcal{Z}$. Upon receiving m_A, Rec computes $p_c := \mathrm{PointGen}(t_B, m_B, m_A, \mathrm{sk}_B, \mathrm{pk}_A)$ and sets $k_c := \mathrm{KeyGen}(p_c, Z_A, Z_B)$. At the end, Rec returns (Receive, sid, ssid, k_c) to $\mathcal{Z}$.

The correctness of the protocol directly follows the correctness of the SM2 key agreement protocol and the batch N-POPF. For the security proof, intuitively, since the SM2 key agreement has perfect Msg_2 uniformity, the message m_B should be indistinguishable from a random element from $\mathbb{G}$; by the honest simulation property of the batch N-POPF, r_c should be indistinguishable from other $\{r_i\}_{i\neq c}$. Therefore, Rec's choice c remains private to Sen. Besides, because of the robustness of the SM2 key agreement

The One-Round 1-out-of-N Endemic Oblivious Transfer Protocol $\Pi^{1,N}_{\text{E-OT}}$

Setup: The sender Sen and the receiver Rec first agree on the SM2 key agreement protocol parameters with the assistance of the PKI. The parameters include the elliptic curve system parameters p, a, b, g, n, h, public keys $\mathsf{pk}_A, \mathsf{pk}_B$, and identifier hash values Z_A, Z_B, the key length is set as the OT message length.

```
(i) Upon receiving $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ from the environment $\mathcal{Z}$, Sen:
    (i) Compute $(m_A, t_A) \leftarrow \mathsf{MsgGen}()$;
    (ii) Send $m_A$ to Rec;
(ii) Upon receiving $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, c)$ from the environment $\mathcal{Z}$, Rec:
    (i) Compute $(m_B, t_B) \leftarrow \mathsf{MsgGen}()$;
    (ii) Compute $\{r_i\}_{i\in[N]} := \mathsf{Program}(c, m_B)$;
    (iii) Send $\{r_i\}_{i\in[N]}$ to Sen;
(iii) Upon receiving $\{r_i\}_{i\in[N]}$ from Rec, Sen:
    (i) For $i \in [N]$:
        (i) Compute $m_{B,i} := \mathsf{Eval}(\{r_j\}_{j\in[N]}, i)$;
        (ii) Compute $p_i := \mathsf{PointGen}(t_A, m_A, m_{B,i}, \mathsf{sk}_A, \mathsf{pk}_B)$;
        (iii) Compute $k_i := \mathsf{KeyGen}(p_i, Z_A, Z_B)$;
    (ii) Return $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid}, \{k_i\}_{i\in[N]})$ to the environment $\mathcal{Z}$;
(iv) Upon receiving $m_A$ from Sen, Rec:
    (i) Compute $p_c := \mathsf{PointGen}(t_B, m_B, m_A, \mathsf{sk}_B, \mathsf{pk}_A)$;
    (ii) Compute $k_c := \mathsf{KeyGen}(p_c, Z_A, Z_B)$;
    (iii) Return $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, k_c)$ to the environment $\mathcal{Z}$;
```

Figure 8: One-round 1-out-of-N endemic oblivious transfer protocol $\Pi^{1,N}_{E\text{–OT}}$.

protocol and the uncontrollable output property of the batch N-POPF, the output messages $\{k_i\}_{i\neq c}$ should be unpredictable to Rec.

More formally, we use a theorem from [15]:

Theorem 11 (See [15]). *If the KA protocol has security, Msg_2 uniformity and robustness and the batch N-POPF are secure, and then, the protocol $\Pi^{1,N}_{E\text{–}OT}$ described in Figure 8 securely realizes the endemic 1-OPRF functionality in the random oracle model.*

Specifically, 1-OPRF is essentially 1-out-of-N OT. Therefore, we have the following result:

Theorem 12. *If ECDLP is hard in $\mathbb{G}$, the hash functions $\{hash_i^{\mathbb{G}}\}_{i\in[N]}$ and the key derivation function KDF are modeled as random oracles, and then, the protocol $\Pi^{1,N}_{E\text{–}OT}$ described in Figure 8 securely realizes $\mathcal{F}^{1,N}_{E\text{–}OT}$ described in Figure 4 against any PPT malicious adversary corrupting Sen or/and Rec.*

The proof is automatically done given Theorems 9 and 11.

The main focus of [12, 14, 15] is the malicious setting. When it comes to the semihonest setting, we notice that $\Pi^{1,N}_{E\text{–OT}}$ securely realizes the standard random OT functionality $\mathcal{F}^{1,N}_{U\text{–OT}}$ as well as the endemic OT functionality $\mathcal{F}^{1,N}_{E\text{–OT}}$. This is because the power of the adversary is limited to observing the protocol messages. We provide the results below.

Theorem 13. *If ECDLP is hard in $\mathbb{G}$, the hash functions $\{hash_i^{\mathbb{G}}\}_{i\in[N]}$ and the key derivation function KDF are modeled as random oracles, and then, the protocol $\Pi^{1,N}_{E\text{–}OT}$ described in Figure 8 securely realizes $\mathcal{F}^{1,N}_{U\text{–}OT}$ described in Figure 9 against any PPT semihonest adversary corrupting Sen or/and Rec.*

The proof can be found in Appendix B.1.

Corollary 14. *If ECDLP is hard in $\mathbb{G}$, the hash functions $\{hash_i^{\mathbb{G}}\}_{i\in[N]}$ and the key derivation function KDF are modeled as random oracles, and then, the protocol $\Pi^{1,N}_{E\text{–}OT}$ described in Figure 8 securely realizes $\mathcal{F}^{1,N}_{E\text{–}OT}$ described in Figure 4 against any PPT semihonest adversary corrupting Sen or/and Rec.*

The proof can be found in Appendix B.2.

5.2. Oblivious Transfer Extension. Although the endemic OT protocol $\Pi^{1,N}_{E\text{–OT}}$ is quite efficient, the exponentiation operations in $\Pi^{1,N}_{E\text{–OT}}$ can still be too expensive when we need millions of OTs in applications. In such cases, a technique called oblivious transfer extension can be adopted to generate OTs much faster. An OT extension protocol takes a bunch of "base" OTs to initiate the protocol, and then, it extends them to polynomially many OTs using only symmetric primitives, instead of asymmetric primitives. The Beaver [42] first introduced the idea of OT extension, and the following works [43, 44] proposed several highly efficient OT extension protocols. In this work, we use the well-optimized 1-out-of-2 OT extension protocol from [45] in the semihonest setting; in the malicious setting, we consider the protocol from [46]

1-out-of-N Random Oblivious Transfer Functionality $\mathcal{F}^{1,N}_{\text{U-OT}}$

It interacts with players $\mathcal{P} := \{\mathsf{Sen}, \mathsf{Rec}\}$ and the adversary $\mathcal{S}$. It is parameterized by the length of the messages length. Let $\tilde{\mathcal{P}}$ be the set of corrupted parties. Initially, set $\tilde{\mathcal{P}} = \emptyset$.

Transfer:

- (i) Upon receiving $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ from Sen:
  - (i) Send a notification $(\textsc{SendNotify}, \mathsf{sid}, \mathsf{ssid})$ to $\mathcal{S}$;
  - (ii) Store $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$;
  - (iii) Ignore future $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ messages with the same $\mathsf{sid}, \mathsf{ssid}$;
- (ii) Upon receiving $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, c)$ from Rec, where c in $[N]$:
  - (i) Send a notification $(\textsc{ReceiveNotify}, \mathsf{sid}, \mathsf{ssid})$ to $\mathcal{S}$;
  - (ii) Store $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, c)$;
  - (iii) Ignore future $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, \ldots)$ messages with the same $\mathsf{sid}, \mathsf{ssid}$;
- (iii) If both $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ and $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, c)$ are stored:
  - (i) For $i \in [N]$, pick random $m_i \leftarrow \{0,1\}^{\mathsf{length}}$;
  - (ii) Send $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid}, \{m_i\}_{i \in [N]})$ to Sen and $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, m_c)$ to Rec;

Corruption handling:

- (i) Upon receiving $(\textsc{Corrupt}, \mathsf{sid}, \mathsf{ssid}, P)$ from the adversary $\mathcal{S}$, if $P \in \mathcal{P}$:
  - (i) Set $\tilde{\mathcal{P}} := \tilde{\mathcal{P}} \cup \{P\}$;
  - (ii) Send $(\textsc{Input}, \mathsf{sid}, \mathsf{ssid}, P, x)$ to $\mathcal{S}$ if party P's input x is already defined;

Figure 9: 1-out-of-N random oblivious transfer functionality $\mathscr{F}^{1,N}_{U\text{–OT}}$.

Multi-instance Endemic Oblivious Transfer Functionality $\mathcal{F}^{\mathsf{num}}_{\text{E-OT}}$

It interacts with players $\mathcal{P} := \{\mathsf{Sen}, \mathsf{Rec}\}$ and the adversary $\mathcal{S}$. It is parameterized by the length of the messages length. num is the number of OT's to be generated. Let $\tilde{\mathcal{P}}$ be the set of corrupted parties. Initially, set $\tilde{\mathcal{P}} = \emptyset$.

Transfer:

- (i) Upon receiving $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ from Sen:
  - (i) Send a notification $(\textsc{SendNotify}, \mathsf{sid}, \mathsf{ssid})$ to $\mathcal{S}$;
  - (ii) Store $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$;
  - (iii) Ignore future $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ messages with the same $\mathsf{sid}, \mathsf{ssid}$;
- (ii) Upon receiving $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, \{c_i\}_{i \in [\mathsf{num}]})$ from Rec:
  - (i) Send a notification $(\textsc{ReceiveNotify}, \mathsf{sid}, \mathsf{ssid})$ to $\mathcal{S}$;
  - (ii) Store $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, \{c_i\}_{i \in [\mathsf{num}]})$;
  - (iii) Ignore future $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, \ldots)$ messages with the same $\mathsf{sid}, \mathsf{ssid}$;
- (iii) If both $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ and $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, \{c_i\}_{i \in [\mathsf{num}]})$ are stored:
  - (i) For $i \in [\mathsf{num}], j \in \{0,1\}$, pick random $m_i^j \leftarrow \{0,1\}^{\mathsf{length}}$;
  - (ii) If $\mathsf{Sen} \in \tilde{\mathcal{P}}$, wait for $(\textsc{FixMessage}, \mathsf{sid}, \mathsf{ssid}, \{\tilde{m}_i^j\}_{i \in [\mathsf{num}], j \in \{0,1\}})$ from $\mathcal{S}$, set $m_i^j := \tilde{m}_i^j$, for $i \in [\mathsf{num}], j \in \{0,1\}$;
  - (iii) If $\mathsf{Rec} \in \tilde{\mathcal{P}}$, wait for $(\textsc{FixMessage}, \mathsf{sid}, \mathsf{ssid}, \{\tilde{m}_i^{c_i}\}_{i \in [\mathsf{num}]})$ from $\mathcal{S}$, set $m_i^{c_i} := \tilde{m}_i^{c_i}$, for $i \in [\mathsf{num}]$;
  - (iv) Send $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid}, \{m_i^j\}_{i \in [\mathsf{num}], j \in \{0,1\}})$ to Sen and $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, \{m_i^{c_i}\}_{i \in [\mathsf{num}]})$ to Rec;

Corruption handling:

- (i) Upon receiving $(\textsc{Corrupt}, \mathsf{sid}, \mathsf{ssid}, P)$ from the adversary $\mathcal{S}$, if $P \in \mathcal{P}$:
  - (i) Set $\tilde{\mathcal{P}} := \tilde{\mathcal{P}} \cup \{P\}$;
  - (ii) Send $(\textsc{Input}, \mathsf{sid}, \mathsf{ssid}, P, x)$ to $\mathcal{S}$ if party P's input x is already defined;

Figure 10: Multi-instance endemic oblivious transfer functionality $\mathscr{F}^{\mathsf{num}}_{E\text{–OT}}$.

with endemic OT as base OT and apply the result of [47] to reduce the communication round. When it comes to the 1-out-of-N OT extension, results of [48, 49] can be adopted. The OT extension protocols securely realize the multi-instance version of the OT functionalities, and we provide the multi-instance 1-out-of-2 endemic OT functionality in Figure 10, which is similar to $\mathscr{F}_{E\text{–OT}}$. The multi-instance 1-out-of-2 uniform OT functionality can be found in Appendix A.1.

The Multi-instance Endemic Oblivious Transfer Protocol $\Pi^{\text{num}}_{\text{E-OT}}$

Setup: The sender Sen and the receiver Rec first agree on the SM2 key agreement protocol parameters with the assistance of the PKI. The parameters include the elliptic curve system parameters p, a, b, g, n, h, public keys $\mathsf{pk}_A, \mathsf{pk}_B$, and identifier hash values Z_A, Z_B, the key length is set as the OT message length.

```
(i) Upon receiving $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ from the environment $\mathcal{Z}$, Sen:
    (i) Compute $(m_A, t_A) \leftarrow \mathsf{MsgGen}()$;
    (ii) Send $m_A$ to Rec;
(ii) Upon receiving $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, \{c_i\}_{i\in[\mathsf{num}]})$ from the environment $\mathcal{Z}$, Rec:
    (i) For $i \in [\mathsf{num}]$:
        (i) Compute $(m_{B,i}, t_{B,i}) \leftarrow \mathsf{MsgGen}()$;
        (ii) Compute $\{r_i^j\}_{j\in[2]} := \mathsf{Program}(c, m_{B,i})$;
    (ii) Send $\{r_i^j\}_{i\in[\mathsf{num}], j\in[2]}$ to Sen;
(iii) Upon receiving $\{r_i^j\}_{i\in[\mathsf{num}], j\in[2]}$ from Rec, Sen:
    (i) For $i \in [\mathsf{num}], j \in [2]$:
        (i) Compute $m_{B,i}^j := \mathsf{Eval}(\{r_i^k\}_{k\in[2]}, j)$;
        (ii) Compute $p_i^j := \mathsf{PointGen}(t_A, m_A, m_{B,i}^j, \mathsf{sk}_A, \mathsf{pk}_B)$;
        (iii) Compute $k_i^j := \mathsf{KeyGen}(p_i^j, Z_A, Z_B)$;
    (ii) Return $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid}, \{k_i^j\}_{i\in[\mathsf{num}], j\in\{0,1\}})$ to the environment $\mathcal{Z}$;
(iv) Upon receiving $m_A$ from Sen, Rec:
    (i) For $i \in [\mathsf{num}]$:
        (i) Compute $p_i^{c_i} := \mathsf{PointGen}(t_{B,i}, m_{B,i}, m_A, \mathsf{sk}_B, \mathsf{pk}_A)$;
        (ii) Compute $k_i^{c_i} := \mathsf{KeyGen}(p_i^{c_i}, Z_A, Z_B)$;
    (ii) Return $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, \{k_i^{c_i}\}_{i\in[\mathsf{num}]})$ to the environment $\mathcal{Z}$;
```

Figure 11: Multi-instance endemic oblivious transfer protocol $\Pi^{\text{num}}_{\text{E-OT}}$.

5.2.1. Batching Base OT. The base OTs can be obtained by repeatedly invoking the single-instance functionality $\mathscr{F}^{1,2}_{E\text{-OT}}$ or $\mathscr{F}^{1,2}_{U\text{-OT}}$; however, the more efficient way is to design a protocol that directly realizes the multi-instance functionality. Our construction $\Pi^{\text{num}}_{E\text{-OT}}$ is essentially the same as $\Pi^{1,N}_{E\text{-OT}}$, where $N = 2$, except that Rec generates multiple message pairs $(m_{B,i}, t_{B,i})$ at once, and each message pair is used to generate one OT instance. The details can be found in Figure 11. This batching method saves Sen from repeatedly generating message pairs $m_{A,i}, t_{A,i}$, thus reducing the computation and communication costs. In [15], McQuoid et al. showed that this batching preserves the security of the original protocol when a tag is used in the generation of the KA protocol output to produce different OT results for each OT instance. In our protocol, we preserve the structure of the SM2 key agreement protocol, and we add the tag in the key derivation function KDF: we set the tag as $i\|1$ for the i-th KDF invocation. Therefore, the protocol $\Pi^{\text{num}}_{E\text{-OT}}$ in Figure 11 is secure, and we have the following theorem:

Theorem 15. *If ECDLP is hard in $\mathbb{G}$, the hash functions $\{hash_i^{\mathbb{G}}\}_{i\in[N]}$ and the key derivation function KDF are modeled as random oracles, and then, the protocol $\Pi^{num}_{E\text{-OT}}$ described in Figure 11 securely realizes $\mathscr{F}^{num}_{E\text{-OT}}$ described in Figure 10 against any PPT malicious adversary corrupting Sen or/and Rec.*

5.2.2. OT Extension in Semi-Honest Setting. In the semihonest setting, the protocol $\Pi^{1,N}_{E\text{-OT}}$ securely realizes the random OT functionality $\mathscr{F}^{1,2}_{U\text{-OT}}$ when $N = 2$, so the protocol $\Pi^{\text{num}}_{E\text{-OT}}$ securely realizes the multi-instance random OT functionality $\mathscr{F}^{\text{num}}_{U\text{-OT}}$. Therefore, we take 1-out-of-2 random OT as the base OT of the OT extension protocol. As depicted in Figure 12, the OT extension protocol $\Pi^{\text{semi}}_{\text{OTE}}$ needs λ base OTs to start the extension; typically, $\lambda = 128$ is used considering both security and performance. To generate the base OTs, the sender Sen of the outer protocol acts as the receiver, and it picks random select bits $\{r_i\}_{i\in[\lambda]}$ and sends $\{r_i\}_{i\in[\lambda]}$ to $\mathscr{F}^{\text{num}}_{U\text{-OT}}$; the receiver Rec of the outer protocol acts as the sender and sends $(\text{Send}, \text{sid}, \text{ssid})$ to $\mathscr{F}^{\text{num}}_{U\text{-OT}}$. The random OT functionality picks random $\{k_i^0, k_i^1\}_{i\in[\lambda]}$, and it sends $\{k_i^0, k_i^1\}_{i\in[\lambda]}$ to Rec and sends $\{k_i^{r_i}\}_{i\in[\lambda]}$ to Sen. After obtaining the base OTs, Rec forms the choice bits $\{c_i\}_{i\in[m]}$ as a column vector. For $i \in [\lambda]$, Rec computes the PRG to generate $t^i \longleftarrow \text{PRG}(k_i^0)$ and parses t^i as a column vector, and it sets $u^i := t^i \oplus \text{PRG}(k_i^1) \oplus C$. After that, Rec forms a $m \times \lambda$ matrix $T := (t^1, \cdots, t^\lambda)$. For $i \in [m]$, Rec computes $h_i^{c_i} \longleftarrow \text{hash}_{\text{length}}(i, T_i)$ as its random OT message where T_i is the i-th row of the matrix T. Subsequently, Rec sends $\{u^i\}_{i\in[\lambda]}$ and outputs. After obtaining the base OT results and $\{u^i\}_{i\in[\lambda]}$, Sen sets $q^i := (r_i \cdot u^i) \oplus \text{PRG}(k^i_{r_i})$ for $i \in [\lambda]$. It then forms a $m \times \lambda$ matrix $Q := (q^1, \cdots, q^\lambda)$ and a row vector $R := (r_1, \cdots, r_\lambda)$. For $i \in [m]$, Sen computes $h_i^0 \longleftarrow \text{hash}_{\text{length}}(i, Q_i)$ and $h_i^1 \longleftarrow \text{hash}_{\text{length}}(i, Q_i \oplus R)$ as its OT output.

Now we examine the correctness of $\Pi^{\text{semi}}_{\text{OTE}}$. For each column of the matrix Q, $Q^i = (r_i \cdot u^i) \oplus \text{PRG}(k_i^{r_i}) = (r_i \cdot (\text{PRG}(k_i^0) \oplus \text{PRG}(k_i^1) \oplus C)) \oplus \text{PRG}(k_i^{r_i})$, we can write

The Semi-honest Setting 1-out-of-2 Oblivious Transfer Extension Protocol $\Pi_{\text{OTE}}^{\text{semi}}$

We use $\text{num} = \lambda$ base OT's. The base OT's length is λ and the extended OT's length is length. $\text{PRG} : \{0,1\}^{\lambda} \to \{0,1\}^{m}$ is a pseudorandom number generator function and $\text{hash}_{\text{length}} : \{0,1\}^{*} \to \{0,1\}^{\text{length}}$ is a correlation robust hash function.

```
(i) Upon receiving $(\textsc{Extend}, \text{sid}, \{c_i\}_{i\in[m]})$ from the environment $\mathcal{Z}$, Rec:
    (i) Send $(\textsc{Send}, \text{sid}, \text{ssid})$ to $\mathcal{F}_{\text{U-OT}}^{\text{num}}$;
(ii) Upon receiving $(\textsc{Extend}, \text{sid})$ from the environment $\mathcal{Z}$, Sen:
    (i) For $i \in [\lambda]$, pick random $r_i \leftarrow \{0,1\}$;
    (ii) Send $(\textsc{Receive}, \text{sid}, \text{ssid}, \{r_i\}_{i\in[\lambda]})$ to $\mathcal{F}_{\text{U-OT}}^{\text{num}}$;
(iii) Upon receiving $(\textsc{Send}, \text{sid}, \text{ssid}, \{k_i^0, k_i^0\}_{i\in[\lambda]})$ from $\mathcal{F}_{\text{U-OT}}^{\text{num}}$, Rec:
    (i) Set $C := (c_1, \ldots, c_m)^{\mathsf{T}}$ as a column vector;
    (ii) For $i \in [\lambda]$:
        (i) Set $t^i \leftarrow \text{PRG}(k_i^0)$ as a column vector;
        (ii) Set $u^i := t^i \oplus \text{PRG}(k_i^1) \oplus C$ as a column vector;
    (iii) Set $T := (t^1, \ldots, t^{\lambda})$ as a $m \times \lambda$ matrix;
    (iv) For $i \in [m]$, compute $h_i^{c_i} := \text{hash}_{\text{length}}(i, T_i)$;
    (v) Send $\{u^i\}_{i\in[\lambda]}$ to Sen;
    (vi) Return $(\textsc{Extend}, \text{sid}, \{h_i^{c_i}\}_{i\in[m]})$ to the environment $\mathcal{Z}$;
(iv) Upon receiving $(\textsc{Receive}, \text{sid}, \text{ssid}, \{k_i^{r_i}\}_{i\in[\lambda]})$ from $\mathcal{F}_{\text{U-OT}}^{\text{num}}$ and receiving $\{u^i\}_{i\in[\lambda]}$ from Rec, Sen:
    (i) For $i \in [\lambda]$, set $q^i := (r_i \cdot u^i) \oplus \text{PRG}(k_i^{r_i})$ as a column vector;
    (ii) Set $Q := (q^1, \ldots, q^{\lambda})$ as a $m \times \lambda$ matrix;
    (iii) Set $R := (r_1, \ldots, r_{\lambda})$ as a row vector;
    (iv) For $i \in [m]$, compute $h_i^0 := \text{hash}_{\text{length}}(i, Q_i)$ and $h_i^1 := \text{hash}_{\text{length}}(i, Q_i \oplus R)$;
    (v) Return $(\textsc{Extend}, \text{sid}, \{h_i^0, h_i^1\}_{i\in[m]})$ to the environment $\mathcal{Z}$;
```

Figure 12: Semihonest setting 1-out-of-2 oblivious transfer extension protocol $\Pi_{\text{OTE}}^{\text{semi}}$.

Multi-instance Random Oblivious Transfer Functionality $\mathcal{F}_{\text{U-OT}}^{\text{num}}$

It interacts with players $\mathcal{P} := \{\text{Sen}, \text{Rec}\}$ and the adversary $\mathcal{S}$. It is parameterized by the length of the messages length. num is the number of OT's to be generated. Let $\tilde{\mathcal{P}}$ be the set of corrupted parties. Initially, set $\tilde{\mathcal{P}} = \emptyset$.

Transfer:

```
(i) Upon receiving $(\textsc{Send}, \text{sid}, \text{ssid})$ from Sen:
    (i) Send a notification $(\textsc{SendNotify}, \text{sid}, \text{ssid})$ to $\mathcal{S}$;
    (ii) Store $(\textsc{Send}, \text{sid}, \text{ssid})$;
    (iii) Ignore future $(\textsc{Send}, \text{sid}, \text{ssid})$ messages with the same sid, ssid;
(ii) Upon receiving $(\textsc{Receive}, \text{sid}, \text{ssid}, \{c_i\}_{i\in[\text{num}]})$ from Rec:
    (i) Send a notification $(\textsc{ReceiveNotify}, \text{sid}, \text{ssid})$ to $\mathcal{S}$;
    (ii) Store $(\textsc{Receive}, \text{sid}, \text{ssid}, \{c_i\}_{i\in[\text{num}]})$;
    (iii) Ignore future $(\textsc{Receive}, \text{sid}, \text{ssid}, \ldots)$ messages with the same sid, ssid;
(iii) If both $(\textsc{Send}, \text{sid}, \text{ssid})$ and $(\textsc{Receive}, \text{sid}, \text{ssid}, \{c_i\}_{i\in[\text{num}]})$ are stored:
    (i) For $i \in [\text{num}], j \in \{0,1\}$, pick random $m_i^j \leftarrow \{0,1\}^{\text{length}}$;
    (ii) Send $(\textsc{Send}, \text{sid}, \text{ssid}, \{m_i^j\}_{i\in[\text{num}], j\in[2]})$ to Sen and $(\textsc{Receive}, \text{sid}, \text{ssid}, \{m_i^{c_i}\}_{i\in[\text{num}]})$ to Rec;
```

Corruption handling:

```
(i) Upon receiving $(\textsc{Corrupt}, \text{sid}, \text{ssid}, P)$ from the adversary $\mathcal{S}$, if $P \in \mathcal{P}$:
    (i) Set $\tilde{\mathcal{P}} := \tilde{\mathcal{P}} \cup \{P\}$;
    (ii) Send $(\textsc{Input}, \text{sid}, \text{ssid}, P, x)$ to $\mathcal{S}$ if party $P$'s input $x$ is already defined;
```

Figure 13: Multi-instance random oblivious transfer functionality $\mathscr{F}_{U\text{-OT}}^{\text{num}}$.

$\text{PRG}(k_i^{r_i}) = (1 \oplus r_i) \cdot \text{PRG}(k_i^0) \oplus r_i \cdot \text{PRG}(k_i^1)$ as before, and this gives us $Q^i = \text{PRG}(k_i^0) \oplus r_i \cdot C = T^i \oplus r_i \cdot C$. Therefore, for each row of the matrix, we have $Q_i = T_i \oplus c_i \cdot R$, and thus, the protocol is correct. As for the security of the protocol, in [45], Asharov et al. prove that $\Pi_{\text{OTE}}^{\text{semi}}$ securely realizes $\mathscr{F}_{U\text{-OT}}^{\text{num}}$. Moreover, We can obtain a result similar to Corollary 14 that $\Pi_{\text{OTE}}^{\text{semi}}$ securely realizes the multi-instance endemic OT functionality $\mathscr{F}_{E\text{-OT}}^{\text{num}}$.

Theorem 16. *The protocol $\Pi_{\text{OTE}}^{\text{semi}}$ described in Figure 12 securely realizes $\mathscr{F}_{U\text{-OT}}^{\text{num}}$ described in Figure 13 against any PPT semihonest adversary corrupting Sen or/and Rec.*

The proof is done by Asharov et al. [45].

Corollary 17. *The protocol $\Pi_{\text{OTE}}^{\text{semi}}$ described in Figure 12 securely realizes $\mathscr{F}_{E\text{-OT}}^{\text{num}}$ described in Figure 10 against any PPT semihonest adversary corrupting Sen or/and Rec.*

The Malicious Setting 1-out-of-2 Oblivious Transfer Extension Protocol $\Pi^{\text{mal}}_{\text{OTE}}$

We use $\mathsf{num} = \lambda$ base OT's. The base OT's length is λ and the extended OT's length is length. We set $m' = m + \mu$ and $\mathbb{F} = \{0,1\}^{\lambda}$. $\mathsf{PRG} : \{0,1\}^{\lambda} \rightarrow \{0,1\}^{m'}$ is a pseudorandom function, $\mathsf{hash}^{m'}_{\mathbb{F}} : \{0,1\}^{m' \times \lambda} \rightarrow \mathbb{F}^{m'}$ is a hash function and $\mathsf{hash}_{\mathsf{length}} : \{0,1\}^{*} \rightarrow \{0,1\}^{\mathsf{length}}$ is a correlation robust hash function.

```
(i) Upon receiving $(\textsc{Extend}, \mathsf{sid}, \{c_i\}_{i\in[m]})$ from the environment $\mathcal{Z}$, Rec:
    (i) Send $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ to $\mathcal{F}^{\mathsf{num}}_{\text{E-OT}}$;
(ii) Upon receiving $(\textsc{Extend}, \mathsf{sid})$ from the environment $\mathcal{Z}$, Sen:
    (i) For $i \in [\lambda]$, pick random $r_i \leftarrow \{0,1\}$;
    (ii) Send $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, \{r_i\}_{i\in[\lambda]})$ to $\mathcal{F}^{\mathsf{num}}_{\text{E-OT}}$;
(iii) Upon receiving $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid}, \{k_i^0, k_i^0\}_{i\in[\lambda]})$ from $\mathcal{F}^{\mathsf{num}}_{\text{E-OT}}$, Rec:
    (i) For $i \in [\lambda]$, pick random $c'_i \leftarrow \{0,1\}$;
    (ii) Set $C := (c_1, \dots, c_m, c'_1, \dots, c'_\lambda)^{\mathsf{T}}$ as a column vector and $C' := (C, \dots, C)$ as a $m' \times \lambda$ matrix;
    (iii) For $i \in [\lambda]$:
        (i) Set $t^i \leftarrow \mathsf{PRG}(k_i^0)$ as a column vector;
        (ii) Set $u^i := t^i \oplus \mathsf{PRG}(k_i^1) \oplus C$ as a column vector;
    (iv) Set $T := (t^1, \dots, t^\lambda)$ and $U := (u^1, \dots, u^\lambda)$ as $m' \times \lambda$ matrices;
    (v) Compute $\{\chi_1, \dots, \chi_{m'}\} := \mathsf{hash}^{m'}_{\mathbb{F}}(U)$;
    (vi) Set $v := \bigoplus_{i\in[m']}(\chi_i \cdot T_i)$ and $w := \bigoplus_{i\in[m']}(\chi_i \cdot C'_i)$;
    (vii) For $i \in [m]$, compute $h_i^{c_i} := \mathsf{hash}_{\mathsf{length}}(i, T_i)$;
    (viii) Send $U, v, w$ to Sen;
    (ix) Return $(\textsc{Extend}, \mathsf{sid}, \{h_i^{c_i}\}_{i\in[m]})$ to the environment $\mathcal{Z}$;
(iv) Upon receiving $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, \{k_i^{r_i}\}_{i\in[\lambda]})$ from $\mathcal{F}^{\mathsf{num}}_{\text{U-OT}}$ and receiving $U, v, w$ from Rec, Sen:
    (i) Compute $\{\chi_1, \dots, \chi_{m'}\} := \mathsf{hash}^{m'}_{\mathbb{F}}(U)$;
    (ii) For $i \in [\lambda]$, set $q^i := (r_i \cdot U^i) \oplus \mathsf{PRG}(k_i^{r_i})$ as a column vector;
    (iii) Set $Q := (q^1, \dots, q^\lambda)$ as a $m' \times \lambda$ matrix;
    (iv) Set $R := (r_1, \dots, r_\lambda)$ as a row vector;
    (v) Assert $\bigoplus_{i\in[m']}(\chi_i \cdot Q_i) = v \oplus R \cdot w$;
    (vi) For $i \in [m]$, compute $h_i^0 := \mathsf{hash}_{\mathsf{length}}(i, Q_i)$ and $h_i^1 := \mathsf{hash}_{\mathsf{length}}(i, Q_i \oplus R)$;
    (vii) Return $(\textsc{Extend}, \mathsf{sid}, \{h_i^0, h_i^1\}_{i\in[m]})$ to the environment $\mathcal{Z}$;
```

FIGURE 14: Malicious setting 1-out-of-2 oblivious transfer extension protocol $\Pi^{\text{mal}}_{\text{OTE}}$.

The proof is similar to the proof of Corollary 14.

5.2.3. OT Extension in Malicious Setting. The malicious setting is more difficult to handle. When the base OTs are endemic OTs, we can only extend them to more endemic OTs, and the sender Sen needs to check the consistency of the messages sent by the receiver Rec. We provide the malicious setting OT extension protocol $\Pi^{\text{mal}}_{\text{OTE}}$ in Figure 14. The main process of extending oblivious transfer remains the same as the semihonest setting protocol $\Pi^{\text{semi}}_{\text{OTE}}$, so is the correctness of the protocol. However, a malicious Rec can violate the requirement that the same choice bits should be used when computing the vectors $\{u^i\}_{i\in[\lambda]}$. In [46], Rec needs to prove its honesty in zero knowledge, and generally speaking, the consistency check uses a random linear combination of the row vectors of the matrices, and the coefficients of the linear combination should be unpredictable to Rec, e.g., they are randomly picked by Sen. In [47], Doerner et al. use Fiat-Shamir heuristic [50] to make the zero-knowledge proof process noninteractive: the coefficients are generated by a hash function taking the matrix U as input. The formal security proof of the protocol $\Pi^{\text{mal}}_{\text{OTE}}$ can be found in [12].

Theorem 18. *The protocol Π^{mal}_{OTE} described in Figure 14 securely realizes $\mathscr{F}^{num}_{E-OT}$ described in Figure 10 against any PPT malicious adversary corrupting Sen or/and Rec.*

The proof is done by Masny and Rindal [12].

6. Generate the Beaver Triple

A wide range of MPC protocols working on circuits requires heavy computation and huge communication to compute AND gates and multiplication gates. To speed up the MPC protocols, a research trend is to split the protocol into a preprocessing phase independent of parties' input and an online phase where the computation proceeds using actual input and data from the preprocessing phase. Therefore, parties can run the preprocessing phase whenever they are available and respond instantly when the computation needs to proceed.

As for secret-sharing-based MPC protocols, Beaver introduces a notion of the Beaver triple (or multiplication triple) in [51]. Basically, a Beaver triple consists of shared triples $\{a_i, b_i, c_i\}_{i\in[N]}$, where a_i, b_i, c_i are chosen randomly with $\oplus c_i = \oplus a_i \wedge \oplus b_i$ that always holds. The length of the Beaver triple can vary with applications, and in this work, we refer the Beaver triple with a length larger than 1 as a

The Beaver/Multiplication Triple Generation Functionality $\mathcal{F}_{\text{TRIPLE}}$

It interacts with players $\mathcal{P} := \{P_1, \ldots, P_N\}$ and the adversary $\mathcal{S}$. It is parameterized by the length of the triples length. Let $\tilde{\mathcal{P}}$ be the set of corrupted parties. Initially, set $\tilde{\mathcal{P}} = \emptyset$.

Generate:

- (i) Upon receiving $(\textsc{Generate}, \mathsf{sid}, \mathsf{ssid}, P_i)$ from P_i:
  - (i) Send a notification $(\textsc{GenerateNotify}, \mathsf{sid}, \mathsf{ssid}, P_i)$ to $\mathcal{S}$;
  - (ii) Store $(\textsc{Generate}, \mathsf{sid}, \mathsf{ssid}, P_i)$;
  - (iii) Ignore future $(\textsc{Generate}, \mathsf{sid}, \mathsf{ssid}, P_i)$ messages with the same $\mathsf{sid}, \mathsf{ssid}, P_i$;
- (ii) If all of $(\textsc{Generate}, \mathsf{sid}, \mathsf{ssid}, P_i)$ are stored for $i \in [N]$:
  - (i) Wait for $(\textsc{FixTriple}, \mathsf{sid}, \mathsf{ssid}, P_i, (a_i, b_i, c_i))$ from $\mathcal{S}$ for $P_i \in \tilde{\mathcal{P}}$:
  - (ii) For $P_i \notin \tilde{\mathcal{P}}$, pick random $a_i, b_i, c_i \leftarrow \{0,1\}^{\mathsf{length}}$;
  - (iii) Reset $c_i := (\sum a_j) \cdot (\sum b_j) - (\sum_{j \neq i} c_j)$ for the smallest i such that $P_i \notin \tilde{\mathcal{P}}$;
  - (iv) For $i \in [N]$, send $(\textsc{Generate}, \mathsf{sid}, \mathsf{ssid}, P_i, (a_i, b_i, c_i))$ to P_i;

Corruption handling:

- (i) Upon receiving $(\textsc{Corrupt}, \mathsf{sid}, \mathsf{ssid}, P_i)$ from the adversary $\mathcal{S}$, if $P_i \in \mathcal{P}$:
  - (i) Set $\tilde{\mathcal{P}} := \tilde{\mathcal{P}} \cup \{P_i\}$;

Figure 15: Beaver/multiplication triple generation functionality $\mathscr{F}_{\text{triple}}$.

multiplication triple, and a Beaver triple with a length of 1 is simply called the Beaver triple.

As depicted in Figure 15, the triple-generation functionality $\mathscr{F}_{\text{triple}}$ first waits for all parties to send a (Generate, sid , P_i) instruction. After that, it allows the adversary $\mathscr{S}$ to determine the content of the triple received by the corrupted P_i by sending (FixTriple, sid, P_i, (a_i, b_i, c_i)). Subsequently, as for the parties not corrupted, $\mathscr{F}_{\text{triple}}$ picks uniformly random a_i, b_i, c_i. However, to ensure that $\oplus c_i = \oplus a_i \wedge \oplus b_i$, $\mathscr{F}_{\text{triple}}$ chooses the party with the smallest index i among the uncorrupted parties and sets $c_i := (\sum a_j) \cdot (\sum b_j) - (\sum_{j \neq i} c_j)$. At the end, $\mathscr{F}_{\text{triple}}$ sends (Generate, sid, P_i, (a_i, b_i, c_i)) back to P_i.

6.1. Two-Party Beaver Triple Generation. We first introduce how to generate the Beaver triple among two parties using endemic OT. In an OT execution, the receiver sends b to obtain m_b, and the sender obtains two messages m_0, m_1. Notice that m_b can be represented as $b \wedge m_1 \oplus (1 \oplus b) \wedge m_0$, and if we set $a = m_0 \oplus m_1$, then $m_0 \oplus m_b = b \wedge m_0 \oplus b \wedge m_1 = a \wedge b$. Therefore, invoking the endemic OT twice with P_1 and P_2 playing the sender in turn is directly a Beaver triple-generation protocol Π_{triple}, which can be found in Figure 16. In Π_{triple}, party P_i first picks random b_i, and then, it invokes the endemic OT functionality $\mathscr{F}_{E\text{-OT}}^{1,2}$ as sender and receiver, respectively. When $\mathscr{F}_{E\text{-OT}}^{1,2}$ sends m_0^i, m_1^i, and $m_{b_i}^{3-i}$ back to P_i, P_i sets $a_i := m_0^i \oplus m_1^i$ and $c_i := a_i \wedge b_i \oplus m_0^i \oplus m_{b_i}^{3-i}$.

In the implementation, the main cost of the protocol Π_{triple} is to invoke $\mathscr{F}_{E\text{-OT}}^{1,2}$ twice. When $\mathscr{F}_{E\text{-OT}}^{1,2}$ is instantiated with $\Pi_{E\text{-OT}}^{1,N}$, which is a one-round protocol, the protocol Π_{triple} also has round complexity 1. Besides, $\Pi_{E\text{-OT}}^{1,N}$ only needs to transfer 3 group elements in total, which means Π_{triple} only needs 6. Moreover, we can use the multi-instance OT functionality to generate a bunch of OT instances in advance, which further reduces the computation and communication costs. Therefore, Π_{triple} can be extremely suitable for lightweight devices in extreme network environments with high delay and low bandwidth.

Although endemic OT is a weak version of general random OT, the protocol Π_{triple} is still secure in the malicious setting, and we provide the theorem together with the proof below.

Theorem 19. *The protocol Π_{triple} described in Figure 16 securely realizes $\mathscr{F}_{triple}$ described in Figure 15 with length = 1 in the $\mathscr{F}_{E\text{-}OT}^{1,2}$-hybrid model against any PPT malicious adversary corrupting P_1 or P_2.*

The proof can be found in Appendix B.3.

Given this two-party Beaver triple-generation protocol Π_{triple}, we can use it to generate sufficiently many Beaver triples in the preprocessing phase and carry out a GMW-style two-party computation protocol [37] in the online phase, where the XOR gates can be computed locally and the AND gates consume one Beaver triple each and need communication. The details of the protocol Π_{2PC} can be found in Figure 17. This provides an efficient solution to a generic two-party computation over the Boolean circuits.

Theorem 20. *The protocol Π_{2PC} described in Figure 17 securely realizes $\mathscr{F}_{mpc}^{f}$ described in Figure 1 in the $\mathscr{F}_{triple}$-hybrid model against any PPT semihonest adversary corrupting P_1 or P_2.*

The proof is done by combining the result of [37, 51].

6.2. Multiparty Multiplication Triple Generation. We can extend the Beaver triple-generation protocol Π_{triple} to the multiparty setting and generate multiplication triple of length ≥ 1 with one more round communication. Therefore, we obtain the protocol $\Pi_{\text{triple}}^{N,\text{length}}$ described in Figure 18 which is secure in the semihonest setting, even when the endemic

```
Two Party Beaver Triple Generation Protocol $\Pi_{\text{TRIPLE}}$ in the $\mathcal{F}^{1,2}_{\text{E-OT}}$-hybrid model

The OT message length is set to 1.

(i) For $i \in [2]$, upon receiving $(\text{Generate}, \text{sid}, \text{ssid}, P_i)$ from the environment $\mathcal{Z}$, the party $P_i$:
    (i) Pick random $b_i \leftarrow \{0,1\}$;
    (ii) Send $(\text{Send}, \text{sid}, \text{ssid}_i)$ and $(\text{Receive}, \text{sid}, \text{ssid}_{3-i}, b_i)$ to $\mathcal{F}^{1,2}_{\text{E-OT}}$;
(ii) For $i \in [2]$, upon receiving $(\text{Send}, \text{sid}, \text{ssid}_i, (m_0^i, m_1^i))$ and $(\text{Receive}, \text{sid}, \text{ssid}_{3-i}, m_{b_i}^{3-i})$ from $\mathcal{F}^{1,2}_{\text{E-OT}}$, the party $P_i$:
    (i) Set $a_i := m_0^i \oplus m_1^i$;
    (ii) Set $c_i := a_i \wedge b_i \oplus m_0^i \oplus m_{b_i}^{3-i}$;
    (iii) Return $(\text{Generate}, \text{sid}, \text{ssid}, P_i, (a_i, b_i, c_i))$ to the environment $\mathcal{Z}$;
```

Figure 16: Two-party Beaver triple generation protocol Π_{triple} in the $\mathcal{F}^{1,2}_{E\text{-}OT}$-hybrid model.

```
GMW-style Two Party Computation Protocol $\Pi_{\text{2PC}}$ in the $\mathcal{F}_{\text{TRIPLE}}$-hybrid model

Let $f$ be the circuit to be computed, WLOG, we assume $f$ only consists of AND gates and XOR gates. The triple length is set to 1.

Initialization Phase

(i) For $j \in [2]$, upon receiving $(\text{Compute}, \text{sid}, x_j := (x_{j,1}, \ldots, x_{j,n_j}))$ from the environment $\mathcal{Z}$, the party $P_j$:
    (i) For $k \in [n_j]$, pick random $x_{j,k}^1 \leftarrow \{0,1\}$ and set $x_{j,k}^2 := x_{j,k} \oplus x_{j,k}^1$;
    (ii) Send $\{x_{j,k}^{3-j}\}_{k \in [n_j]}$ to $P_{3-j}$;

Evaluation Phase

After the initialization phase, $P_1$ and $P_2$ obtain their shares of the input wires. They then evaluate the circuit gate by gate.

(i) For an XOR gate $(\text{id}, \alpha, \beta, \gamma)$:
    (i) For $j \in [2]$, the party $P_j$ computes $x_\gamma^j = x_\alpha^j \oplus x_\beta^j$;
(ii) For an AND gate $(\text{id}, \alpha, \beta, \gamma)$:
    (i) For $j \in [2]$, $P_j$ send $(\text{Generate}, \text{sid}, \text{ssid}, P_j)$ to $\mathcal{F}_{\text{TRIPLE}}$;
    (ii) For $j \in [2]$, upon receiving $(\text{Generate}, \text{sid}, \text{ssid}, P_j, (a_j, b_j, c_j))$ from $\mathcal{F}_{\text{TRIPLE}}$, the party $P_j$
        (i) Set $d_j := x_\alpha^j \oplus a_j$ and $e_j := x_\beta^j \oplus b_j$;
        (ii) Send $d_j, e_j$ to $P_{3-j}$;
    (iii) For $j \in [2]$, upon receiving $d_{3-j}, e_{3-j}$ from $P_{3-j}$, the party $P_j$:
        (i) Set $d := d_1 \oplus d_2$ and $e := e_1 \oplus e_2$;
        (ii) Set $x_\gamma^j := d \wedge e \oplus d \wedge b_j \oplus e \wedge a_j \oplus c_j$;

Output Phase

(i) For $j \in [2]$, the party $P_j$ broadcast $\{x_i^j\}_{i \in \text{out}}$;
(ii) For $j \in [2]$, upon receiving $\{x_i^{3-j}\}_{i \in \text{out}}$ from $P_{3-j}$, the party $P_j$:
    (i) For $i \in \text{out}$, compute $y_i := x_i^1 \oplus x_i^2$;
    (ii) Return $(\text{Compute}, \text{sid}, P_j, y := \{y_i\}_{i \in \text{out}})$ to the environment $\mathcal{Z}$;
```

Figure 17: GMW-style two-party computation protocol Π_{2PC} in the $\mathcal{F}_{\text{triple}}$-hybrid model.

OT functionality is used which gives more power to the adversary $\mathcal{A}$.

The core idea is still using the OT functionality to generate correlated messages, and again, we can simply invoke the multi-instance OT functionality and use the OT extension technique to generate polynomially many OT instances in advance. In Π_{triple}, all computations are on the ring $\mathbb{Z}_2$, while now, we consider $\mathbb{Z}_{2_{\text{length}}}$. Assume an endemic OT outputs m_0, m_1 to sender and m_b to receiver for choice bit b, and it holds that $m_b = b \cdot m_1 + (1-b) \cdot m_0 \bmod 2^{\text{length}}$.

Consider the simple case where length = 1, we have

$$\begin{aligned} s_{j,i} - m_{i,j}^0 &\equiv m_{j,i}^{b_i} + b_i \cdot r_{j,i} - m_{i,j}^0 \\ &\equiv b_i \cdot m_{j,i}^1 + (1-b_i) \cdot m_{j,i}^0 + b_i \\ &\quad \cdot \left(a_j + m_{j,i}^0 - m_{j,i}^1\right) - m_{i,j}^0 \\ &\equiv m_{j,i}^0 + b_i \cdot a_j - m_{i,j}^0, \end{aligned} \tag{9}$$

Multi-Party Triple Generation Protocol $\Pi^{N,\mathsf{length}}_{\text{TRIPLE}}$ in the $\mathcal{F}^{1,2}_{\text{E-OT}}$-hybrid model

We works on the ring $\mathbb{Z}_{2^{\mathsf{length}}}$, where length is the triple length. The OT message length is set to length.

```
(i) For $i \in [N]$, upon receiving $(\textsc{Generate}, \mathsf{sid}, \mathsf{ssid}, P_i)$ from the environment $\mathcal{Z}$, the party $P_i$:
    (i) Pick random $a_i, b_i \leftarrow \{0,1\}^{\mathsf{length}}$;
    (ii) For $j \neq i, k \in [\mathsf{length}]$, send $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid}_{i,j,k})$ and $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}_{j,i,k}, b_i[k])$ to $\mathcal{F}^{1,2}_{\text{E-OT}}$;
(ii) For $i \in [N]$, upon receiving $\{(\textsc{Send}, \mathsf{sid}, \mathsf{ssid}_{i,j,k}, m^0_{i,j,k}, m^1_{i,j,k})\}_{j\neq i, k\in[\mathsf{length}]}$ and
     $\{(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}_{j,i}, m^{b_i[k]})\}_{j\neq i, k\in[\mathsf{length}]}$ from $\mathcal{F}^{1,2}_{\text{E-OT}}$, the party $P_i$:
    (i) For $j \neq i, k \in [\mathsf{length}]$, set $r_{i,j,k} := a_i + m^0_{i,j,k} - m^1_{i,j,k} \mod 2^{\mathsf{length}}$;
    (ii) For $j \neq i$, send $\{r_{i,j,k}\}_{k\in[\mathsf{length}]}$ to $P_j$;
(iii) For $i \in [N]$, upon receiving $\{r_{j,i,k}\}_{k\in[\mathsf{length}]}$ from $\{P_j\}_{j\neq i}$, the party $P_i$:
    (i) For $j \neq i, k \in [\mathsf{length}]$, set $s_{j,i,k} := m^{b_i[k]}_{j,i,k} + b_i[k] \cdot r_{j,i,k} \mod 2^{\mathsf{length}}$;
    (ii) Set $c_i := a_i \cdot b_i + (\sum_{j\neq i}\sum_k (s^{j,i,k} - m^{i,j,k}_0) \cdot 2^{k-1}) \mod 2^{\mathsf{length}}$;
    (iii) Return $(\textsc{Generate}, \mathsf{sid}, \mathsf{ssid}, P_i, (a_i, b_i, c_i))$ to the environment $\mathcal{Z}$;
```

Figure 18: Multiparty multiplication triple generation protocol $\Pi^{N,\mathrm{length}}_{\mathrm{triple}}$ in the $\mathscr{F}^{1,2}_{\mathrm{E-OT}}$-hybrid model.

SPDZ-style Multi-Party Computation Protocol Π^{MPC} in the $\mathcal{F}_{\text{TRIPLE}}$-hybrid model

Let f be the circuit to be computed, WLOG, we assume f only consists of addition gates and multiplication gates over $\mathbb{Z}_{2^{\mathsf{length}}}$. The triple length is set to length and the computations are over $\mathbb{Z}_{2^{\mathsf{length}}}$.

Initialization Phase

```
(i) For $j \in [N]$, upon receiving $(\textsc{Compute}, \mathsf{sid}, x_j := (x_{j,1}, \ldots, x_{j,n_j}))$ from the environment $\mathcal{Z}$, the party $P_j$:
    (i) For $k \in [n_j]$, for $i \neq j$, pick random $x^i_{j,k} \leftarrow \{0,1\}^{\mathsf{length}}$;
    (ii) For $k \in [n_j]$, set $x^j_{j,k} := x_{j,k} - \sum_{i\neq j} x^i_{j,k}$;
    (iii) For $i \neq j$, send $\{x^i_{j,k}\}_{k\in[n_j]}$ to $P_i$;
```

Evaluation Phase

After the initialization phase, the parties $P_1, \ldots, P_N$ obtain their shares of the input wires. They then evaluate the circuit gate by gate.

```
(i) For an addition gate $(\mathsf{id}, \alpha, \beta, \gamma)$:
    (i) For $j \in [N]$, the party $P_j$ computes $x^j_\gamma = x^j_\alpha \oplus x^j_\beta$;
(ii) For a multiplication gate $(\mathsf{id}, \alpha, \beta, \gamma)$:
    (i) For $j \in [N]$, $P_j$ send $(\textsc{Generate}, \mathsf{sid}, \mathsf{ssid}, P_j)$ to $\mathcal{F}_{\text{TRIPLE}}$;
    (ii) For $j \in [N]$, upon receiving $(\textsc{Generate}, \mathsf{sid}, \mathsf{ssid}, P_j, (a_j, b_j, c_j))$ from $\mathcal{F}_{\text{TRIPLE}}$, the party $P_j$:
        (i) Set $d_j := x^j_\alpha - a_j$ and $e_j := x^j_\beta - b_j$;
        (ii) Broadcast $d_j, e_j$;
    (iii) For $j \in [N]$, upon receiving $\{d_i, e_i\}_{i\neq j}$ from $\{P_i\}_{i\neq j}$, the party $P_j$:
        (i) Set $d := \sum d_i$ and $e := \sum e_i$;
        (ii) Set $x^j_\gamma := d \cdot e + d \cdot b_j + e \cdot a_j + c_j$;
```

Output Phase

```
(i) For $j \in [N]$, the party $P_j$ sends $\{x^j_k\}_{k\in\mathsf{out}}$ to $\{P_i\}_{i\neq j}$;
(ii) For $j \in [N]$, upon receiving $\{x^i_k\}_{k\in\mathsf{out}}$ from $\{P_i\}_{i\neq j}$, the party $P_j$:
    (i) For $k \in \mathsf{out}$, compute $y_i := \sum x^i_k$;
    (ii) Return $(\textsc{Compute}, \mathsf{sid}, P_j, y := \{y_k\}_{k\in\mathsf{out}})$ to the environment $\mathcal{Z}$;
```

Figure 19: SPDZ-style multiparty computation protocol Π^{MPC} in the $\mathscr{F}_{\mathrm{triple}}$-hybrid model.

The Private Set Intersection Protocol Π_{PSI} in the $\mathcal{F}^{1,2}_{\text{U-OT}}$-hybrid model

```
Sen and Rec first agree on the protocol parameters $m, w, \ell_1, \ell_2$, hash functions $\mathsf{hash}_{\ell_1} : \{0,1\}^* \to \{0,1\}^{\ell_1}$ and
$\mathsf{hash}_{\ell_2} : \{0,1\}^w \to \{0,1\}^{\ell_2}$, and the pseudorandom function $\mathsf{PRF} : \{0,1\}^\lambda \times \{0,1\}^{\ell_1} \to [m]^w$.

(i) Upon receiving $(\textsc{Compute}, \mathsf{sid}, \mathsf{Rec}, Y = (y_1, \dots, y_{n_2}))$ from the environment $\mathcal{Z}$, Rec:
    (i) Initialize an $m \times w$ binary matrix $D$ to all 1's;
    (ii) Pick random $k \leftarrow \{0,1\}^\lambda$;
    (iii) For $i \in [n_2]$:
        (i) Set $v_i := \mathsf{PRF}_k(\mathsf{hash}_{\ell_1}(y_i))$;
        (ii) For $j \in [w]$, set $D^j[v[j]] = 0$;
    (iv) For $i \in [w]$, send $(\textsc{Send}, \mathsf{sid})$ to $\mathcal{F}^{1,2}_{\text{U-OT}}$ as Sen;
(ii) Upon receiving $(\textsc{Compute}, \mathsf{sid}, \mathsf{Sen}, X = (x_1, \dots, x_{n_1}))$ from the environment $\mathcal{Z}$, Sen:
    (i) Pick random $s \leftarrow \{0,1\}^w$;
    (ii) For $i \in [w]$, send $(\textsc{Receive}, \mathsf{sid}, s[i])$ to $\mathcal{F}^{1,2}_{\text{U-OT}}$ as Rec;
(iii) Upon receiving $(\textsc{Send}, \mathsf{sid}, \{r_i^0, r_i^1\})$ from $\mathcal{F}^{1,2}_{\text{U-OT}}$ for $i \in [w]$, Rec:
    (i) Form a $m \times w$ binary matrix $A$ where $A^i = r_i^0$, for $i \in [w]$;
    (ii) Set $B = A \oplus D$;
    (iii) For $i \in [w]$, set $\Delta_i = B^i \oplus r_i^1$;
    (iv) Send $\{\Delta_i\}_{i \in [w]}, k$ to Sen;
(iv) Upon receiving $(\textsc{Receive}, \mathsf{sid}, r_i^{s[i]})$ from $\mathcal{F}^{1,2}_{\text{U-OT}}$ for $i \in [w]$ and receiving $\{\Delta_i\}_{i \in [w]}, k$ from Rec, Sen:
    (i) Form a $m \times w$ binary matrix $C$ where $C^i = r_i^{s[i]} \oplus s[i] \cdot \Delta_i$;
    (ii) For $i \in [n_1]$:
        (i) Set $u_i := \mathsf{PRF}_k(\mathsf{hash}_{\ell_1}(x_i))$;
        (ii) Set $\phi_i := \mathsf{hash}_{\ell_2}(C^1[u_i[1]] || \dots || C^w[u_i[w]])$;
    (iii) Send $\Phi := \{\phi_i\}_{i \in [n_1]}$ to Rec;
    (iv) Return $(\textsc{Compute}, \mathsf{sid})$ to the environment $\mathcal{Z}$;
(v) Upon receiving $\Phi$ from Sen, Rec:
    (i) For $i \in [n_2]$, Set $\psi_i := \mathsf{hash}_{\ell_2}(A^1[v_i[1]] || \dots || A^w[v_i[w]])$;
    (ii) Set $\Psi := \{\psi_i\}_{i \in [n_2]}$;
    (iii) Set $\Omega := \Phi \cap \Psi$;
    (iv) Set $\mathsf{Res} := \{y_i | \psi_i \in \Omega\}$;
    (v) Return $(\textsc{Compute}, \mathsf{sid}, \mathsf{Res})$ to the environment $\mathcal{Z}$;
```

Figure 20: Private set intersection protocol Π_{PSI} in the $\mathcal{F}^{1,2}_{U-OT}$-hybrid model.

and we can extend Equation (9) to Equation (10), which proves the correctness of the protocol $\Pi^{N,\text{length}}_{\text{triple}}$.

$$\begin{aligned}\sum_i c_i &\equiv \sum_i \left(a_i \cdot b_i + \sum_{j \neq i} \left(s_{j,i} - m^0_{i,j} \right) \right) \\ &\equiv \sum_i \left(a_i \cdot b_i + \sum_{j \neq i} \left(m^0_{j,i} + b_i \cdot a_j - m^0_{i,j} \right) \right) \\ &\equiv \left(\sum_i a_i \right) \cdot \left(\sum_i b_i \right).\end{aligned} \tag{10}$$

Now, we proceed to provide the security of the protocol $\Pi^{N,\text{length}}_{\text{triple}}$.

Theorem 21. *The protocol $\Pi^{N,length}_{triple}$ described in Figure 18 securely realizes $\mathcal{F}_{triple}$ described in Figure 15 with length ≥ 1 in the $\mathcal{F}^{1,2}_{E-OT}$-hybrid model against any PPT semihonest adversary corrupting no more than $N-1$ parties.*

The proof can be found in Appendix B.4.

Given this multiparty multiplication triple-generation protocol $\Pi^{N,\text{length}}_{\text{triple}}$, we can use it in the semihonest SPDZ-style multiparty computation protocol [25], where each multiplication gate consumes one multiplication triple. Note that the original SPDZ protocol is designed for the malicious setting, and it includes information-theoretic MAC to ensure correctness, while in the semihonest setting, these checks can be removed. The details of the resulting MPC protocol Π^{MPC} can be found in Figure 19. Π^{MPC} provides an efficient solution to generic multiparty computation over the arithmetic circuits, which are more powerful than the Boolean circuits.

Theorem 22. *The protocol Π^{MPC} described in Figure 19 securely realizes $\mathcal{F}^f_{mpc}$ described in Figure 1 in the $\mathcal{F}_{triple}$-hybrid model against any PPT semihonest adversary corrupting up to $N-1$ parties.*

The proof is done by combining the result of [37, 51].

7. Private Set Intersection from OT

Apart from generic MPC protocols, there are also protocols dedicated for special usage, e.g., PSI. The PSI protocol Π_{PSI}

Sender Chosen Oblivious Transfer Protocol $\Pi^{1,N}_{\text{S-OT}}$ in the $\mathcal{F}^{1,N}_{\text{E-OT}}$-hybrid model

(i) Upon receiving $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid}, \{m_i\}_{i\in[N]})$ from the environment $\mathcal{Z}$, Sen:
  (i) Send $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid})$ to $\mathcal{F}^{1,N}_{\text{E-OT}}$;
(ii) Upon receiving $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, c)$ from the environment $\mathcal{Z}$, Rec:
  (i) Send $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, c)$ to $\mathcal{F}^{1,N}_{\text{E-OT}}$;
(iii) Upon receiving $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid}, \{k_i\}_{i\in[N]})$ from $\mathcal{F}^{1,N}_{\text{E-OT}}$, Sen:
  (i) For $i \in [N]$, sets $m'_i := m_i \oplus k_i$;
  (ii) Send $\{m'_i\}_{i\in[N]}$ to Rec;
  (iii) Return $(\textsc{Send}, \mathsf{sid}, \mathsf{ssid}, \{m_i\}_{i\in[N]})$ to the environment $\mathcal{Z}$;
(iv) Upon receiving $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, k_c)$ from $\mathcal{F}^{1,N}_{\text{E-OT}}$ and receiving $\{m'_i\}_{i\in[N]}$ from Sen, Rec:
  (i) Set $m_c := m'_c \oplus k_c$;
  (ii) Return $(\textsc{Receive}, \mathsf{sid}, \mathsf{ssid}, m_c)$ to the environment $\mathcal{Z}$;

Figure 21: Sender chosen oblivious transfer protocol $\Pi^{1,N}_{S\text{-}OT}$ in the $\mathscr{F}^{1,N}_{E\text{-}OT}$-hybrid model.

(Figure 5) is taken from the work of Chase and Miao [52], and we instantiate the OT functionality with $\Pi^{1,N}_{E\text{-}OT}$ as another application of this endemic OT protocol. When the required OT number is large, we can use the OT extension technique to extend the number of OT instances.

The security of Π_{PSI} in the semihonest setting directly follows the result of [52] since according to Theorem 13, the protocol $\Pi^{1,N}_{E\text{-}OT}$ (as well as the protocol $\Pi^{\mathrm{semi}}_{\mathrm{OTE}}$) securely realizes the random OT functionality in this setting. Moreover, Chase and Miao consider one-sided malicious security, where the Sen can be maliciously corrupted by the adversary. However, we notice that $\mathscr{F}_{E\text{-}OT}$ does not meet the security requirement of Π_{PSI}, and in such a case, the protocol can be insecure. As stated above, in the malicious setting, $\Pi^{1,N}_{E\text{-}OT}$ only realizes $\mathscr{F}^{1,2}_{E\text{-}OT}$, which allows the adversary $\mathscr{A}$ to control the OT messages. Specifically, $\mathscr{A}$ can influence the protocol output by changing the OT messages, enabling the environment to distinguish between the real world and the ideal world. We provide the result along with its proof below.

Theorem 23. *The protocol Π_{PSI} described in Figure 20 is not secure against a PPT malicious adversary corrupting Rec when the endemic OT functionality $\mathscr{F}^{1,2}_{E\text{-}OT}$ is used even if F is a secure PRF, $hash_{\ell_1}$ and $hash_{\ell_2}$ are modeled as random oracles, and parameters m, w, ℓ_1, ℓ_2 are chosen properly.*

Proof. To prove Theorem 23, we construct an adversary $\mathscr{A}$ and an environment $\mathscr{Z}$ such that for any PPT simulator $\mathscr{S}$, $\mathscr{Z}$ can distinguish between (i) the real execution $\mathrm{exec}^{\mathscr{F}^{1,2}_{E\text{-}OT}}_{\Pi_{\mathrm{PSI}},\mathscr{A},\mathscr{Z}}$, where the parties $\mathscr{P} := \{\mathrm{Sen}, \mathrm{Rec}\}$ run protocol Π_{PSI} in the $\mathscr{F}^{1,2}_{E\text{-}OT}$-hybrid model and the corrupted Sen is controlled by $\mathscr{A}$, and (ii) the ideal execution $\mathrm{exec}_{\mathscr{F}_{\mathrm{PSI}},\mathscr{S},\mathscr{Z}}$, where the parties Sen and Rec interact with the functionality $\mathscr{F}_{\mathrm{PSI}}$ in the ideal world, and corrupted Sen is controlled by the simulator $\mathscr{S}$. □

7.1. Adversary. The adversary $\mathscr{A}$ instructs Sen to run the protocol faithfully except for the following steps.

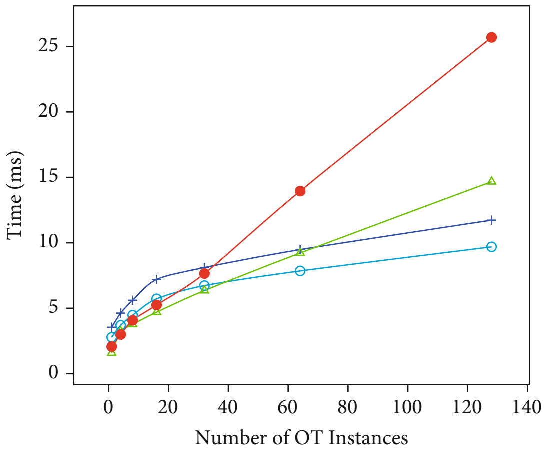


Figure 22: Running OT protocols on LAN.

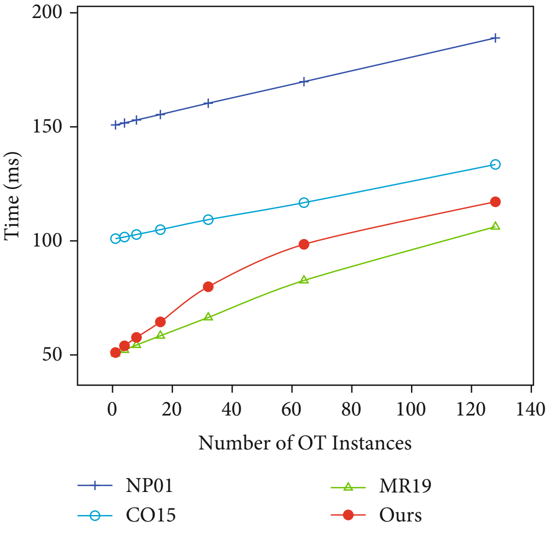


Figure 23: Running OT protocols on WAN.

For $i \in [w]$, upon receiving (SendNotify, sid) from $\mathscr{F}^{1,2}_{E\text{-}OT}$, the adversary $\mathscr{A}$ sends (Receive, sid, 0) to $\mathscr{F}^{1,2}_{E\text{-}OT}$ on behalf of Sen.

Table 2: Running time of OT protocols in milliseconds.

		Num									
Protocol	Rounds	1	8	32	64	128	1	8	32	64	128
		LAN					WAN				
[10]	3	3.56	5.61	8.11	9.48	11.73	150.8	153.0	160.4	169.8	189.0
[11]	2	2.78	4.47	6.73	7.85	9.69	101.0	102.8	109.3	116.8	133.5
[12]	1	1.58	3.77	6.35	9.22	14.67	50.6	54.3	66.4	82.6	106.2
Ours	1	2.07	4.08	7.65	13.95	25.70	51.1	57.7	79.9	98.5	117.1

Table 3: Running times in milliseconds of the semihonest OT extension protocol with different base OT protocols.

	Num							
Protocol	10^4	10^5	10^6	10^7	10^4	10^5	10^6	10^7
	LAN				WAN			
[10]	19	48	220	1550	249	421	1847	15543
[11]	17	45	217	1546	242	414	1842	15439
[12]	23	52	228	1553	214	389	1813	15332
Ours	34	63	238	156	227	404	1826	15410

For $i \in [w]$, upon receiving (ReceiveNotify, sid) from $\mathscr{F}^{1,2}_{E\text{–OT}}$, $\mathscr{A}$ sends (FixMessage, sid, 0) to $\mathscr{F}^{1,2}_{E\text{–OT}}$.

$\mathscr{A}$ makes poly(λ) random queries to hash^1 after receiving $\{\Delta_i\}_{i\in[w]}, k$ from Rec on behalf of Sen.

The environment $\mathscr{Z}$ outputs 1 if Rec sends (Compute, sid, Res) back where $\text{Res} = X \cap Y$, and it outputs 0 otherwise.

This can be seen as a drawback of the endemic OT functionality in that its application scenarios are limited, and sometimes, we need other types of OT functionalities. As depicted in Figure 21, we can adopt the transformation of [53] and obtain $\mathscr{F}^{1,N}_{S\text{–OT}}$ at the cost of one more round communication. After transformation, our endemic OT protocol can be used to construct this highly efficient PSI protocol Π_{PSI} even in the malicious setting. We can also obtain $\mathscr{F}_{U\text{–OT}}$ and $\mathscr{F}_{R\text{–OT}}$ following the protocols in [12].

8. Implementation and Benchmarks

In our protocols, we instantiate all the hash functions involved with SM3 hash function $\text{SM3} : \{0,1\}^* \longrightarrow \{0,1\}^\ell$. When the required output length is not to ℓ, we adopt a similar technique as used in the construction of the key derivation function (cf. Section 4.1). The pseudorandom function PRF and pseudorandom number generator function PRG can be instantiated with the SM4 block cipher algorithm $\text{SM4} : \{0,1\}^\lambda \times \{0,1\}^\ell \longrightarrow \{0,1\}^\ell$. Roughly speaking, to implement $\text{PRF} : \{0,1\}^\lambda \times \{0,1\}^\ell \longrightarrow \{0,1\}^\ell$, we use $\text{PRF}_k(m) = \text{SM4}_k(m)$. To implement $\text{PRG} : \{0,1\}^\lambda \longrightarrow \{0,1\}^m$, where $m = n \cdot \ell$, we use $\text{PRG}(k) = \text{SM4}_k(0) \| \cdots \| \text{SM4}_k(n-1)$. When $\ell \nmid m$, we truncate the extra bits as in Section 4.1. As for other mentioned protocols, we instantiate the hash function, PRF and PRG functions as described in their works, e.g., SHA256, for the hash function, and AES for PRF.

8.1. Experimental Setup. We perform the experiments on Dell OptiPlex 7080 equipped with an Intel Core 8700 CPU @ 3.20 GHz with 32.0 GB RAM, running Ubuntu 18.04 LTS. We evaluate all protocols in two simulated network settings: (i) a LAN setting with 1 Gbps bandwidth and 1 ms delay and (ii) a WAN setting with 100 Mbps bandwidth and 50 ms delay. All test results are the average of 10 tests.

8.2. Oblivious Transfer Evaluation. We first compare the performance of our multi-instance OT protocol $\Pi^{\text{num}}_{E\text{–OT}}$ with several state-of-the-art OT protocols [10, 11] and [12]. Note that the OT protocol in [10] is a sender OT protocol, and it needs an additional round to transfer messages. While [11, 12] and our protocol only generate random correlated messages. Besides, our protocol is based on the SM series cryptography, especially the SM2 key agreement protocol, while the other three protocols are inspired by the Diffie-Hellman key agreement protocol [54].

In Figure 22, we show the running time of the protocols in the LAN setting. Since the SM2 key agreement protocol needs more exponentiation operations than the Diffie-Hellman key agreement protocol, our protocol will be slower than the other protocols when the number of OT instances is large. However, as is shown in Figure 23, in the WAN setting, our protocol is faster than the protocol of Naor and Pinkas [10] as well as the protocol of Chou and Orlandi [11] because our protocol only needs one round as the protocol of Masny and Rindal [12]. Therefore, our protocol is specifically suitable for bad network environments, e.g., the wireless network. We also provide the detailed running time in Table 2. In the LAN setting, [11] is the fastest protocol for a large number of OT instances since it requires the least number of exponentiation operations. And in the WAN setting, [12] is the fastest protocol because it only needs one round of communication, and our protocol is slightly slower

Table 4: Running times in milliseconds of the triple generation protocol with different OT protocols.

	Num							
Protocol	10^4	10^5	10^6	10^7	10^4	10^5	10^6	10^7
	LAN				WAN			
[10]	39	97	452	3203	515	903	4106	32072
[11]	35	91	446	3193	496	884	4098	31723
[12]	46	106	465	3210	436	843	4048	31658
Ours	68	128	490	3238	464	871	4086	31814

than [12]. We note that our protocol is the only one that is based on the SM series cryptography and can be legally used for commercial purposes in China.

In real-life applications, OT extension techniques are often used to generate hundreds of thousands of OT instances with high speed. In such cases, the performance of the base OT protocols only has a minor impact on the performance of the overall protocol. To illustrate this, we provide the test result in Table 3. We use the OT protocols of [10–12] and our multi-instance OT protocol as the base OT for the semihonest setting OT extension. As the number of OT instances increases, the running time of different protocols is relatively closer. In the LAN setting, our protocol only takes 1.563 seconds to generate 10 million OT instances. And in the WAN setting, our protocol can generate the same number of OT instances in 15.41 seconds. Therefore, our protocol is comparable with other OT protocols in many application scenarios.

8.3. Triple-Generation Evaluation. One of the applications of our OT protocol is to generate the Beaver triples, which can be used in many MPC applications. Our Beaver triple-generation protocol Π_{triple} invokes the endemic OT functionality. We use different OT protocols as the base OT protocols for the OT extension protocol, and we use the OT extension protocols for the triple generation protocols. The test results can be found in Table 4, and as one can see, the performance of the triple-generation protocol mainly depends on its underlying OT extension protocols. In the LAN setting, our protocol needs 3.238 seconds to generate 10 million Beaver triples. And in the WAN setting, our protocol can generate the same number of Beaver triples in 31.814 seconds.

9. Conclusion

In this work, we investigate the problem of secure computation from the SM series cryptography, which complies with the Chinese cryptographic laws and is authorized for commercial usages in China. We show how to generate OT using the SM2 and SM3 algorithms. Moreover, we instantiate the OT extension protocols in the semihonest setting and malicious setting with the SM3 and SM4 algorithms, which can efficiently extend some base OTs to a polynomial number of OTs. With the generated OT, we can securely realize the Beaver multiplication triple-generation functionality and further construct generic MPC protocols. Besides, we show that the specific MPC, PSI, can also be implemented using the SM2, SM3, and SM4 algorithms. The proposed protocols are secure in the random oracle model and the public key infrastructure setting. The evaluation results indicate that our constructions are comparable to existing protocols and especially suitable for the wireless network environment. Therefore, we provide an efficient secure computation solution from SM series cryptography, and it is the first solution that can be used for commercial purposes in China.

Appendix

A. Functionalites

A.1. Random Oblivious Transfer. As depicted in Figure 9, the 1-out-of-N endemic OT functionality $\mathscr{F}^{1,N}_{U-\mathrm{OT}}$ waits for $(\mathrm{Send}, \mathrm{sid}, \mathrm{ssid})$ from Sen and $(\mathrm{Receive}, \mathrm{sid}, \mathrm{ssid}, c)$ from Rec, where $c \in [N]$ denotes Rec's choices. After both messages are obtained, $\mathscr{F}^{1,N}_{U-\mathrm{OT}}$ picks n uniformly random messages $\{m_i\}_{i\in[N]}$. At the end, $\mathscr{F}^{1,N}_{U-\mathrm{OT}}$ sends $\{m_i\}_{i\in[N]}$ to Sen and m_c to Rec.

Figure 13 depicts the multi-instance version of 1-out-of-2 $\mathscr{F}_{U-\mathrm{OT}}$.

B. Proof of Theorems

B.1. Proof of Theorem 13

Proof. To prove Theorem 13, we construct a simulator $\mathscr{S}$ such that for any nonuniform PPT environment $\mathscr{Z}$, the following ensembles are indistinguishable: (i) the real execution $\mathrm{exec}_{\Pi^{1,N}_{E-\mathrm{OT}},\mathscr{A},\mathscr{Z}}$, where the parties $\mathscr{P} := \{P_1, P_2\}$ run protocol $\Pi^{1,N}_{E-\mathrm{OT}}$ and the corrupted party is controlled by a dummy adversary $\mathscr{A}$ who simply forwards messages from/to $\mathscr{Z}$, and (ii) the ideal execution $\mathrm{exec}_{\mathscr{F}^{1,N}_{U-\mathrm{OT}},\mathscr{S},\mathscr{Z}}$, where the parties P_1 and P_2 interact with functionality $\mathscr{F}^{1,N}_{U-\mathrm{OT}}$ in the ideal world and the corrupted party is controlled by the simulator $\mathscr{S}$. We consider following cases. □

Case 1. Sen is corrupted; Rec is honest.

Simulator. The simulator $\mathscr{S}$ internally runs $\mathscr{A}$, forwarding messages to/from the environment $\mathscr{Z}$. $\mathscr{S}$ simulates the interface of $\mathscr{F}^{1,N}_{U-\mathrm{OT}}$ as well as honest Rec. In addition, the simulator $\mathscr{S}$ simulates the following interactions with $\mathscr{A}$:

(i) Upon receiving $(\mathrm{ReceiveNotify}, \mathrm{sid}, \mathrm{ssid})$ from the external functionality $\mathscr{F}^{1,N}_{U-\mathrm{OT}}$ and receiving $(\mathrm{Send}, \mathrm{sid}, \mathrm{ssid}, c)$ from the environment $\mathscr{Z}$ for Sen, the simulator $\mathscr{S}$ sends $(\mathrm{Send}, \mathrm{sid}, \mathrm{ssid})$ to $\mathscr{F}^{1,N}_{U-\mathrm{OT}}$, it then receives $(\text{SendNotify}, \text{sid}, \text{ssid})$ and $(\text{Send}, \text{sid}, \text{ssid}, \{k_i\}_{i\in[N]})$ from $\mathscr{F}_{U\text{–OT}}^{1,N}$. For $i \in [N]$, $\mathscr{S}$ picks random $r_i \longleftarrow \mathbb{G}$, and it sends $\{r_i\}_{i\in[N]}$ to Sen

(ii) For $i \in [N]$, when Sen queries the KDF for the i-th time, $\mathscr{S}$ returns k_i

Indistinguishability. The indistinguishability is proven through a series of hybrid worlds $\mathscr{H}_0, \cdots, \mathscr{H}_3$.

Hybrid $\mathscr{H}_0$: it is the real protocol execution.

Hybrid $\mathscr{H}_1$: $\mathscr{H}_1$ is the same as $\mathscr{H}_0$ except that in $\mathscr{H}_1$, the simulator $\mathscr{S}$ receives $\{k_i\}_{i\in[N]}$ for corrupted Sen from $\mathscr{F}_{U\text{–OT}}^{1,N}$. The view of Sen is not changed since Rec behaves exactly the same.

Hybrid $\mathscr{H}_2$: $\mathscr{H}_2$ is the same as $\mathscr{H}_1$ except that in $\mathscr{H}_2$, Rec picks random $r_i \longleftarrow \mathbb{G}$, for $i \in [N]$. In $\mathscr{H}_1$, $r_c = m_A - \text{hash}_c^{\mathbb{G}}(\{r_i\}_{i\neq c})$, where $\text{hash}_c^{\mathbb{G}}(\{r_i\}_{i\neq c}$ should be indistinguishable from a random element and serve as a one-time pad (OTP); therefore, r_c in $\mathscr{H}_1$ and $\mathscr{H}_2$ should be indistinguishable.

Hybrid $\mathscr{H}_3$: $\mathscr{H}_3$ is the same as $\mathscr{H}_2$ except that in $\mathscr{H}_3$, the random oracle KDF returns k_i for the i-th query. Because of the key indistinguishability of the SM2 key agreement protocol, $\mathscr{H}_2$ and $\mathscr{H}_3$ should be indistinguishable.

The adversary's view of $\mathscr{H}_3$ is identical to the simulated view. Therefore, $\text{exec}_{\Pi_{E\text{–OT}}^{1,N},\mathscr{A},\mathscr{Z}}$ and $\text{exec}_{\mathscr{F}_{U\text{–OT}}^{1,N},\mathscr{S},\mathscr{Z}}$ are indistinguishable.

Case 2. Rec is corrupted; Sen is honest.

Simulator. The simulator $\mathscr{S}$ internally runs $\mathscr{A}$, forwarding messages to/from the environment $\mathscr{Z}$. $\mathscr{S}$ simulates the interface of $\mathscr{F}_{U\text{–OT}}^{1,N}$ as well as honest Sen. In addition, the simulator $\mathscr{S}$ simulates the following interactions with $\mathscr{A}$:

(i) Upon receiving $(\text{SendNotify}, \text{sid}, \text{ssid})$ from the external functionality $\mathscr{F}_{U\text{–OT}}^{1,N}$ and receiving $(\text{Receive}, \text{sid}, \text{ssid}, c)$ from the environment $\mathscr{Z}$ for Rec, the simulator $\mathscr{S}$ sends $(\text{Receive}, \text{sid}, \text{ssid}, c)$ to $\mathscr{F}_{\text{U–OT}}^{1,N}$, and it then receives $(\text{ReceiveNotify}, \text{sid}, \text{ssid})$ and $(\text{Receive}, \text{sid}, \text{ssid}, k_c)$ from $\mathscr{F}_{U\text{–OT}}^{1,N}$. $\mathscr{S}$ invokes $(m_B, t_B) \longleftarrow \text{MsgGen}()$, and it sends m_B to Rec

(ii) When Rec queries the KDF, $\mathscr{S}$ returns k_c

Indistinguishability. The indistinguishability is proven through a series of hybrid worlds $\mathscr{H}_0, \cdots, \mathscr{H}_2$.

Hybrid $\mathscr{H}_0$: it is the real protocol execution.

Hybrid $\mathscr{H}_1$: $\mathscr{H}_1$ is the same as $\mathscr{H}_0$ except that in $\mathscr{H}_1$, the simulator $\mathscr{S}$ receives m_c for corrupted Rec from $\mathscr{F}_{U\text{–OT}}^{1,N}$. The view of Rec is not changed since Sen behaves exactly the same.

Hybrid $\mathscr{H}_2$: $\mathscr{H}_2$ is the same as $\mathscr{H}_1$ except that in $\mathscr{H}_2$, the random oracle KDF returns k_c when Rec queries it. Because of the key indistinguishability of the SM2 key agreement protocol, $\mathscr{H}_1$ and $\mathscr{H}_2$ should be indistinguishable.

The adversary's view of $\mathscr{H}_2$ is identical to the simulated view. Therefore, $\text{exec}_{\Pi_{E\text{–OT}}^{1,N},\mathscr{A},\mathscr{Z}}$ and $\text{exec}_{\mathscr{F}_{U\text{–OT}}^{1,N},\mathscr{S},\mathscr{Z}}$ are indistinguishable.

Case 3. Both Sen and Rec are corrupted.

Simulator. The simulator $\mathscr{S}$ internally runs $\mathscr{A}$, forwarding messages to/from the environment $\mathscr{Z}$.

Indistinguishability. This is a trivial case, since both Sen and Rec are controlled by the adversary $\mathscr{A}$.

B.2. Proof of Corollary 14

Proof. The simulator used to proof Corollary 14 is exactly the same as the simulator used to proof Theorem 13. Although the functionality $\mathscr{F}_{E\text{–OT}}^{1,N}$ allows the simulator to fix the corrupted party's message, we never invoke the $(\text{FixMessage}, \cdots)$ instruction. □

B.3. Proof of Theorem 19

Proof. To prove Theorem 19, we construct a simulator $\mathscr{S}$ such that for any nonuniform PPT environment $\mathscr{Z}$, the following ensembles are indistinguishable: (i) the real execution $\text{exec}_{\Pi_{\text{triple}},\mathscr{A},\mathscr{Z}}^{\mathscr{F}_{E\text{–OT}}^{1,2}}$, where the parties $\mathscr{P} \coloneqq \{P_1, P_2\}$ run protocol Π_{triple} in the $\mathscr{F}_{E\text{–OT}}^{1,2}$-hybrid model and the corrupted party is controlled by a dummy adversary $\mathscr{A}$ who simply forwards messages from/to $\mathscr{Z}$, and (ii) the ideal execution $\text{exec}_{\mathscr{F}_{\text{triple}},\mathscr{S},\mathscr{Z}}$, where the parties P_1 and P_2 interact with functionality $\mathscr{F}_{\text{triple}}$ in the ideal world and the corrupted party is controlled by the simulator $\mathscr{S}$. Since the protocol is symmetric, we only consider the case where P_1 is corrupted. □

Simulator. The simulator $\mathscr{S}$ internally runs $\mathscr{A}$, forwarding messages to/from the environment $\mathscr{Z}$. $\mathscr{S}$ simulates the interface of $\mathscr{F}_{E\text{–OT}}^{1,2}$ as well as honest P_2. In addition, the simulator $\mathscr{S}$ simulates the following interactions with $\mathscr{A}$:

(i) Upon receiving $(\text{GenerateNotify}, \text{sid}, \text{ssid}, P_2)$ from the external $\mathscr{F}_{\text{triple}}$, the simulator $\mathscr{S}$ sends $(\text{SendNotify}, \text{sid}, \text{ssid}_2)$ and $(\text{ReceiveNotify}, \text{sid}, \text{ssid}_1)$ to the adversary $\mathscr{A}$ on behalf of $\mathscr{F}_{E\text{–OT}}^{1,2}$. $\mathscr{S}$ also sends $(\text{Generate}, \text{sid}, P_1)$ to $\mathscr{F}_{\text{triple}}$ and receives $(\text{GenerateNotify}, \text{sid}, P_1)$

(ii) Upon receiving $(\text{Send}, \text{sid}, \text{ssid}_1)$ from P_1 via the interface of $\mathscr{F}_{E\text{–OT}}^{1,2}$, $\mathscr{S}$ sends $(\text{SendNotify}, \text{sid}, \text{ssid}_1)$ to $\mathscr{A}$ on behalf of $\mathscr{F}_{E\text{–OT}}^{1,2}$. Upon receiving $(\text{FixMessage}, \text{sid}, \text{ssid}_1, (m_0^1, m_1^1))$ from $\mathscr{A}$ via the interface of $\mathscr{F}_{E\text{–OT}}^{1,2}$, $\mathscr{S}$ sets $a_1 = m_0^1 \oplus m_1^1$, and it sends $(\text{Send}, \text{sid}, (m_0^1, m_1^1))$ to P_1 on behalf of $\mathscr{F}_{E\text{–OT}}^{1,2}$

(iii) Upon receiving $(\text{Receive}, \text{sid}, \text{ssid}_2, b_1)$ from P_1 via the interface of $\mathscr{F}_{E\text{–OT}}^{1,2}$, $\mathscr{S}$ sends $(\text{ReceiveNotify}, \text{sid}, \text{ssid}_2)$ to $\mathscr{A}$ on behalf of $\mathscr{F}_{E\text{–OT}}^{1,2}$. Upon receiving $(\text{FixMessage}, \text{sid}, \text{ssid}_2, m_{b_1}^2)$ from $\mathscr{A}$ via the interface of $\mathscr{F}^{1,2}_{E\text{-OT}}$, $\mathscr{S}$ sends $(\text{Receive}, \text{sid}, m^2_{b_1})$ to P_2 on behalf of $\mathscr{F}^{1,2}_{E\text{-OT}}$

(iv) $\mathscr{S}$ sets $c_1 := a_1 \wedge b_1 \oplus m^1_0 \oplus m^2_{b_1}$, and it sends $(\text{FixTriple}, \text{sid}, P_1, (a_1, b_1, c_1))$ to $\mathscr{F}_{\text{triple}}$. It outputs whatever $\mathscr{F}_{\text{triple}}$ outputs

Indistinguishability. The indistinguishability is proven through a series of hybrid worlds $\mathscr{H}_0, \cdots, \mathscr{H}_3$.

Hybrid $\mathscr{H}_0$: it is the real protocol execution.

Hybrid $\mathscr{H}_1$: $\mathscr{H}_1$ is the same as $\mathscr{H}_0$ except that in $\mathscr{H}_1$, the simulator $\mathscr{S}$ simulates the functionality $\mathscr{F}^{1,2}_{E\text{-OT}}$, and it receives b_1 from P_1 and $m^2_{b_1}, m^1_0, m^1_1$ from the adversary $\mathscr{A}$. The view of P_1 is not changed since $\mathscr{S}$ behaves exactly the same as $\mathscr{F}^{1,2}_{E\text{-OT}}$.

Hybrid $\mathscr{H}_2$: $\mathscr{H}_2$ is the same as $\mathscr{H}_1$ except that in $\mathscr{H}_2$, the simulator $\mathscr{S}$ sets $a_1 = m^1_0 \oplus m^1_1$ and $c_1 := a_1 \wedge b_1 \oplus m^1_0 \oplus m^2_{b_1}$, and it sends $(\text{FixTriple}, \text{sid}, P_1, (a_1, b_1, c_1))$ to the external $\mathscr{F}_{\text{triple}}$ to modify the triple. The view of P_1 is not changed since no message sent to P_1 is changed.

Hybrid $\mathscr{H}_3$: $\mathscr{H}_3$ is the same as $\mathscr{H}_2$ except that in $\mathscr{H}_3$, the output of P_2 is directly from $\mathscr{F}_{\text{triple}}$. The output distribution remains the same since (1) the simulator modifies (a_1, b_1, c_1) in $\mathscr{H}_2$ to the values obtained by P_1; (2) in both $\mathscr{H}_2$ and $\mathscr{H}_3$, b_2 is randomly picked; (3) in $\mathscr{H}_2$, $a_2 = m^2_0 \oplus m^2_1$, where one of the values is randomly picked, and in $\mathscr{H}_3$, a_2 is randomly picked; and (4) in both $\mathscr{H}_2$ and $\mathscr{H}_3$, it holds that $(a_1 \oplus a_2) \wedge (b_1 \oplus b_2) = c_1 \oplus c_2$.

The adversary's view of $\mathscr{H}_3$ is identical to the simulated view $\text{exec}_{\mathscr{F}_{\text{triple}}, \mathscr{S}, \mathscr{Z}}$. Therefore, it is perfectly indistinguishable.

B.4. Proof of Theorem 21

Proof. To prove Theorem 21, we construct a simulator $\mathscr{S}$ such that for any nonuniform PPT environment $\mathscr{Z}$, the following ensembles are indistinguishable: (i) the real execution $\text{exec}^{\mathscr{F}^{1,2}_{E\text{-OT}}}_{\Pi^{N,\text{length}}_{\text{triple}}, \mathscr{A}, \mathscr{Z}}$, where the parties $\mathscr{P} := \{P_1, P_2\}$ run protocol $\Pi^{N,\text{length}}_{\text{triple}}$ in the $\mathscr{F}^{1,2}_{E\text{-OT}}$-hybrid model and the corrupted party is controlled by a dummy adversary $\mathscr{A}$ who simply forwards messages from/to $\mathscr{Z}$, and (ii) the ideal execution $\text{exec}_{\mathscr{F}_{\text{triple}}, \mathscr{S}, \mathscr{Z}}$, where the parties P_1 and P_2 interact with functionality $\mathscr{F}_{\text{triple}}$ in the ideal world and the corrupted party is controlled by the simulator $\mathscr{S}$. We consider the extreme case where only P_1 is not corrupted. □

Simulator. The simulator $\mathscr{S}$ internally runs $\mathscr{A}$, forwarding messages to/from the environment $\mathscr{Z}$. $\mathscr{S}$ simulates the interface of $\mathscr{F}^{1,2}_{E\text{-OT}}$ as well as honest P_1. In addition, the simulator $\mathscr{S}$ simulates the following interactions with $\mathscr{A}$:

(i) Upon receiving $(\text{GenerateNotify}, \text{sid}, \text{ssid}, P_1)$ from the external $\mathscr{F}_{\text{triple}}$, the simulator $\mathscr{S}$ sends $(\text{SendNotify}, \text{sid}, \text{ssid}_{1,j,k})$ and $(\text{ReceiveNotify}, \text{sid}, \text{ssid}_{j,1,k})$ to the adversary $\mathscr{A}$ on behalf of $\mathscr{F}^{1,2}_{E\text{-OT}}$, for $j \neq 1$ and $k \in [\text{length}]$. $\mathscr{S}$ also sends $(\text{Generate}, \text{sid}, P_j)$ to $\mathscr{F}_{\text{triple}}$ and receives $(\text{GenerateNotify}, \text{sid}, P_j)$, for $j \neq 1$

(ii) For $j \neq 1, k \in [\text{length}], j' \in [N] \setminus \{1, j\}$:

(a) Upon receiving $(\text{Send}, \text{sid}, \text{ssid}_{j,1,k})$ from P_j via the interface of $\mathscr{F}^{1,2}_{E\text{-OT}}$, $\mathscr{S}$ sends $(\text{SendNotify}, \text{sid}, \text{ssid}_{j,1,k})$ to $\mathscr{A}$ on behalf of $\mathscr{F}^{1,2}_{E\text{-OT}}$. Upon receiving $(\text{FixMessage}, \text{sid}, \text{ssid}_{j,1,k}, (m^{j,1,k}_0, m^{j,1,k}_1))$ from $\mathscr{A}$ via the interface of $\mathscr{F}^{1,2}_{E\text{-OT}}$, $\mathscr{S}$ sends $(\text{Send}, \text{sid}, \text{ssid}_{j,1,k}, (m^{j,1,k}_0, m^{j,1,k}_1))$ to P_j on behalf of $\mathscr{F}^{1,2}_{E\text{-OT}}$

(b) Upon receiving $(\text{Receive}, \text{sid}, \text{ssid}_{1,j,k}, b_j[k])$ from P_j via the interface of $\mathscr{F}^{1,2}_{E\text{-OT}}$, $\mathscr{S}$ sends $(\text{ReceiveNotify}, \text{sid}, \text{ssid}_{1,j,k})$ to $\mathscr{A}$ on behalf of $\mathscr{F}^{1,2}_{E\text{-OT}}$. Upon receiving $(\text{FixMessage}, \text{sid}, \text{ssid}_{1,j,k}, m^{1,j,k}_{b_j[k]})$ from $\mathscr{A}$ via the interface of $\mathscr{F}^{1,2}_{E\text{-OT}}$, $\mathscr{S}$ sends $(\text{Receive}, \text{sid}, m^{1,j,k}_{b_j[k]})$ to P_2 on behalf of $\mathscr{F}^{1,2}_{E\text{-OT}}$

(c) Upon receiving $(\text{Send}, \text{sid}, \text{ssid}_{j,j',k})$ from P_j via the interface of $\mathscr{F}^{1,2}_{E\text{-OT}}$, $\mathscr{S}$ sends $(\text{SendNotify}, \text{sid}, \text{ssid}_{j,j',k})$ to $\mathscr{A}$ on behalf of $\mathscr{F}^{1,2}_{E\text{-OT}}$. Upon receiving $(\text{Receive}, \text{sid}, \text{ssid}_{j,j',k}, b_{j'}[k])$ from $P_{j'}$ via the interface of $\mathscr{F}^{1,2}_{E\text{-OT}}$, $\mathscr{S}$ sends $(\text{ReceiveNotify}, \text{sid}, \text{ssid}_{j,j',k})$ to $\mathscr{A}$ on behalf of $\mathscr{F}^{1,2}_{E\text{-OT}}$. Upon receiving $(\text{FixMessage}, \text{sid}, \text{ssid}_{j,j',k}, (m^{j,j',k}_0, m^{j,j',k}_1))$ from $\mathscr{A}$ via the interface of $\mathscr{F}^{1,2}_{E\text{-OT}}$, $\mathscr{S}$ sends $(\text{Send}, \text{sid}, \text{ssid}_{j,j',k}, (m^{j,j',k}_0, m^{j,j',k}_1))$ to P_j and $(\text{Receive}, \text{sid}, \text{ssid}_{j,j',k}, m^{j,j',k}_{b_{j'}[k]})$ to $P_{j'}$ on behalf of $\mathscr{F}^{1,2}_{E\text{-OT}}$

(iii) $\mathscr{S}$ picks random $r_{1,j,k} \longleftarrow \{0,1\}^{\text{length}}$, for $j \neq 1, k \in [\text{length}]$. It then sends $\{r_{1,j,k}\}_{k \in [\text{length}]}$ to P_j, for $j \neq 1$

(iv) For $j \neq 1$, upon receiving $\{r_{j,1,k}\}_{k \in [\text{length}]}$ from P_j for P_1, $\mathscr{S}$ computes $a_j = r_{j,1,1} - m^{j,1,1}_0 + m^{j,1,1}_1 \bmod 2^{\text{length}}$. After that, $\mathscr{S}$ computes $r_{j,j',k} := a_j + m^{j,j',k}_0 - m^{j,j',k}_1 \bmod 2^{\text{length}}$, for $j \neq 1, j' \in [N] \setminus \{1, j\}, k \in [\text{length}]$. Subsequently, $\mathscr{S}$ computes $s_{j',j,k} := m^{j',j,k}_{b_j[k]} + b_j[k] \cdot r_{j',j,k} \bmod 2^{\text{length}}$, for $j \neq 1, j' \neq j, k \in [\text{length}]$. At the end, $\mathscr{S}$ computes $c_j = a_j \cdot b_j + (\sum_{j' \neq j} \sum_k (s^{j',j,k} - m^{j,j',k}_0) \cdot 2^{k-1}) \bmod 2^{\text{length}}$, and it sends $(\mathit{FixTriple}, \text{sid}, P_j, (a_j, b_j, c_j))$ to $\mathscr{F}_{\text{triple}}$, for $j \neq 1$. It outputs whatever $\mathscr{F}_{\text{triple}}$ outputs

Indistinguishability. The indistinguishability is proven through a series of hybrid worlds $\mathscr{H}_0, \cdots, \mathscr{H}_4$.

Hybrid $\mathscr{H}_0$: it is the real protocol execution.

Hybrid $\mathscr{H}_1$: $\mathscr{H}_1$ is the same as $\mathscr{H}_0$ except that in $\mathscr{H}_1$, the simulator $\mathscr{S}$ simulates the functionality $\mathscr{F}^{1,2}_{E-\text{OT}}$ to extract $\{b_j\}_{j\neq 1}$ and obtains $\{m_0^{j,j',k}, m_1 j, j', k\}_{j\neq 1, j'\neq j, k\in[\text{length}]}$ and $\{m^{1,j,k}_{b_j[k]}\}_{j\neq 1, k\in[\text{length}]}$. The view of P_1 is not changed since $\mathscr{S}$ behaves exactly the same as $\mathscr{F}^{1,2}_{E-\text{OT}}$.

Hybrid $\mathscr{H}_2$: $\mathscr{H}_2$ is the same as $\mathscr{H}_1$ except that in $\mathscr{H}_2$, the simulator $\mathscr{S}$ computes $\{a_j, c_j\}_{j\neq 1}$ using the knowledge of $\{b_j\}_{j\neq 1}$, $\{m_0^{j,j',k}, m_1^{j,j',k}\}_{j\neq 1, j'\neq j, k\in[\text{length}]}$, $\{m^{1,j,k}_{b_j[k]}\}_{j\neq 1, k\in[\text{length}]}$, and $\{r_{1,j,k}, r_{j,1,k}\}_{j\neq 1, k\in[\text{length}]}$. It then sends (FixTriple, sid, P_j, (a_j, b_j, c_j)) to $\mathscr{F}_{\text{triple}}$ to the external $\mathscr{F}_{\text{triple}}$ to modify the triple, for $j\neq 1$. The view of P_1 is not changed since no message sent to P_1 is changed.

Hybrid $\mathscr{H}_3$: $\mathscr{H}_3$ is the same as $\mathscr{H}_2$ except that in $\mathscr{H}_3$, the simulator $\mathscr{S}$ picks random $r_{1,j,k} \longleftarrow \{0,1\}^{\text{length}}$, for $j\neq 1$, $k\in[\text{length}]$, instead of computing $r_{1,j,k} \coloneqq a_1 + m_0^{1,j,k} - m_1^{1,j,k} \bmod 2^{\text{length}}$. The views of the other parties in $\mathscr{H}_2$ and $\mathscr{H}_3$ have the same distribution since one of $m_0^{1,j,k}, m_1^{1,j,k}$ is uniformly random.

Hybrid $\mathscr{H}_4$: $\mathscr{H}_4$ is the same as $\mathscr{H}_3$ except that in $\mathscr{H}_4$, the output of P_1 is directly from $\mathscr{F}_{\text{triple}}$. The output distribution remains the same since (1) the simulator modifies (a_j, b_j, c_j) in $\mathscr{H}_3$ to the values obtained by P_j; (2) in both $\mathscr{H}_3$ and $\mathscr{H}_4$, a_1 is randomly picked; (3) in both $\mathscr{H}_3$ and $\mathscr{H}_4$, b_1 is randomly picked; and (4) in both $\mathscr{H}_3$ and $\mathscr{H}_4$, it holds that $(\sum_i a_i)\cdot(\sum_i b_i) \equiv \sum_i c_i$. The adversary's view of $\mathscr{H}_4$ is identical to the simulated view. Therefore, it is perfectly indistinguishable.

Data Availability

The data used in the submitted manuscript are available by email contacting the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Key R&D Program of China (No. 2021YFB3101601) and the National Natural Science Foundation of China (Grant No. 62072401 and No. 62232002). It is also supported by the "Open Project Program of Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province". This project is supported by Input Output (iohk.io).

References

[1] K. Gulati, R. S. K. Boddu, D. Kapila, S. L. Bangare, N. Chandnani, and G. Saravanan, *A Review Paper on Wireless Sensor Network Techniques in Internet of Things (IoT)*, vol. 51, Materials Today: Proceedings, 2022.

[2] M.-k. Choi, R. J. Robles, C.-h. Hong, and T.-h. Kim, "Wireless network security: vulnerabilities, threats and countermeasures," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 3, no. 3, pp. 77–86, 2008.

[3] A. Kavianpour and M. C. Anderson, "An overview of wireless network security," in *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 306–309, New York, NY, USA, June 2017.

[4] Z. Hu, L. Wang, L. Qi, Y. Li, and W. Yang, "A novel wireless network intrusion detection method based on adaptive synthetic sampling and an improved convolutional neural network," *IEEE Access*, vol. 8, pp. 195741–195751, 2020.

[5] P. Manickam, K. Shankar, E. Perumal, M. Ilayaraja, and K. S. Kumar, "Secure data transmission through reliable vehicles in vanet using optimal lightweight cryptography," in *Cybersecurity and secure information systems*, pp. 193–204, Springer, 2019.

[6] M. Šarac, N. Pavlović, N. Bacanin, F. al-Turjman, and S. Adamović, "Increasing privacy and security by integrating a blockchain secure interface into an iot device security gateway architecture," *Energy Reports*, vol. 7, pp. 8075–8082, 2021.

[7] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*, Cambridge University Press, 2015.

[8] A. C. Yao, "Protocols for secure computations," in *23rd annual symposium on foundations of computer science (sfcs 1982)*, pp. 160–164, Chicago, IL, USA, November 1982.

[9] J. Kilian, "Founding crytpography on oblivious transfer," in *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pp. 20–31, Chicago, Illinois, USA, 1988.

[10] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," in *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pp. 448–457, Washington, DC, USA, 2001.

[11] T. Chou and C. Orlandi, "The simplest protocol for oblivious transfer," in *Progress in Cryptology – LATINCRYPT 2015*, pp. 40–58, Springer, 2015.

[12] D. Masny and P. Rindal, "Endemic oblivious transfer," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 309–326, London, UK, November 2019.

[13] S. Garg, Y. Ishai, and A. Srinivasan, "Two-round mpc: information-theoretic and black-box," in *Theory of Cryptography. TCC 2018*, pp. 123–151, Springer, 2018.

[14] I. McQuoid, M. Rosulek, and L. Roy, "Minimal symmetric pake and 1-out-of-n ot from programmable-once public functions," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 425–442, October 2020.

[15] I. McQuoid, M. Rosulek, and L. Roy, "Batching base oblivious transfers," in *Advances in Cryptology – ASIACRYPT 2021. ASIACRYPT 2021*, pp. 281–310, Springer, 2021.

[16] D. Mahto and D. K. Yadav, "RSA and ECC: a comparative analysis," *International Journal of Applied Engineering Research*, vol. 12, no. 19, pp. 9053–9061, 2017.

[17] M. Bafandehkar, S. M. Yasin, R. Mahmod, and Z. M. Hanapi, "Comparison of ECC and RSA algorithm in resource constrained devices," in *2013 international conference on IT convergence and security (ICITCS)*, pp. 1–3, Macao, China, December 2013.

[18] Standing Committee of the National People's Congress, "Public key cryptographic algorithm SM2 based on elliptic curves,"

April 2022, https://www.oscca.gov.cn/sca/xxgk/2010-12/17/1002386/files/b791a9f908bb4803875ab6aeeb7b4e03.pdf.

[19] State Cryptography Administration of China, "Cryptography law of the People's Republic of China," April 2022, http://www.npc.gov.cn/npc/c30834/201910/6f7be7dd5ae5459a8de8baf36296bc74.shtml.

[20] J. Garay, Y. Ishai, R. Ostrovsky, and V. Zikas, "The price of low communication in secure multi-party computation," in *Advances in Cryptology – CRYPTO 2017. CRYPTO 2017*, pp. 420–446, Springer, 2017.

[21] C. Gentry, S. Halevi, H. Krawczyk et al., "YOSO: you only speak once," in *Advances in Cryptology – CRYPTO 2021*, pp. 64–93, Springer, 2021.

[22] G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs, "Multiparty computation with low communication, computation and interaction via threshold FHE," in *Advances in Cryptology – EUROCRYPT 2012*, pp. 483–501, Springer, 2012.

[23] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 1219–1234, New York, USA, May 2012.

[24] P. Mukherjee and D. Wichs, "Two round multiparty computation via multi-key FHE," in *Advances in Cryptology – EUROCRYPT 2016*, pp. 735–763, Springer, 2016.

[25] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Advances in Cryptology – CRYPTO 2012. CRYPTO 2012*, pp. 643–662, Springer, 2012.

[26] X. Wang, S. Ranellucci, and J. Katz, "Authenticated garbling and efficient maliciously secure two-party computation," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pp. 21–37, Dallas, Texas, USA, October 2017.

[27] X. Wang, S. Ranellucci, and J. Katz, "Global-scale secure multiparty computation," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 39–56, Dallas, Texas, USA, October 2017.

[28] H. Carter, B. Mood, P. Traynor, and K. R. B. Butler, "Secure outsourced garbled circuit evaluation for mobile devices," in *USENIX Security*, pp. 289–304, USENIX Association, 2013.

[29] D. Demmler, T. Schneider, and M. Zohner, "Ad-hoc secure two-party computation on mobile devices using hardware tokens," in *23rd USENIX Security Symposium (USENIX Security 14)*, pp. 893–908, San Diego, CA, USA, 2014.

[30] S. Felsen, Á. Kiss, T. Schneider, and C. Weinert, "Secure and private function evaluation with intel SGX," in *Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pp. 165–181, London, UK, November 2019.

[31] Y. Lu, B. Zhang, H.-S. Zhou, W. Liu, L. Zhang, and K. Ren, "Correlated randomness teleportation via semi-trusted hardware—enabling silent multi-party computation," in *Computer Security – ESORICS 2021*, pp. 699–720, Springer, 2021.

[32] R. Anderson and M. Kuhn, "Low cost attacks on tamper resistant devices," in *Security Protocols*, pp. 125–136, Springer, 1997.

[33] P. Kocher, J. Horn, A. Fogh et al., "Spectre attacks: exploiting speculative execution," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 1–19, San Francisco, CA, USA, May 2019.

[34] R. Canetti, "Universally composable security: a new paradigm for cryptographic protocols," in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pp. 136–145, Newport Beach, CA, USA, October 2001.

[35] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security - CCS '93*, pp. 62–73, Fairfax, Virginia, USA, December 1993.

[36] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Advances in Cryptology — EUROCRYPT 2002*, pp. 337–351, Springer, 2002.

[37] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proceedings of the nineteenth annual ACM conference on Theory of computing - STOC '87*, New York, NY, USA, January 1987.

[38] A. C.-C. Yao, "How to generate and exchange secrets," in *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pp. 162–167, Toronto, ON, Canada, October 1986.

[39] A. Yang, J. Nam, M. Kim, and K.-K. R. Choo, "Provably-secure (Chinese government) SM2 and simplified SM2 key exchange protocols," *The Scientific World Journal*, vol. 2014, Article ID 825984, 8 pages, 2014.

[40] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology — CRYPTO' 93*, pp. 232–249, Springer, 1993.

[41] M. Bellare and P. Rogaway, "Provably secure session key distribution: the three party case," in *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing - STOC '95*, pp. 57–66, Las Vegas, Nevada, USA, May 1995.

[42] D. Beaver, "Correlated pseudorandomness and the complexity of private computations," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, pp. 479–488, Philadelphia, Pennsylvania, USA, July 1996.

[43] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner, "More efficient oblivious transfer extensions with security for malicious adversaries," in *Advances in Cryptology – EUROCRYPT 2015*, pp. 673–701, Springer, 2015.

[44] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank, "Extending oblivious transfers efficiently," in *Advances in Cryptology - CRYPTO 2003*, pp. 145–161, Springer, 2003.

[45] G. Asharov, Y. Lindell, T. Schneider, and M. Zohner, "More efficient oblivious transfer and extensions for faster secure computation," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security - CCS '13*, pp. 535–548, Berlin, Germany, November 2013.

[46] M. Keller, E. Orsini, and P. Scholl, "Actively secure ot extension with optimal overhead," in *Advances in Cryptology – CRYPTO 2015. CRYPTO 2015*, pp. 724–741, Springer, 2015.

[47] J. Doerner, Y. Kondi, E. Lee, and A. Shelat, "Secure two-party threshold ECDSA from ECDSA assumptions," in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 980–997, San Francisco, CA, USA, May 2018.

[48] V. Kolesnikov and R. Kumaresan, "Improved ot extension for transferring short secrets," in *Advances in Cryptology – CRYPTO 2013*, pp. 54–70, Springer, 2013.

[49] M. Orrù, E. Orsini, and P. Scholl, "Actively secure 1-out-of-n ot extension with application to private set intersection," in *Topics in Cryptology – CT-RSA 2017*, pp. 381–396, Springer, 2017.

[50] A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," in *Advances in Cryptology — CRYPTO' 86*, pp. 186–194, Springer, 1986.

[51] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Advances in Cryptology — CRYPTO '91*, pp. 420–432, Springer, 1991.

[52] M. Chase and P. Miao, "Private set intersection in the internet setting from lightweight oblivious PRF," in *Advances in Cryptology – CRYPTO 2020*, pp. 34–63, Springer, 2020.

[53] D. Beaver, "Precomputing oblivious transfer," in *Advances in Cryptology — CRYPT0' 95*, pp. 97–109, Springer, 1995.

[54] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 3976302, 14 pages
https://doi.org/10.1155/2023/3976302

Review Article

Big Data Analytics, Processing Models, Taxonomy of Tools, V's, and Challenges: State-of-Art Review and Future Implications

Sandeep Dasari and Rajesh Kaluri

School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, India

Correspondence should be addressed to Rajesh Kaluri; rajesh.kaluri@vit.ac.in

Received 6 June 2022; Revised 8 October 2022; Accepted 13 April 2023; Published 10 May 2023

Academic Editor: Zengpeng Li

In the current digital era, data is budding tremendously from various sources like banks, businesses, education, entertainment, etc. Due to its significant consequence, it became a prominent proceeding for numerous research areas like the semantic web, machine learning, computational intelligence, and data mining. For knowledge extraction, several corporate sectors depend on tweets, blogs, and social data to get adequate analysis. It helps them predict the customer's tastes and preferences, optimize the usage of resources. In some cases, the same data creates complications that lead to a problem named as big data. To solve this, so many researchers have given various solutions. Based on literature analysis formulated 6-s simulation towards big data, detailed information about characteristics, a taxonomy of tools, and discussed various processing paradigms. No one tool can truly fit for all solutions, so this paper helps to make decisions smoothly by providing enough information and discussing major privacy issues and future directions.

1. Introduction

Advancement in digital resources leads to the rapid evolution of massive collections of data, which include complex, sprouting data sets that are congregated from heterogeneous data sources. Such voluminous data is termed as big data. It was coined by a scientist named John R. Mashey in the 1990s. Every online activity creates some data, especially internet usage, which has become a basic need in COVID-19 pandemic time. Facebook receives more than 500 terabytes of data in terms of photos, videos, messages, comments, etc. [1]. On average each day, 2.5 quintillion bytes of data can be generated, and in today's world, 90% of the information was developed in the last two years [2]. This data is available in the form of structured, unstructured, and semistructured data types. Rapid evolution in data processing mechanisms makes big data to become a hotfoot in all science and engineering domains, including physical, biological, and biomedical sciences [3].

Before explorations in data analysis technologies, organizations could not be able to handle their archives efficiently by applying traditional techniques [4]. Commodity systems are restricted with storage, invariable tool management, performance, scalability, and flexibility [5]. In the case of privacy concerns, data will also get an effect. Especially that the privacy of data is very important in the case of maintaining sensitive credentials like bank details, legal information, medical records, biometric details of officials, etc. By adding machine-learning features, the scope of safeguarding the data can be increased [6]. It can be perfected by applying a concept called privacy preservation machine learning (PPML) [7]. Its processing mechanism is shown in Figure 1. Majorly, it includes differential privacy, homomorphic encryption, multiparty computation, federated learning, and ensemble techniques [8]. Each approach plays an effective role in solving various privacy concerns and reduces the vulnerability intensity exposed to an attacker. In the online era, preserving the privacy of end users on social media has been regarded as a bottleneck issue.

Ironically, as the data analytics introduced in this paper become more progressive, the risk of privacy is also growing. As such, many privacy-preserving solutions have been

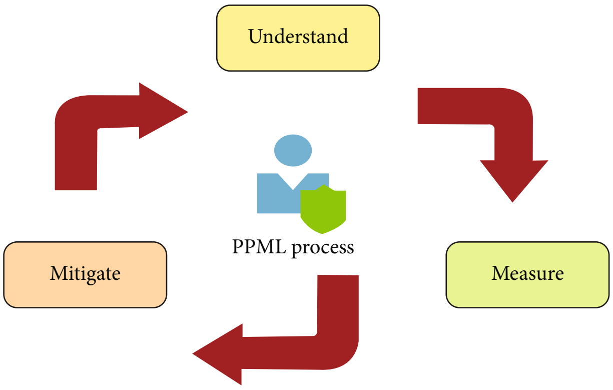


Figure 1: PPML process.

proposed to address these issues. There are two main approaches. The first one is k-anonymity, which is a property implemented by certain unknown data. Given the private data and a set of selected fields, the system has to make the practically useful data without finding the individuals who are the subjects of the data. The second one is differential privacy, which can provide an effective way to improve the quality of queries from statistical databases by reducing the chances of finding its records. The evolution of big data can be considered in 3 phases, as shown in Figure 2, each with its own set of features and capabilities.

Although big data has been accepted by so many organizations, research on big data in the cloud is in its early stages. Several concerns have not been fully addressed. Moreover, new challenging issues continue to emerge from various applications by organizations. Some of the prominent research challenges are as follows [9].

1.1. Data Integrity. An important feature of big data is to obtain integrity in security. Integrity means that data will be updated only through authorized persons. The expansion of cloud services provides users with a repository and mastery of their data in the data centers of the cloud. Such applications must check the data integrity, and another main challenge is to make sure about the rightness of the users' data. Users may not be physically able to get the data, but the cloud should provide flexibility for the user to scrutinize how the data can be maintained.

1.2. Data Transformation. Converting the data into a form that can be worthy of analysis is a major issue in big data. Due to the various data formats, big data can be transformed into an analysis workflow in two types. If the data is structured, it can be preprocessed before it is stored in relational databases. Later, the data can be retrieved to perform analysis. In case of unstructured data, it must be stored in distributed databases, such as HBase, before it is processed for analysis.

1.3. Data Quality. In the earlier days, data processing could be done on clean datasets from known and limited sources. Therefore, the outcomes are in an acceptable manner. However, with the inception of big data, data can be produced from various sources; all these resources are not well-known or verifiable. Fewer data quality has become a significant issue for so many service providers because data are frequently collected from huge sources [9]. For instance, large volumes of data are produced from smartphones, where inconsistent data formats can be produced as a resultant of heterogeneous sources.

1.4. Data Governance. It involves leveraging information by aligning the objectives of multiple functions, such as telecommunication carriers having access to huge troves of user's information in various forms like call details and marketing seeking to monetize this information by selling it to third parties. Moreover, big data plays an active role in providing opportunities to service providers by making users information more valuable.

2. Role of Data Analytics

Data will not be helpful or valuable until it is appropriately integrated and understood. Organizations are utilizing big data advances to explore the results of continuous information-driven choices. Various data analytics processing mechanisms help to stay ahead of business challenges by improving new designs and products, customer personalization, successful marketing, the generation of new profit options, and improved processing efficiency [10].

Real-time scenarios that spectacle the big data importance are as follows:

(i) By 2025, the big data analytics industry is likely to be generated more than $103 billion

(ii) The US economy loses up to $3.1 trillion per year due to poor data quality

(iii) Every individual in 2020 created 1.7 gigabytes in less than a second

(iv) Every day, internet users generate 2.5 quintillion bytes of data

(v) 97.2 percent of entrepreneurs are investing in artificial intelligence and big data

(vi) Netflix gets a profit of $1 billion each year on user retention thanks to big data

2.1. Working with Big Data Analytics. Data scientists and other statisticians collect data from different data sources. Data preprocessing can be performed on data such that it can be made accessible for processing systems to understand and analyze the massive volume of data. Valuable insights can be generated by applying machine learning algorithms [11]. The data analytics process includes four steps: data collection, cleaning, processing, and analysis [12]. The data necessary for analysis is based on a question or an experience. Depending on the demands that lead to analyzing the data to use as input to the study, specific population factors can be identified and information can be categorized [13].

Data collection is the practice of acquiring information on specified variables specified as data requirements. The main emphasis is on collecting data that is accurate and

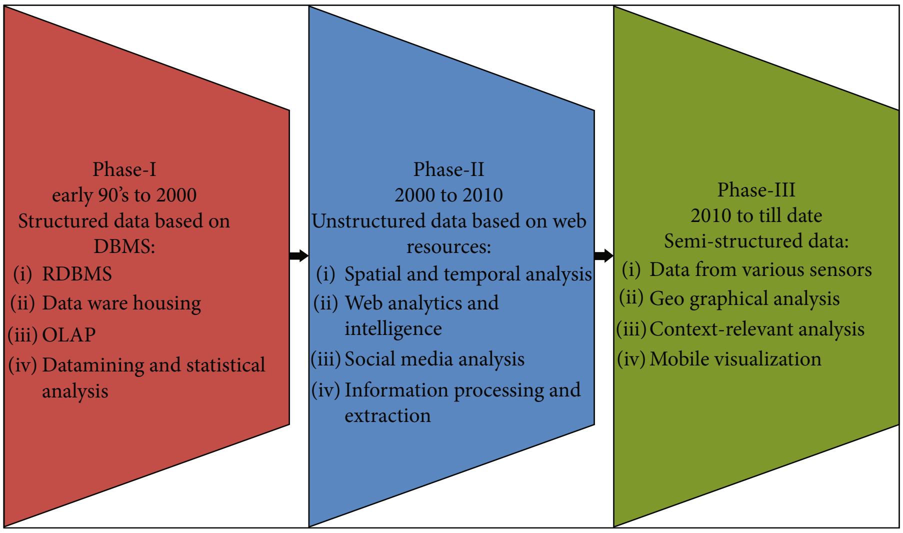

Phase-I
early 90's to 2000
Structured data based on DBMS:
(i) RDBMS
(ii) Data ware housing
(iii) OLAP
(iv) Datamining and statistical analysis
Phase-II
2000 to 2010
Unstructured data based on web resources:
(i) Spatial and temporal analysis
(ii) Web analytics and intelligence
(iii) Social media analysis
(iv) Information processing and extraction
Phase-III
2010 to till date
Semi-structured data:
(i) Data from various sensors
(ii) Geo graphical analysis
(iii) Context-relevant analysis
(iv) Mobile visualization

Figure 2: Evolution phases of big data technologies.

authentic. It does not guarantee that the data obtained is accurate. It gives a baseline against which to measure progress as well as a user's requirement [14]. Data is gathered from various sources, including organizational databases and web page information. The resulting data could be unstructured and contain irrelevant information, and it must undergo data processing and cleaning. For analysis, the obtained information must be transformed or structured, which includes reconstructing the data to match the needs of the various analysis tools [15].

The data that has been extracted and arranged may be duplicated, incomplete, or result in false reports. To avoid this problem, data cleaning can be induced. Depending on the nature of the data, different forms are available. Specific results might be evaluated against reliable published amounts or established thresholds. While cleansing financial data, quantitative data methods can also be used to find outliers that are eliminated from further investigation [16]. After it had been processed, organized, and sanitized, the data would be ready for analysis. Various data analysis techniques are available to analyze, evaluate, and develop conclusions based on the criteria [17]. Data visualization is used to examine data graphically to obtain an effective analysis of the data. With the help of visual elements like line graphs, bar graphs, and area charts, these tools provide an acceptable way to see and understand patterns, outliers, and trends in data. Additionally, it is a better way for entrepreneurs to impart data to a nontechnical audience.

To find the relationships between statistical data models and variables, such as regression and correlation analysis can be utilized. These data-descriptive models aid in the simplification of evolution and communicational outcomes. The process may involve additional cleaning or data collection, and therefore, these activities are iterative. The following Figure 3 represents various analytic use cases involved in data.

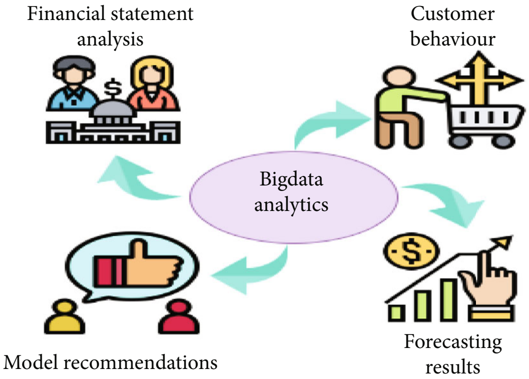


Figure 3: Different use cases of big data analytics.

2.1.1. Descriptive Analysis. It is a statistical approach that can be used to summarize historical data and find patterns and facilitate in the preparation of reports including financial statements, revenue, and marketing by summarizing data in an accessible way [15]. Data aggregation and data mining are two approaches involved in this analysis to discover historical data. Data is first collected and arranged by data aggregation to make the datasets more viable for analysts. Data mining involves a search of the data to identify patterns and meaning. Identified patterns are analyzed to discover the specific ways that learners interacted with the learning content and within the learning environment. Quick and easy reports which show how performance can be achieved, identifying performance issues and gaps earlier—before they become serious complications.

2.1.2. Diagnostic Analysis. To point out what created the issue, data mining, drill-down, and data recovery are examples of techniques [18]. Businesses use this because they give a thorough understanding of the issue. For companies that collect customer data, diagnostic analysis is the key to understanding why customers select particular orders. These insights will help improve products and the user experience.

2.1.3. Predictive Analysis. This type of analytics looks at both previous and present data to make forecasts for the future. It examines existing data and forecasts future data mining, artificial intelligence, and machine learning trends [18, 19]. It speculates on customer and market trends; for online learning specifically, predictive analytics is often found incorporated. Personalizes the training needs of learners to find strengths, weaknesses, and gaps; required learning resources and training can be offered to improve individual needs.

2.1.4. Prescriptive Analysis. This type of analytics provides a solution to a specific issue. Both predictive and descriptive analysis can be used in perspective analytics [16]. AI and machine learning can be used for this type of analysis. It generates recommendations and makes decisions based on the computational findings of algorithmic models. Automated decisions or recommendations required specific, unique, and clear directions from those utilizing the analytical technique to analyze specific learners who required additional support, regardless of how many students or employees and identify0 outstanding learners to provide additional resources [20].

2.2. Big Data Can Influence the Decision-Making Process as Follows

2.2.1. Customer Service Metrics in Real Time. Providing services to the customer is one of the essential areas in which firms must deliver metrics nowadays. Firms exploit real-time data to provide customers with one-on-one customized services and solutions [21]. Big data helps provide users with personalized loyalty programs.

2.2.2. Improving Operational Efficiency. Currently, companies are trying to use data for automating operations, optimizing trading strategies, and getting overall corporate efficiency outcomes. For example, vehicles equipped with sensors collect data and transmit it to central servers for analysis [22]. Personal car owners are also notified about prior repairs or services, which resource the company in enhancing the performance of its automobiles [23].

2.2.3. Increased Efficiency without Investing More. Without investing more resources, one can envision the customers. Sprint, a telecommunications firm, employs big data analytics to analyze real-time information to assess network failures, maximize resources, and improve customer experience [24]. This can result in enhancing the brand's delivery rate.

Some of the remarkable benefits of using big data analytics are as follows:

(i) Data from various sources like Twitter and Facebook helps better customer retention, marketing strategy decisions, and business intelligence for better insights and predictions

(ii) Customer service issues can be resolved quickly with NLP (natural language processing) features to improve customer satisfaction, solve problems, etc.

(iii) Operational efficiency improves results and produces when large volumes of data are well-analyzed and used to update services, security issues, healthcare, IoT, etc. [11].

(iv) Errors can be identified in an early stage to reduce the risk to services

(v) Big data analytics lay a path for large data warehouses to integrate with multiple sources and advanced technologies and process the data

3. Related Work

Budding big data applications has got progressively significant over the past couple of years. Previously proposed systems help big data applications assist to store, investigate, and measure information. Adopting the right combination of technology is also vital to obtaining the best results [25]. Parallel processing became an arising multidisciplinary area of research that is progressively drawing the consideration of scientists and professionals from different areas, including technology, intellectual, and sociologies [4]. In this paper, the authors contributed their work on existing frameworks and challenges of big data with assessment evaluation which provides a comparative study of various accessible frameworks and workloads. Existing information handling innovations are not appropriate for adopting the extraordinary measures of produced information like big data sets and conventional information strategies; they show moderate responsiveness and lack of versatility, execution, and precision. So many data collection challenges remain to be addressed. It may lead to a compromise of information security to a great extent [26].

The combination of AI and information management approaches offers better outcomes. However, many explorations work center on big data. Hadoop, Spark, Storm, and Flink frameworks, Flume, Kafka, GFS, HDFS, Smaza, and spark streaming technologies effectively implement data [5]. Provided elaborate focus on case studies on mLib, Spark, Flink, and Mahout [6]. Deep learning helps to handle a vast volume of data, which is common in big data, and gives acceptable results with complex datasets [7, 27, 28]. Boundless numbers of challenges, tools, and technology considerations are provided in [14, 16, 29, 30].

In [31], the authors worked on dimensions in big data and data analysis and then focused attention on the issue of inconsistencies and their impact on data analysis. They introduced four classifications of inconsistencies, frameworks, and tools in big data and pointed out the efficiency of inconsistency-induced learning as a tool for data analysis.

In [32], the authors discussed the infrastructure of big data service architecture, which includes collecting and storing data. Practical application scenarios of Flume, Kafka, GFS, HDFS, and different data processing modes are focused. They introduced mobile big data development, the commonly applied data analysis techniques. Three typical mobile big data analysis applications, namely, wireless channel modeling, human offline and online behavior analysis, and speech recognition in the Internet of Vehicles, are introduced. To obtain the minimum consistent subset, the concepts of edited nearest neighbor, traditional condensed nearest neighbor, and reduced nearest neighbor were used [33].

In [34], the authors spotted light on complications faced by data scientists in integrating and implementing a high volume of medical data acquired from multiple platforms. Provided information about the medical field, digitization of information, bioinformatics tools, and suggested improvements required in the e-health sector. Table 1 describes key findings and the scope of research contributions in big data and machine learning.

4. Stimulation towards Big Data

The emerging goal of massive data analysis is to update commodity techniques such as rule-based systems, model mining, and decision trees. Other data mining techniques to develop business rules on massive data sets effectively can be realized either by improving algorithms that use the storage of distributed data, computation in memory, or by using the computation in clusters for parallelism [48]. Previously, this was done using grid computing, which has been taken by cloud computing in the present day [49].

Traditional databases use code locality to process the data, but it is hard to import or export. Meanwhile, there may be a chance that data gets manipulated. Processing the data within time is also a big deal [22]. Cloud techniques help to work effectively, revolutionizing the IT industry by adding elasticity to the way it was consumed and enabling organizations to pay only for the resources and services they use. It is necessary to integrate big data issues with cloud technologies to produce effective results in the current scenario [50].

To handle large records, statistics integration usually needs to extract vast quantities of records from massive sources [51]. These dispensed statistics are desired to gather with the aid of a suitable device or software program, and information storage control schemes need to be provided for these massive facts within the sequential processing steps. The forms of big data majorly involved are static batch data and dynamic stream data [52]. Batch data processing is stored in a static format, and stream data is a continuous real-time data instance sequence. The streaming data cannot be stored fully, and various elements can be removed after processing.

4.1. 6S: A Beeline towards Big Data.

The following points present why all major sectors are adopting big data technologies.

(1) *Supervision: better data management*. The majority of data processing platforms and business intelligence tools let data scientists sit in one place and drive the data analysis, helping to perform various types of operations without technical complications. This includes organizing, collecting, and storing. Spending on analytics to gain competitive intelligence on future market conditions, to target customers more successfully, and to optimize operations and supply chains generates operating profit increases profits

(2) *Suppleness: better speed, capacity, and flexibility*. Big data services can be provided to utilize substantially large data sets which can provide the necessary storage and computing power to change data according to requirement

(3) *Sageness: better visualization*. Data visualization tools support observing data in pictorial representations like graphs which can be easy to understand. In real-time scenarios also visualization tools can process voluminous data quickly. Managers can use big data to understand more about their businesses insights and transform the generated knowledge into effective decisions to enhance company performance

(4) *Scope: better opportunities*. More consumers understand the competitive benefit of becoming a data-driven company as big data analytics technologies improve. Nowadays, marketers majorly focus on sentiment analysis, where they can collect data on how customers think about certain products and services by analyzing consumer responses on social media sites like Facebook and Twitter

(5) *Statistics: better data analysis methods*. The data are not just figures in a database anymore. Text, audio, and video files can also provide valuable insight; good tools can even recognize specific models according to predefined criteria. This happens in large part through the use of natural language processing tools, which can be essential for text mining, sentiment, and clinical analysis. The healthcare sector uses big data to improve patient care and to find better ways to manage resources and personnel

(6) *Surety: better risk analysis*. Risk is a facet of almost every business decision. There is no way to avoid risk, especially when a company is looking for growth, diversifying products, or trying to achieve new targets. In addition to financial markets, big data risk management can be applied in healthcare, retail, manufacturing, and e-commerce and can be applied to a wide variety of corporate threats, such as regulatory risk [53]. Financial institutions can fastly find that big data analysis is expert at identifying fraud before it becomes extensive, preventing further damage to the organization

5. Big Data Applications

5.1. Recommendations to Customers.

Customer data are collected in various no. of ways, including what websites they browse, where they reside, when they approach customer support, and if they communicate with their brand on social media. It is a massive amount of seemingly unrelated data, but organizations that can correctly mine it may provide a more tailored experience [39]. Companies must offer the appropriate products to the targeted consumer on the right

Table 1: Major contributions in big data and machine learning research.

References	Year	Scope of analysis	Key findings
Islam et al. [35]	2022	Detection of distributed DDOS attacks on financial organizations. Multiple classification models are used for the prediction of DDOS attacks. SVM, KNN, and RF algorithms are used.	SVM produces the best results, compared with other algorithms.
Najar and Naik [36]	2022	Detection of DDoS attack packets using K-nearest neighbor, random forest, support vector machine, and multilayer perceptions.	Random forest shows better performance compared with other algorithms.
Ananthu et al. [37]	2021	Recognizing fraud transactions by analyzing the transaction records and integrating big data with machine algorithms for accurate and fast detection.	Performed comparison of RF, logistic regression, and decision tree classifier; RF produces better results.
Wang et al. [38]	2020	Big data architecture, data collection and storage processing using ETL, different types of NoSQL databases with their merits, demerits, scenarios, and various processing strategy modes are discussed.	Investigated service, present architecture, and cloud services in big data.
Mahmud et al. [39]	2020	Discussed problems in sampling and partitioning of data analysis, record and block-level samplings, and three classical horizontal partitioning schemes.	Sampling-based approximation projects are considered for analysis.
Shoumy et al. [40]	2020	Applications, trends, and state of art technical analysis and their performances.	Building a comprehensive multivariant database for qualitative analysis.
Hasliza et al. [29]	2020	Challenges and issues faced in the consolidation of data and implementation in the public sector.	Consequences in data due to lack of management support, policies, human errors, and improper maintenance.
Ketu et al. [23]	2020	Illustrated analysis on Hadoop and Spark, evaluation can be done based on functional principle, performance, compatibility, failure tolerance, cost, flexible use, data processing, and security.	In-memory computations of spark are more effective than on-disk Hadoop computations.
Kshirsagar and Kumar [41]	2020	Feature reduction method based on correlation, information gain ratio, and ReliefF.	Accuracy and feature section procedure was improved by implementing PART classifier.
Hiriri et al. [34]	2019	Presented a background study of V's, theories and techniques involved in big data, and comparison of uncertain types respective of data analysis.	Discussed state of art analytical techniques, the impact of uncertainty, and open issues focused on data related to financial sector decision processing.
Rao et al. [42]	2019	Tools, technologies, and functionalities involved in big data, parameters to be considered for Hadoop query processing, HACE theorem, and various data sources for data analysis.	Considered real-time scenarios on distributed ML tools presented core features of recent developments in large-scale graph processing and tools
Bindra and Sood [43]	2019	Detection of DDoS attacks by applying machine learning models.	Random forest is the best choice for identifying DDoS attacks.
Inoubli et al. [3]	2018	Investigated challenges in big data and provided a glance of various frameworks, presented an experimental analysis, and a comparative study of the most in demand frameworks with various batch and iterative workloads.	A comparison study on frameworks, processing models, and best practices.
Kolajo et al. [44]	2018	Focused on fraud detection systems, analyzing massive streams of credit card transactions, addressing verification latency, class imbalance, and concept drift.	Impact of concept drift and class unbalance in a real-world data set consisting of 75 million transactions.
Chen et al. [45]	2018	Classification of DDoS attack.	Classification model built on spark framework to achieve performance and accuracy.
Qiu et al. [46]	2016	Provided promising learning methods such as transfer learning, active learning, representation learning, deep learning, distributed learning, parallel learning, and kernel-based learning.	The connection of modern signal processing technologies with machine learning was analyzed and provided open issues and new research trends.
Landset et al. [47]	2015	Discussed different processing paradigms and comparison of engines including MapReduce, Spark, Flink, and Storm.	Flink gives the results with a combination of batch and streaming models.

channel to accurately forecast the future and maintain resources in terms of human capital management [54].

5.2. Medical Field. Healthcare analysis aims to assist doctors in making data-driven decisions in proper time and improving patient care [55, 56]. This is more effective in the scenario of one who has a long medical history and is suffering from several health issues [57]. New AI systems and technologies would also be able to anticipate who is at risk of diabetes, allowing for additional screenings or weight control to be recommended [58].

5.3. Mobile Communications. Mobile service providers can inspect network speed and control the whole network using network analytics, which is a huge concern in telecom [59]. This enables network faults to be resolved in a matter of minutes while also improving service quality and customer satisfaction. With the proliferation of smartphones, location-based support services can be supplied to customers upon demand, based on analysis of real-time location and behavioral data [60]. This could increase the number of people who use mobile services [40].

5.4. Economic Firms. Financial analytics provides compelling chances to improve predictive modeling and better prediction of rates of return and investment effects. More precise forecasts and the capacity to lessen the inborn perils of monetary exchange result from admittance to large information and more noteworthy algorithmic information. Companies are attempting to comprehend client needs and preferences to guess future behaviors, grow in sales leads, utilize new channels and technology, meet consumer needs, and enhance customer satisfaction [61–64].

5.5. Customer Practice. The marketing functions of social media are constantly being investigated and employed nowadays, and their economic value is more apparent. New business trends in social media are moderately obtaining acceptance and popularity among consumers. Entrepreneurs can master even more extensive personal information about consumers through social media, which allows them to correctly assess their personal preferences, activities, and other information, allowing them to effectively address deep needs and access potential demands [60].

5.6. Business Marketing. The more information a company gathers, the more opportunities it has to make a pave in marketing, provide better services, increase consumer interaction, boost its brands, and reach the right people. Advertising is also an important aspect of marketing. In fact, marketing can be grown by advertising products in numerous channels [32]. Therefore, if a company adopts a practical approach to the data it collects from various channels, it can change the entire marketing strategy through data analysis.

5.7. Law Enforcement. Using historical data, such as kind of crime, location, scheme, social media data, drone, and smartphone tracking, law enforcement officers attempt to forecast the next crime site [65]. The police agency could determine the spatial association between crime sites and environmental factors.

5.8. New Product Development. The process of introducing a new product involves a lot of trial and error. Big data eliminates guesswork and aids in the development of perfectly suitable through effective management [66]. It makes to achieve product optimization, time and cost reduction, and service offering that would lead to less supervision and latency reduction; therefore, the minimum amount of resources is required [67].

5.9. Banking. In service delivery and operations, the banking sector has progressed by leaps and bounds over the past few decades [68]. Remarkably, most banks have not been able to utilize the data stored in their systems. Through proper data analysis, bank scans achieve fraud detection, prevention, customer segmentation, risk management, and recognized product offerings [69].

5.10. Education. In education, the pedagogical judgments made by a committee to assess a student's knowledge of the content or structuring of a course may have more impact on student learning and graduation rates [70]. This increases learning efficiency and not only enhances the student's experience but it also helps to evade some of the educational system's needs [71]. Data analytics opens up new possibilities for improving education by assisting instructors and students in making better decisions earlier in the learning process. Developments in applying data science to drive process innovation are rapidly expanding.

5.11. Emotional Analysis. The magnification of various social association networks produces huge data related to emotional analysis. It is a procedure of measuring peer emotions, thoughts, and results used to draw out effective information [71]. Machine learning algorithms play a meaningful role in this process; algorithms like SVM, K-nearest neighbor, genetic algorithm, ANN, and random forests can be used to facilitate this analysis [72].

6. Big Data Challenges

Big data originated with autonomous sources is heterogeneous, large-volume, distributed, and decentralized control and seeks to explore complex and evolving relationships among data [73].

Innovative data analysis tools and techniques must be required to meet large-scale data set analysis challenges and targets [74]. Therefore, we require a potential framework that meets processing speed and scalable storage systems for large-scale datasets. A lot of good research has been done to overcome these issues [75, 76]. Key challenges are categorized as data characteristics V's, process, and management challenges.

6.1. Data Characteristics: V. Big data producing data on a large scale poses three significant problems. They are data volume, velocity, and a variety of data. These are referred to as the 3 V-model of big data. Further, the model has been

Table 2: The V models.

S no.	Model name	V-full forms
1.	3 V's	Volume, velocity, variety
2.	4 V's	3 V's+veracity.
3.	5 V's	4 V's+value.
4.	7 V's	5 V's+variability and visualization.
5.	10 V's	7 V's+validity, vulnerability, and volatility.
6.	13 V's	10 V's+vocabulary, vagueness, venue.
7.	42 V's	13 V's+vane, vanilla, vantage, variability, varifocal, varmint, varnish, vaticination, vault, veer, veil, versed, vexed, victual, viral, virtuosity, viscosity, vivify, voice, voodoo, voyage, vulpine, valor, verdict, version control, vibrant, vet, vastness, and visibility.
8.	51 V's	42 V's+verification, verbosity, versatility, voluntariness, virtualization, violation, vitality, verve, and venturesomeness.

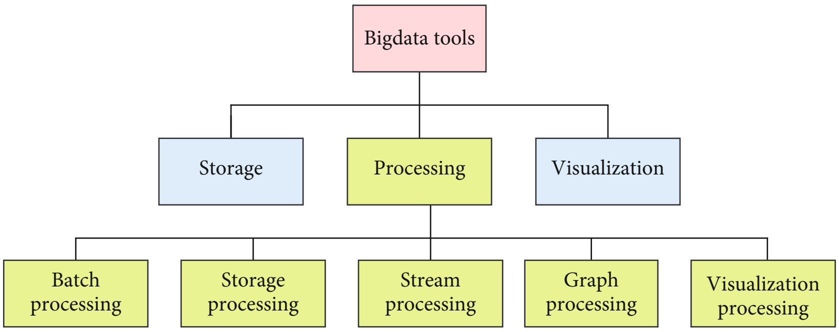


Figure 4: Big data processing tools classifications.

extended as 4 V's, 5 V's, 7 V'S, 10 V's, 13 V's, 42 V's, 44 V's, and 51 V's [77, 78], as shown in Table 2.

6.2. Process Challenges. These are related to processing and analyzing large datasets. It creates a major challenge, as the data is available in various forms and changing it into a suitable form for analysis is a challenging task. It involves data acquisition and storage, preprocessing, data analysis, modeling, and data visualization.

6.3. Management Challenges. These are related to the challenges encountered by an organization which are related to the privacy, security, and governance of data. These are also faced due to a lack of skilled experts who are well known for the trending tools and techniques. To place the appropriate method for dealing with each phase of data. Security is always a major concern as data is highly confidential such as financial and military data.

7. Tools for Big Data Analysis

The computational tools for big data are used to process data at different levels, which will help integrate and analyze various processing mechanisms. Big data software tools extract information from many data sources and process it. In conventional databases, tracking this data is quite challenging. As a consequence, we can employ tools to manage data [74]. Most of the big data technology logos include jungle animals. Logos are intended to signify something and often have a backstory. Big data can be represented with an elephant image that expresses that it is giant, intense, and complex to handle. Some more examples include pig, hive, and zookeeper. The evolution of tools required latency, throughput, fault tolerance, usability, resource expense, and scalability. Major classifications of processing tools are shown in Figure 4.

Before choosing a tool, one has to check the following aspects:

(1) License cost

(2) Customer service quality

(3) The expense of teaching staff

(4) The tool support and updates the policy of the vendor's software requirements

(5) Company assessment

7.1. Taxonomy of Tools

7.1.1. Storage Processing. The quick expansion of information requires more stringent data storage and management requirements [79]. The management and storage of large-scale data sets while maintaining data access dependability and availability are said to as big data storage. Major concepts include substantial storage systems, distributed storage solutions, and effective data storage mechanisms. On the one hand, the storage architecture should provide dedicated

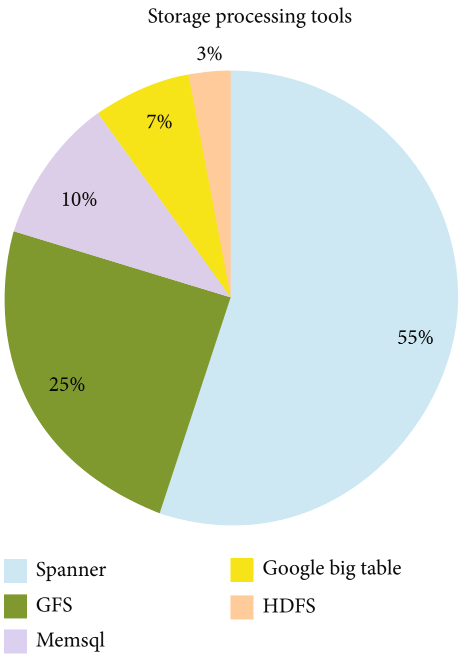


Figure 5: Storage processing tool statistics.

space for information storage; on the other hand, it needs to provide a robust, accessible interface for queries and understanding large amounts of data [80]. Data storage devices are traditionally used as auxiliary server equipment to store, manage, search, and analyze data using structured RDBMS. With the rapid improvement of data, data storage devices are getting more significant, and numerous internet companies are pursuing large storage capacities to stay competitive. As a result, there is an additional demand for data storage in research. For storage processing analysis, tools considered are GFS, HDFS, spanner, memSQL, Google Bigtable, Sqoop, and Flume, and among them, Figure 5 presents the popular storage processing tools statistics.

7.1.2. Batch Processing. Batch processing is essential for companies to manage massive amounts of data effectively. Particularly well suited to working frequent repetitive chores like accounting, the fundamentals of batch processing are the same in every business and for every project [70]. It has become prominent due to its numerous benefits in the field of enterprise data management. Batch processing has several advantages for businesses. Efficiency, when computing or other tools are readily available, helps process jobs. Companies can schedule batch operations for jobs that are not as urgent and prioritize time-sensitive jobs [81]. It can also be processed in the background to lower the processor burden. Compared to stream processing, batch processing is a less sophisticated system that does not require particular hardware or system support for data entry [44]. It requires minimal maintenance. For batch processing analysis, consider the following tools: Hadoop, Dryad, Mahout, Jaspersoft, Pentaho, Skytree Server, Tableau, Karmasphere, Talend, and MapReduce, and among them, Figure 6 presents the popular batch processing tools statistics.

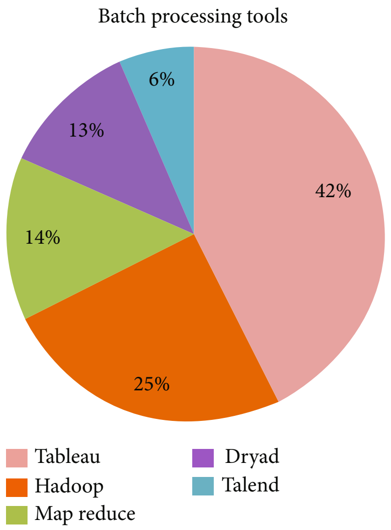


Figure 6: Batch processing tool statistics.

7.1.3. Stream Processing. The majority of data is now created in the structure of a stream. Batch data is just a snapshot of low-level data taken at a specific point in time [44]. In this perspective, data is delivered rapidly, one instance at a time, and algorithms must analyze it in a single pass while adhering to very severe space and time limits [82]. Streamlining algorithms apply probabilistic assurance to provide quick estimated results [83]. Settings in the streaming paradigm have been developed and abstracted by researchers.

On the one hand, MapReduce is unsuitable for expressing streaming algorithms [84]. On the other hand, conventional online algorithms are constrained by the memory and bandwidth of a single console. DSPEs (distributed stream processing engines) are a new breed of MapReduce-inspired technologies that aim to solve this problem [85]. These engines enable parallel computing to be expressed as streams, combining multiprocessor scalability with the efficiency of streaming algorithm techniques [86]. For stream processing tools analysis, considered the following tools: Storm, S4, SQLstream s-Server, Splunk, Apache Kafka, SAP HANA, Samza, Flink, Samoa, Millwheel, heron, cloud-based streaming, Amazon Kinesis, s2, Microsoft Azure, and IBM streaming analysis. Among this, Figure 7 gives a statistical view of popular stream processing tools.

7.1.4. Graph Processing. Graph analysis can be employed to produce recommendations and customization models for customers and to take critical decisions based on the data analysis findings [87]. This helps the enterprises potentially guide customers to buy their products, marketing approach, and customer service behavior. Several scenarios present graph databases as a more suitable match for data management than relational databases and other NoSQL data storage [83].

Graph data solutions assist in detecting fraudulent transactions in a payment processing application using related data that comprises people, transactions, products, and events. Topic modeling entails approaches for grouping documents and extracting thematic representations from the

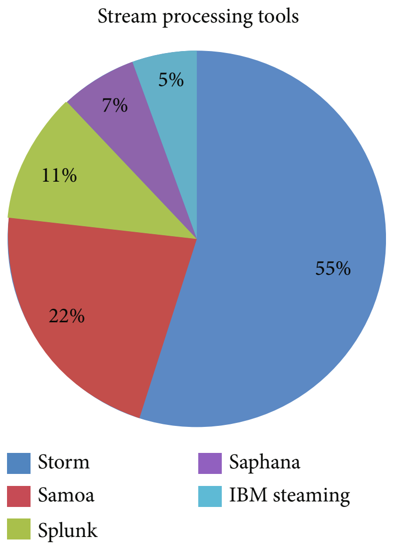


Figure 7: Stream processing tool statistics.

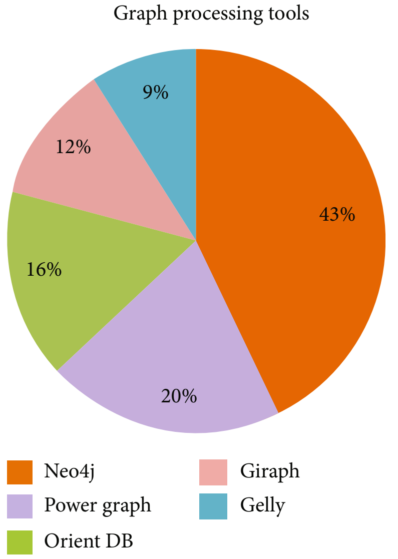


Figure 8: Graph processing tool statistics.

information included in those sources [88]. In social network applications, the shortest distances and pathways are also significant. They can be used to determine the network importance of a specific user. Closer users are more relevant than those distant away, as can be predicted. For graph processing tools analysis, considered the following tools: Neo4j, Orientdb, power graph, Gelly, Giraph, GraphLab, Hama, GraphX, and Pregel. Among this, Figure 8 gives a statistical view of popular tools in graph processing.

7.1.5. Hybrid Processing. Hybrid process systems incorporate both batch and stream processing mechanisms, which incorporate processing units that are functioning at their optimum efficiency to execute one or more process jobs. Combining the APIs and related components, it facilitates multiple data processing procedures [89]. Due to its sustainable and innovative processing choices, it has become a choice for many complex operations.

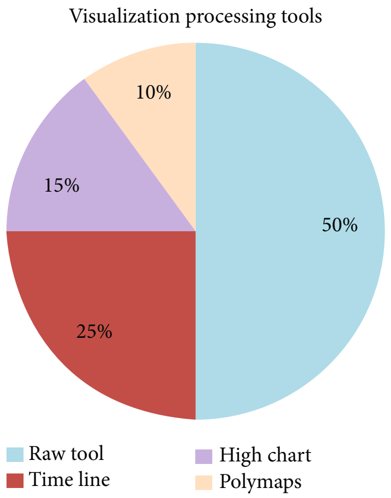


Figure 9: Visualization processing tool statistics.

7.1.6. Visualization Processing. Data visualization is one of the essential instruments for determining a qualitative understanding. It can be used to convey and highlight the key relationships in layouts and charts that are more useful to stakeholders to measure the correlation or significance [46, 54, 78, 90]. This might be useful when examining a dataset and retrieving features about it, and spotting patterns, corrupt data, outliers, and other things [38]. For analysis of visualization, processing tools considered the following: D3, raw graphs, Google charts, modest maps, open heat map, color brewer, datawrappers, digraphs, chartjs, charted, infogram, chart blocks, polymaps, and ember charts, and among this, Figure 9 presents the popular tools for visualization processing.

The above pie chart resembles popular tools in storage, batch, stream, visualization, and graph processing that are used in various application areas such as health, finance, social communications, business, and industrial areas for analysis. It also considers different literature papers and Google Trends data. The analysis in Figure 10 will be helpful in selecting the appropriate tool for a specific sector.

8. Future Directions

The results of this methodical study address various implications for the researchers who consider further exploring the consequence of big data technology and tools. In addition, this study will also contribute comprehensive information to practitioners who are involved in developing and using big data analysis and techniques to improve the quality of their services. It enables researchers to identify areas where further research efforts are needed. It is evident that even though several research works deal with the development and evaluation of advanced analytical techniques, there is still a dearth of information on the implementation of data analytics. By combining data analytics with machine learning, world can obtain more productive results. Big data tools

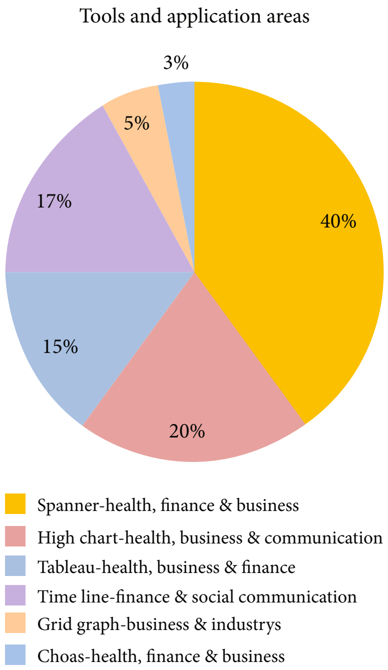


Figure 10: Popular tool and application areas.

have also started adopting new technologies to facilitate the users, but it is still required new tools to face new challenges arising from various application fields.

9. Conclusion

Technology has ushered in a new era of progress over the past two decades. This research study conceptualized the big data importance analysis of both structured and unstructured information using various tools. It is regarded as the foundation for all decision-making responsibilities and has become an essential component in the majority of company operations. The continuously growing nature of user's day-to-day activities in various sectors brings new concerns to the world. So always, data scientists can combine data with numerous new emerging methodologies to solve users' problems [91]. There is a need to build new standards to make better information systems. V's and 6S aid in becoming aware of big data features and APIs that allow clients to make potential strategies for adaptable outcomes. To properly manage massive data, working through parallel processing gives effective results. In the future, there may be scope for enhancing big data V's as there are more than 100 V's, as well as researchers need to do more explorations in this field. In the case of tool analysis, nearly 50 popular tools were considered. In the subsequent works, we would like to extend it with more tools along with PPML concepts.

Data Availability

The data supporting this systematic review are from previously reported studies, which have been cited. The raw data supporting the conclusions of this article will be made available by the author, without undue reservation.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Conceptualization was worked on by S.D. and R.K. Data curation and formal analysis were conducted by S.D. and R.K. The investigation and methodology were performed by R.K. Project administration was done by S.D. and R.K. Resources were obtained by S.D. and R.K. Supervision was conducted by R.K. Validation was accomplished by R.K. The visualization was completed by S.D. and R.K. Writing—review and editing—were done by S.D. and R.K. The authors have read and agreed to the published version of the manuscript.

Acknowledgments

I would like to convey my humble gratitude to Dr. Rajesh Kaluri my research supervisor for his effective guidance and motivation in every step of this research work.

References

[1] H. Margetts and C. Dorobantu, "Rethink government with AI," *Nature*, vol. 568, no. 7751, pp. 163–165, 2019.

[2] P. Beri and S. Ojha, "Comparative analysis of big data management for social networking sites," in *Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016*, pp. 1196–1200, New Delhi, India, 2016.

[3] W. Inoubli, S. Aridhi, H. Mezni, M. Maddouri, and E. Mephu Nguifo, "An experimental survey on big data frameworks," *Future Generation Computer Systems*, vol. 86, pp. 546–564, 2018.

[4] I. Lee, "Big data: dimensions, evolution, impacts, and challenges," *Business Horizons*, vol. 60, no. 3, pp. 293–303, 2017.

[5] X. Wu, X. Zhu, G. Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[6] M. Larch, J. Wanner, Y. V. Yotov, and T. Zylkin, "Currency unions and trade: a PPML re-assessment with high-dimensional fixed effects," *Oxford Bulletin of Economics and Statistics*, vol. 81, no. 3, pp. 487–510, 2019.

[7] N. Bhandari and P. Pahwa, "Comparative analysis of privacy-preserving data mining techniques," in *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018*, vol. 56, pp. 535–541, Singapore, 2019.

[8] J. Zhou, Z. Cao, X. Dong, and X. Lin, "PPDM: a privacy-preserving protocol for cloud-assisted e-healthcare systems," *IEEE Journal on Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1332–1344, 2015.

[9] I. A. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of "big data" on cloud computing: review and open research issues," *Information systems*, vol. 47, pp. 98–115, 2015.

[10] D. Arunachalam, N. Kumar, and J. P. Kawalek, "Understanding big data analytics capabilities in supply chain management: unravelling the issues, challenges and implications for

practice," *Transportation Research Part E: Logistics and Transportation Review*, vol. 114, pp. 416–436, 2018.

[11] R. A. Ariyaluran Habeeb, F. Nasaruddin, A. Gani, I. A. Targio Hashem, E. Ahmed, and M. Imran, "Real-time big data processing for anomaly detection: a survey," *International Journal of Information Management*, vol. 45, pp. 289–307, 2019.

[12] F. Liang, W. Yu, D. An, Q. Yang, X. Fu, and W. Zhao, "A survey on big data market: pricing, trading and protection," *IEEE Access*, vol. 6, pp. 15132–15154, 2018.

[13] A. Gandomi and M. Haider, "Beyond the hype: big data concepts, methods, and analytics," *International Journal of Information Management*, vol. 35, no. 2, pp. 137–144, 2015.

[14] A. K. Bhadani and D. Jothimani, "Big data: challenges, opportunities and realities," *Effective big data management and opportunities for implementation*, pp. 1–24, 2017, https://www.igi-global.com/chapter/big-data/157681.

[15] J. Singh and V. Singla, "Big data: tools and technologies in big data," *International Journal of Computer Applications*, vol. 112, no. 15, pp. 975–8887, 2015.

[16] F. L. F. Almeida, "Benefits, challenges and tools of big data management," *Journal of Systems Integration*, vol. 8, no. 4, pp. 12–20, 2017.

[17] Y. Hajjaji, W. Boulila, I. R. Farah, I. Romdhani, and A. Hussain, "Big data and IoT-based applications in smart environments: a systematic review," *Computer Science Review*, vol. 39, article 100318, 2021.

[18] S. Ren, Y. Zhang, Y. Liu, T. Sakao, D. Huisingh, and C. M. V. B. Almeida, "A comprehensive review of big data analytics throughout product lifecycle to support sustainable smart manufacturing: a framework, challenges and future research directions," *Journal of Cleaner Production*, vol. 210, pp. 1343–1365, 2019.

[19] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna et al., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020.

[20] S. Kumar and M. Singh, "Big data analytics for healthcare industry: impact, applications, and tools," *Big data mining and analytics*, vol. 2, no. 1, pp. 48–57, 2019.

[21] E. Park, Y. Jang, J. Kim, N. J. Jeong, K. Bae, and A. P. del Pobil, "Determinants of customer satisfaction with airline services: an analysis of customer feedback big data," *Journal of Retailing and Consumer Services*, vol. 51, pp. 186–190, 2019.

[22] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Information Fusion*, vol. 42, pp. 146–157, 2018.

[23] S. Ketu, P. K. Mishra, and S. Agarwal, "Performance analysis of distributed computing frameworks for big data analytics: Hadoop vs Spark," *Computación y Sistemas*, vol. 24, no. 2, pp. 669–686, 2020.

[24] M. K. Saggi and S. Jain, "A survey towards an integration of big data analytics to big insights for value-creation," *Information Processing and Management*, vol. 54, no. 5, pp. 758–790, 2018.

[25] C. A. Ardagna, P. Ceravolo, and E. Damiani, "Big data analytics as-a-service: Issues and challenges," in *In 2016 IEEE international conference on big data*, pp. 3638–3644, Washington, DC, USA, 2016.

[26] V. Marx, "The big challenges of big data," *Nature*, vol. 498, no. 7453, pp. 255–260, 2013.

[27] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of big data*, vol. 2, no. 1, pp. 1–21, 2015.

[28] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[29] N. Hasliza, M. Hassan, K. Ahmad, and H. Salehuddin, "Diagnosing the issues and challenges in data integration implementation in public sector," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 10, no. 2, pp. 529–535, 2020.

[30] C. L. Philip Chen and C. Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: a survey on big data," *Information sciences*, vol. 275, pp. 314–347, 2014.

[31] A. Mohamed, M. K. Najafabadi, Y. B. Wah, E. A. K. Zaman, and R. Maskat, "The state of the art and taxonomy of big data analytics: view from new big data framework," *Artificial Intelligence Review*, vol. 53, no. 2, pp. 989–1037, 2020.

[32] J. Xie, Z. Song, Y. Li et al., "A survey on machine learning-based Mobile big data analysis: challenges and applications," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8738613, 19 pages, 2018.

[33] M. L. Song, R. Fisher, J. L. Wang, and L. B. Cui, "Environmental performance evaluation with big data: theories and methods," *Annals of Operations Research*, vol. 270, no. 1, pp. 459–472, 2018.

[34] R. H. Hariri, E. M. Fredericks, and K. M. Bowers, "Uncertainty in big data analytics: survey, opportunities, and challenges," *Journal of Big Data*, vol. 6, no. 1, pp. 1–6, 2019.

[35] U. Islam, A. Muhammad, R. Mansoor et al., "Detection of distributed denial of service (DDoS) attacks in IOT based monitoring system of banking sector using machine learning models," *Sustainability*, vol. 14, no. 14, p. 8374, 2022.

[36] A. A. Najar and S. Manohar Naik, "DDoS attack detection using MLP and random forest algorithms," *International Journal of Information Technology*, vol. 14, no. 5, pp. 2317–2327, 2022.

[37] S. Ananthu, N. Sethumadhavan, and H. Narayanan Ag, "Credit card fraud detection using Apache Spark analysis," in *2021 5th international conference on trends in electronics and informatics (ICOEI)*, pp. 998–1002, Tirunelveli, India, 2021.

[38] J. Wang, Y. Yang, T. Wang, R. S. Sherratt, and J. Zhang, "Big data service architecture: a survey," *Journal of Internet Technology*, vol. 21, no. 2, pp. 393–405, 2020.

[39] M. S. Mahmud, J. Z. Huang, S. Salloum, T. Z. Emara, and K. Sadatdiynov, "A survey of data partitioning and sampling methods to support big data analysis," *Big Data Mining and Analytics*, vol. 3, no. 2, pp. 85–101, 2020.

[40] N. J. Shoumy, L. M. Ang, K. P. Seng, D. M. Rahaman, and T. Zia, "Multimodal big data affective analytics: a comprehensive survey using text, audio, visual and physiological signals," *Journal of Network and Computer Applications*, vol. 149, article 102447, 2020.

[41] D. Kshirsagar and S. Kumar, "An efficient feature reduction method for the detection of DoS attack," *ICT Express*, vol. 7, no. 3, pp. 371–375, 2021.

[42] T. R. Rao, P. Mitra, R. Bhatt, and A. Goswami, "The big data system, components, tools, and technologies: a survey," *Knowledge and Information Systems*, vol. 60, no. 3, pp. 1165–1245, 2019.

[43] N. Bindra and M. Sood, "Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset," *Automatic Control and Computer Sciences*, vol. 53, no. 5, pp. 419–428, 2019.

[44] T. Kolajo, O. Daramola, and A. Adebiyi, "Big data stream analysis: a systematic literature review," *Journal of Big Data*, vol. 6, no. 1, pp. 1–30, 2019.

[45] L. Chen, Y. Zhang, Q. Zhao, G. Geng, and Z. Yan, "Detection of dns ddos attacks with random forest algorithm on spark," *Procedia computer science*, vol. 134, pp. 310–315, 2018.

[46] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, 2016.

[47] S. Landset, T. M. Khoshgoftaar, A. N. Richter, and T. Hasanin, "A survey of open source tools for machine learning with big data in the Hadoop ecosystem," *Journal of Big Data*, vol. 2, no. 1, pp. 1–36, 2015.

[48] C. Kacfah Emani, N. Cullot, and C. Nicolle, "Understandable big data: a survey," *Computer science review*, vol. 17, pp. 70–81, 2015.

[49] N. R. Vajjhala and E. Ramollari, "Big data using cloud computing-opportunities for small and medium-sized enterprises," *European Journal of Economics and Business Studies*, vol. 4, no. 1, p. 129, 2016.

[50] D. Zhang, "Inconsistencies in big data," in *IEEE 12th International Conference on Cognitive Informatics and Cognitive Computing*, pp. 61–67, New York, NY, USA, 2013.

[51] P. Struijs, B. Braaksma, and P. J. H. Daas, "Official statistics and big data," *Big Data & Society*, vol. 1, no. 1, article 205395171453841, 2014.

[52] R. Sahal, J. G. Breslin, and M. I. Ali, "Big data and stream processing platforms for industry 4.0 requirements mapping for a predictive maintenance use case," *Journal of Manufacturing Systems*, vol. 54, pp. 138–151, 2020.

[53] O. M. Araz, T. M. Choi, D. L. Olson, and F. S. Salman, "Role of analytics for operational risk management in the era of big data," *Decision Sciences*, vol. 51, no. 6, pp. 1320–1346, 2020.

[54] R. H. Hamilton and W. A. Sodeman, "The questions we ask: opportunities and challenges for using big data analytics to strategically manage human capital resources," *Business Horizons*, vol. 63, no. 1, pp. 85–95, 2020.

[55] L. Hong, M. Luo, R. Wang, P. Lu, W. Lu, and L. Lu, "Big data in health care: applications and challenges," *Data and information management*, vol. 2, no. 3, pp. 175–197, 2018.

[56] N. Mehta and A. Pandit, "Concurrence of big data analytics and healthcare: a systematic review," *International Journal of Medical Informatics*, vol. 114, pp. 57–65, 2018.

[57] S. Dash, S. K. Shakyawar, M. Sharma, and S. Kaushik, "Big data in healthcare: management, analysis and future prospects," *Journal of Big Data*, vol. 6, no. 1, pp. 1–25, 2019.

[58] Y. Duan, J. S. Edwards, and Y. K. Dwivedi, "Artificial intelligence for decision making in the era of big data - evolution, challenges and research agenda," *International Journal of Information Management*, vol. 48, pp. 63–71, 2019.

[59] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile networks and applications*, vol. 19, no. 2, pp. 171–209, 2014.

[60] M. Anshari, M. N. Almunawar, S. A. Lim, and A. Al-Mudimigh, "Customer relationship management and big data enabled: personalization & customization of services," *Applied Computing and Informatics*, vol. 15, no. 2, pp. 94–101, 2019.

[61] S. Akter, S. F. Wamba, A. Gunasekaran, R. Dubey, and S. J. Childe, "How to improve firm performance using big data analytics capability and business strategy alignment?," *International Journal of Production Economics*, vol. 182, pp. 113–131, 2016.

[62] S. F. Wamba, A. Gunasekaran, S. Akter, S. J. Ren, R. Dubey, and S. J. Childe, "Big data analytics and firm performance: effects of dynamic capabilities," *Journal of Business Research*, vol. 70, pp. 356–365, 2017.

[63] M. Ge, H. Bangui, and B. Buhnova, "Big data for internet of things: a survey," *Future Generation Computer Systems*, vol. 87, pp. 601–614, 2018.

[64] O. Müller, M. Fay, and J. Vom Brocke, "The effect of big data and analytics on firm performance: an econometric analysis considering industry characteristics," *Journal of Management Information Systems*, vol. 35, no. 2, pp. 488–509, 2018.

[65] D. S. Terzi, R. Terzi, and S. Sagiroglu, "A survey on security and privacy issues in big data," in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 202–207, London, UK, 2016.

[66] P. Pääkkönen and D. Pakkala, "Reference architecture and classification of technologies, products and Services for big data systems," *Big data research*, vol. 2, no. 4, pp. 166–186, 2015.

[67] P. Tabesh, E. Mousavidin, and S. Hasani, "Implementing big data strategies: a managerial perspective," *Business Horizons*, vol. 62, no. 3, pp. 347–358, 2019.

[68] Y. Sun, Y. Shi, and Z. Zhang, "Finance big data: management, analysis, and applications," *International Journal of Electronic Commerce*, vol. 23, no. 1, pp. 9–11, 2019.

[69] V. Grover, R. H. L. Chiang, T. P. Liang, and D. Zhang, "Creating strategic business value from big data analytics: a research framework," *Journal of management information systems*, vol. 35, no. 2, pp. 388–423, 2018.

[70] X. Jin, B. W. Wah, X. Cheng, and Y. Wang, "Significance and challenges of big data research," *Big data research*, vol. 2, no. 2, pp. 59–64, 2015.

[71] T. S. Ing, T. C. Lee, S. W. Chan, J. Alipal, and N. A. Hamid, "An overview of the rising challenges in implementing industry 4.0," *International Journal of Supply Chain Management*, vol. 8, no. 6, pp. 1181–1188, 2019.

[72] M. Bansal, A. Goyal, and A. Choudhary, "A comparative analysis of K-nearest neighbour, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning," *Decision Analytics Journal*, vol. 3, article 100071, 2022.

[73] A. Oussous, F. Z. Benjelloun, A. Ait Lahcen, and S. Belfkih, "Big data technologies: asurvey," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 4, pp. 431–448, 2018.

[74] S. Boubiche, D. E. Boubiche, A. Bilami, and H. Toral-Cruz, "Big data challenges and data aggregation strategies in wireless sensor networks," *IEEE Access*, vol. 6, pp. 20558–20571, 2018.

[75] A. Katal, M. Wazid, and R. H. Goudar, "Big data: issues, challenges, tools and good practices," in *2013 Sixth international conference on contemporary computing (IC3)*, pp. 404–409, Noida, India, 2013.

[76] J. L. Torrecilla and J. Romo, "Data learning from big data," *Statistics & Probability Letters*, vol. 136, pp. 15–19, 2018.

[77] M. Bansal, I. Chana, and S. Clarke, "A survey on IoT big data," *ACM Computing Surveys*, vol. 53, no. 6, pp. 1–59, 2021.

[78] N. Khan, A. Naim, M. R. Hussain, Q. N. Naveed, N. Ahmad, and S. Qamar, "The 51 v's of big data: survey, technologies, characteristics, opportunities, issues and challenges," in *Proceedings of the international conference on omni-layer intelligent systems*, pp. 19–24, Heraklion, Crete, Greece, 2019.

[79] G. George, M. R. Haas, and A. Pentland, "Big data and management," *Academy of management Journal*, vol. 57, no. 2, pp. 321–326, 2014.

[80] G. Bello-Orgaz, J. J. Jung, and D. Camacho, "Social big data: recent achievements and new challenges," *Information Fusion*, vol. 28, pp. 45–59, 2016.

[81] X. Zhao, J. Zhang, and X. Qin, "k NN-DP: handling data skewness in kNN joins using MapReduce," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 3, pp. 600–613, 2018.

[82] M. Zaharia, R. S. Xin, P. Wendell et al., "Apache Spark," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[83] H. Yan, D. Sun, S. Gao, and Z. Zhou, "Performance analysis of storm in a real-world big data stream computing environment," in *Collaborative Computing: Networking, Applications and Worksharing: 13th International Conference, CollaborateCom 2017*, pp. 624–634, Springer International Publishing, Edinburgh, UK, 2018.

[84] L. Abualigah and B. Al Masri, "Advances in MapReduce big data processing: platform, tools, and algorithms," *Artificial Intelligence and IoT*, vol. 85, pp. 105–128, 2021.

[85] N. Tantalaki, S. Souravlas, and M. Roumeliotis, "A review on big data real-time stream processing and its scheduling techniques," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 35, no. 5, pp. 571–601, 2020.

[86] N. AlNuaimi, M. M. Masud, M. A. Serhani, and N. Zaki, "Streaming feature selection algorithms for big data: a survey," *Applied Computing and Informatics*, vol. 18, no. 1/2, pp. 113–135, 2022.

[87] S. Heidari, Y. Simmhan, R. N. Calheiros, and R. Buyya, "Scalable graph processing frameworks," *ACM Computing Surveys*, vol. 51, no. 3, p. 60, 2019.

[88] L. Belcastro, F. Marozzo, and D. Talia, "Programming models and systems for big data analysis," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 34, no. 6, pp. 632–652, 2019.

[89] D. Cheng, X. Zhou, Y. Wang, and C. Jiang, "Adaptive scheduling parallel jobs with dynamic batching in spark streaming," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 12, pp. 2672–2685, 2018.

[90] P. Kulkarni, M. Awwad, R. Bapna, and A. Marathe, "Big data analytics in supply chain: a literature review," in *Proceedings of the international conference on industrial engineering and operations management*, vol. 2018, pp. 418–425, Washington DC, USA, September 2018.

[91] S. Madden, "From databases to big data," *IEEE Internet Computing*, vol. 16, no. 3, pp. 4–6, 2012.

Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 4778967, 13 pages
https://doi.org/10.1155/2023/4778967

Research Article

A Multi-Blockchain-Based Cross-Domain Authentication and Authorization Scheme for Energy Internet

Donglan Liu, Xin Liu, Rui Wang, Hao Zhang, Fangzhe Zhang, Lili Sun, Honglei Yao, and Hao Yu

State Grid Shandong Electric Power Research Institute, Jinan 250003, China

Correspondence should be addressed to Donglan Liu; liudonglan2006@126.com

Received 6 August 2022; Revised 2 October 2022; Accepted 24 November 2022; Published 29 April 2023

Academic Editor: Zengpeng Li

The expansion of the scale of the Power Internet of Things stimulated by the development of the Energy Internet makes the growth in demand for the effective authentication and access control technologies in the cross-domain data exchange. Based on the cross-chain technology of the blockchain and the cuckoo filter, this paper proposes a cross-domain authentication scheme for Power Internet of Things. Firstly, a cross-chain authentication architecture is established. Combined with the existing authentication technologies used in intra-domain authentication, a cross-domain authentication process based on the cross-chain technology is proposed to realize the automatic transmission of the authentication credentials from application domain to authentication domain. The cuckoo filter is deployed on the blockchain as smart contracts, and the user certificate fingerprint is inserted into the filter to realize user registration, query, and revocation, which reduces the cost of the user certificate management. Experimental results show the effectiveness and feasibility of our scheme. Based on the proposed authentication scheme, a cross-domain access control scheme based on roles and object classes is presented, by treating the object classes as controlled objects and then applying the role-based access control to the object classes, on the condition that the heterogeneous domains in the Energy Internet have the same kinds of resources.

1. Introduction

In the scenario of Energy Internet and Power Internet of Things, there are a large number of terminal nodes deployed in a wide range, and the physical environments of some of the nodes are uncontrollable. This makes them vulnerable to many threats such as physical hijacking, node replication, signal interception, theft and replay, man in the middle attack, and so on. Therefore, the terminal authentication has a greater and greater impact on the security of the power system. The traditional Power Internet of Things access security mainly depends on the centralized key management mechanism, which has the disadvantages of low authentication efficiency and the risk of single point failure [1]. As many comprehensive services require cross-domain data sharing, centralized authentication and authorization can no longer meet the trust requirements for multiparty cross-domain business systems such as Power Internet of Things source-network-load-storage interaction and accurate material supply. The scenario is shown in Figure 1. Therefore, it is necessary to propose cross-domain authentication and authorizaiton schemes. In addition, in the design of cross-domain schemes, the limited computing resources, storage, and communication capabilities of Power Internet of Things terminals need to be considered.

To support cross-domain authentication, many solutions have been proposed. Generally speaking, these solutions can be divided into four categories: public key infrastructure (PKI) based solutions, identity-based encryption (IBE) based solutions, password-based solutions, and blockchain based solutions. PKI based schemes are only suitable for the Internet scenario rather than the Internet of Things scenario, as the result of the high cost of the certificate management. IBE schemes can eliminate the overhead of the certificate transmission, authentication, and maintenance [2] so that it is a potential solution for cross-domain

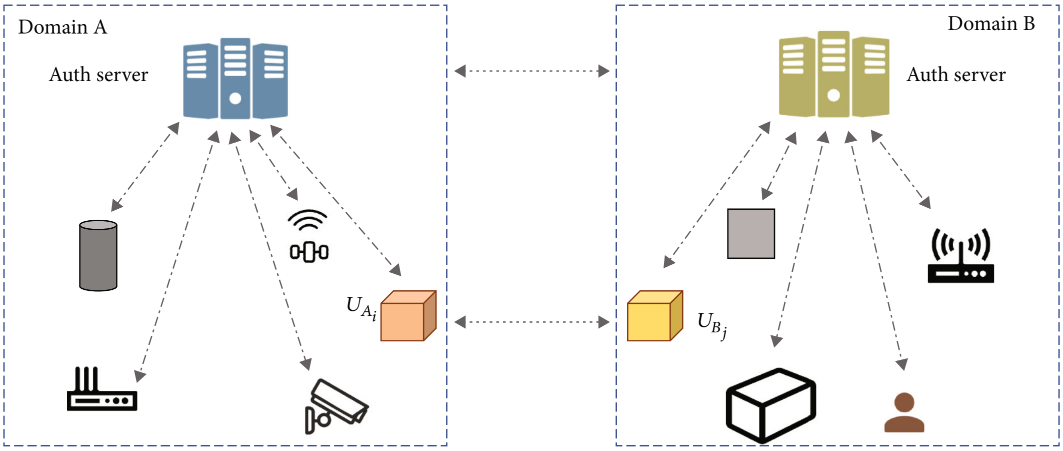


Figure 1: Cross domain scenario.

authentication. Blockchain is composed of data blocks in the form of ordered chain and can be treated as a distributed ledger maintained by multiple parties [3]. It uses encryption, Merkel tree, consensus mechanism, and other technologies to realize the transparency, tamper-resilience, and traceability of transaction data. Blockchain also provides a platform to deploy the smart contract [4], which is defined as a computer program that can automatically execute predefined protocols. Applying blockchain to cross-domain authentication can promote the efficiency and security.

1.1. Related Work. Cross-domain authentication can be realized based on public key infrastructure. For example, Vaidya et al. proposed the multidomain mechanism of V2G infrastructure [5], which supports point-to-point cross-domain authentication by constructing a model based on hybrid public key infrastructure and establishing the trust relationship between nodes through intra-domain and inter-domain digital identities. However, all PKI-based schemes require certification authority (CA) to store, issue, and manage digital certificates for each user [6], which costs heavy overheads. Since the introduction of the identity-based encryption, which does not need to manage digital certificates, researchers began to take advantage of it to design efficient protocols. In 2010, Cao et al. [7] proposed an efficient two-party authentication protocol reducing the number of interactive rounds. However, the protocol needs a trusted third party, and can not achieve cross-domain authentication between heterogeneous domains which have deployed different authentication mechenisms. Benzarti and Triki [8] designed an authentication framework based on identity encryption and signature, in which group identifier, object identifier, IP address, and a unique tamper-proof RFID tag of a user are combined with a temporary identity to realize authentication. However, the scheme cannot support cross-domain authentication. Shen et al. [9] proposed a cross-domain authentication scheme based on blockchain and identity-based signature.

As an effective method of the trust transmission in decentralized scenarios, blockchain technology has been widely studied in the field of Internet of Things security [10–19]. Ouaddah et al. [11] proposes a blockchain based access control framework for the Internet of Things. In terms of authentication, Fromknecht et al. [10] proposes a blockchain based distributed public key infrastructure (PKI) system, which records user certificates through the public general ledger to solve the single point of failure problem of the traditional PKI systems. In the resource sharing scenario, [12] proposed a cross-domain framework based on consortium blockchain technology. Yao et al. [14] proposed a blockchain assisted lightweight anonymous authentication mechanism. Guo et al. [15] designed a method to support authentication of different systems and domains. These schemes use blockchain to replace TTP, store tamper-proof authentication data, and support cross-domain authentication. However, they are inefficient and inflexible. Zhang et al. [16] proposes a thoroughly cross-domain authentication scheme based on blockchain, but this scheme does not consider the independent deployments of the blockchains in different domains, i.e., cross chain. Besides, it is inappropriate in this scheme that, when the user is about to be removed from the domain, it requires the user to actively ask for being revoked and needs some information generated by the user under his private key. For the blockchain-based authorization and access control, Gauhar et al. [18] proposed a blockchain based IoT authorization framework to realize authorization based on the authorization policies stored on the blockchain. In 2021, Zhu et al. [19] proposed an approach utilizing capability-based cross-domain access control and risk-based access control mechanisms in a domain while taking IoT nodes as data resources. However, these solutions cannot effectively leverage the access control mechanisms that are already adopted within the domains.

1.2. Our Contribution. The current researches mainly focus on using a single blockchain to improve the distribution and reliability of the authentication services [20]. However, the single blockchain structure is difficult to meet the requirements of the Energy Internet in terms of operation efficiency, maintenance cost, and privacy protection. To solve the problems of "incompletely cross-domain" [16], we propose a multi-blockchain-based cross-domain authentication scheme supporting heterogeneous domains for the Energy Internet. In our scheme, users coming from different domains can be authenticated by the authentication server of the domain he wants to access with the help of the chaincodes which have been deployed in the blockchain platforms. Each domain which may be an energy company can deploy its own blockchain which stores some authentication data of users in it to realize the isolation and data protection. There is a blockchain deployed to coordinate and audit the cross-domain authentication which is called supervision chain. We design a revocation process which is more suitable for the practical use, i.e., the authentication server in each domain can revoke users adaptively without users' private information. To realize the user revocation, we first initialize a cuckoo filter, and then user information is mapped to the fingerprint and inserted into the cuckoo filter. When the authentication server wants to revoke a user, it only needs to compute the fingerprint information of the user and delete the fingerprint from the cuckoo filter. To support cross-domain authorization, we assume that the heterogeneous domains have the same kinds of resources and treat the object classes as controlled objects. By applying the role-based access control to the object classes, we can extend our blockchain-based cross-domain authentication scheme and propose a cross-domain access control model based on roles and object classes.

The rest of the paper is organized as follows: In Section 2, we give the preliminaries of blockchain and cuckoo filter. In Section 3, we present the multi-blockchain-based cross-domain authentication scheme supporting heterogeneous domains for the Energy Internet. We also analyze the security and performance of the scheme in this section. In Section 4, we show how to extend the proposed authentication scheme to support the cross-domain authorization. Finally, we draw a conclusion in Section 5.

2. Preliminaries

We will provide in this section some preliminaries which are necessary for the understanding of the subsequent schemes.

2.1. Blockchain. Blockchain is a chain structure that links blocks in order [3]. A block is a collection of data in which relevant information and records are stored. The block is composed of a block header and a block body. The block header stores the version number, the hash value of the previous block, Merkle root, timestamp, etc. The block body contains the information of multiple transactions that have occurred since the previous block. The blockchain realizes the consistency of data through the consensus mechanism of self-trust. It uses encryption, Merkel tree, consensus mechanism, and other technologies to realize the transparency, tamper-resilience, and traceability of transaction data, and can be treated as a distributed ledger maintained by multiple parties. It can use the smart contract to automatically process data and realize the efficient and secure data exchange between entities without the need for a trusted third party. Blockchain provides a solution for trust establishment between entities in the distributed environment.

Smart contract [4] is a sequence of computer program with predefined protocols, which can be deployed on the blockchain. Smart contract can automatically execute the predefined protocols to complete information exchange and asset management. Smart contract is called "chaincode" in hyperledger fabric [21]. Chaincode runs in a secured container isolated from the endorsing peer process. Users can read and write a set of key-value pair status data on the ledger.

2.2. Cuckoo Filter. Cuckoo filter is a data structure based on cuckoo hashing algorithm [22]. The algorithm uses two hash functions h_1 and h_2 to map the element to be inserted to the corresponding position in one of the two maintained hash tables each of which has m elements. Each element x will either be inserted at position $h_1(x)$ in the first hash table or $h_2(x)$ in the second one. When inserting an element x into the i-th $(i \in 1, 2)$ hash table, it first checks whether there has been an element placed in the posision $h_i(x)$. If there has been no element in $h_i(x)$, then x is placed in the posision $h_i(x)$ of the i-th hash table. Otherwise, let y be the element which has been placed in $h_i(x)$ of the i-th hash table, it should try to insert y to the other hash table in the position $h_{3-i}(y)$ after placing x in the posision $h_i(x)$ of the i-th hash table. The inserting operation will not stop until no element needs to be moved. For a new element, the algorithm can place it to the hash tables by executing the above operation starting from inserting it into the first hash table. Cuckoo hashing improves the load factor and query performance of the hash table. Note that the first hash table and the second hash table can be combined to be one hash table [23].

The cuckoo filter [23] designed by Fan et al. expands the cuckoo hash table to a multidimensional structure by adding several slots at each bucket of the hash table. It can significantly reduce the number of the element relocation operations. The cuckoo filter stores the binary fingerprint information of the elements and defines $h_2(x) = h_1(x) \oplus \text{fingerprint}(x)$ to help distribute the items uniformly in the hash table and complete an insertion only using information in the hash table rather than retrieving the original item x. The time complexities of the cuckoo filter for looking up and deleting an element are $O(1)$.

3. Multi-Blockchain-Based Cross-Domain Authentication Scheme

3.1. System Model. We design a cross-domain authentication system model utilizing the multichain structure as shown in Figure 2, which consists of domains, authentication servers, users, and multichain networks.

Domain: we define a domain as a group in which the users trust each other, i.e., Domain A and B in Figure 2. Note that different domains may adopt different cryptographic settings

Authentication Server: the authentication server provides authentication for the users in the domain and can handle authentication requests from within and across

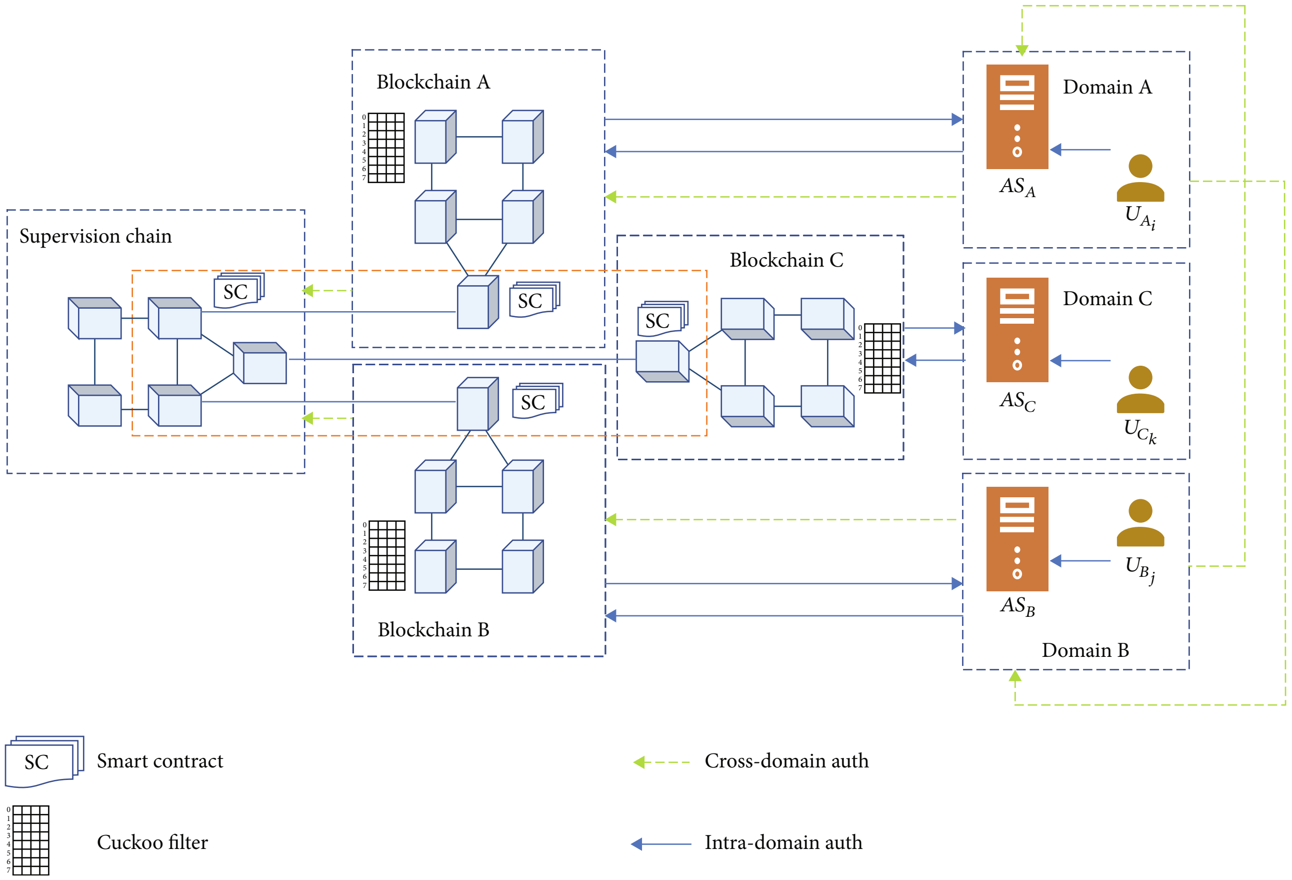


Figure 2: Blockchain based cross-domain authentication system model.

domains with the help of the blockchain. As shown in Figure 2, the authentication servers in domain A and domain B are represented by AS_A and AS_B, respectively

User: there are two roles of the users, i.e., data owner who owns specific resources within a domain and data user who wants to access resources within the same domain or across domains. As shown in Figure 2, the users in domains A and B are represented by U_{A_i} and U_{B_j}

Multichain Networks: each domain deploy a blockchain which stores information of the users used in authentication in the domain. The supervision chain is used to implement cross-domain authentication

3.2. Details of the Scheme. In this section, we describe the details of our scheme in the scenario shown in Figure 2. Specifically, each domain should create its blockchain using Hyperledger Fabric, in which the chaincode of the cuckoo filter should be deployed. Also, the supervision chain should be created which help the cross-domain authentication. Then, each domain chooses a peer in its blockchain as the link peer and join it to the channel of the supervision chain as follows:

Actually, in fabric, joining the link peer to the channel of the supervision chain is adding an orgnization to a channel. Channel in fabric is only an abstract concept rather than a real entity. Therefore, configuring the channel is basically the management of *Channel Configuration*. After a blockchain is successfully deployed, the genesis block files in the folder “channel-artifacts” form the *Channel Configuration*. However, these configuration files are binary files. So, we need to turn them into readable configuration files by executing the following operations in Figure 3. The resulting configuration file is shown on the right of Figure 3.

Then, we can modify these configuration files as follows:

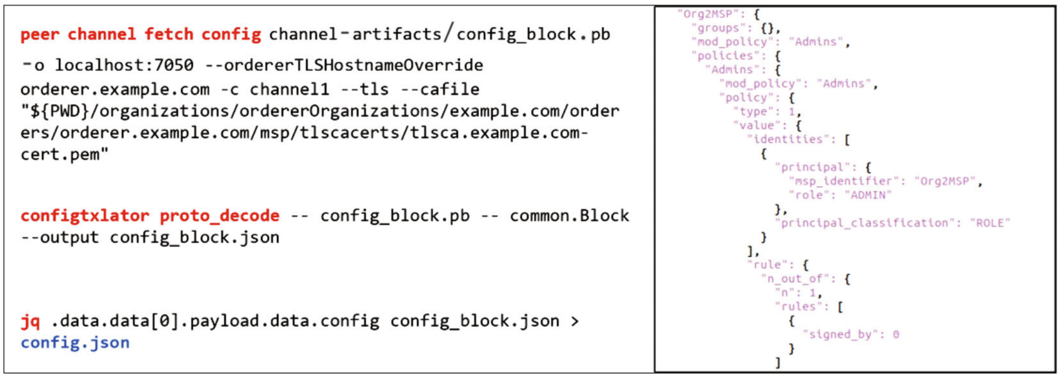
(i) Use the jq command to write the information of the link pear into the above mentioned configuration files

(ii) Use configtxlator proto_encode command packages the original configuration files and the modified configuration files, respectively, to obtain the binary file supported by fabric

(iii) Use configtxlator compute_update command calculates the differences between the original configuration files and the modified configuration files

(iv) Use configtxlator to update the calculated changes to the original configuration files

(v) Each domain and the orgnization of the supervision chain use peer channel signconfigtx to sign the new



```
peer channel fetch config channel-artifacts/config_block.pb
-o localhost:7050 --ordererTLSHostnameOverride
orderer.example.com -c channel1 --tls --cafile
"${PWD}/organizations/ordererOrganizations/example.com/order
ers/orderer.example.com/msp/tlscacerts/tlsca.example.com-
cert.pem"


configtxlator proto_decode -- config_block.pb -- common.Block
--output config_block.json


jq .data.data[0].payload.data.config config_block.json >
config.json
```

```
"Org2MSP": {
  "groups": {},
  "mod_policy": "Admins",
  "policies": {
    "Admins": {
      "mod_policy": "Admins",
      "policy": {
        "type": 1,
        "value": {
          "identities": [
            {
              "principal": {
                "msp_identifier": "Org2MSP",
                "role": "ADMIN"
              },
              "principal_classification": "ROLE"
            }
          ],
          "rule": {
            "n_out_of": {
              "n": 1,
              "rules": [
                {
                  "signed_by": 0
                }
              ]
```

Figure 3: Turn binary files into readable configuration files.

configuration files, respectively. Also, they use peer channel update to enable new configuration files

Now, the link peer is successfully joined into the channel of the supervision chain.

Next, we describe how to deploy the key chaincode about crossing chain in different chains. In fabric, a peer can query and change the world status of other chaincodes in the same channel, while a peer can query the world status of other chaincodes in the different channel rather than changing them. As described above, link peer in each domain's blockchain is joined into the channel of the supervision chain. Therefore, each domain can let the supervision chain obtain the authentication data stored in its blockchain. That is to say, link peer should deploy the chaincode about crossing chain as well as the supervision chain. The multichain structure can be seen in Figure 2.

Algorithm 1 shows the key chaincode about crossing chain. Algorithms 2–5 show the chaincodes about creating the cuckoo filter, insert algorithm, lookup algorithm, and delete algorithm of the cuckoo filter.

At last, we describe the details of our authentication scheme including registration, intra-domain authentication, cross-domain authentication, and revocation.

Registration. For a user U_{A_i}, firstly, registration is done according to the authentication mechanism of the domain which the user belongs to. Then the authentication information in the domain such as user identification U_{A_i}, public key certificate $Cert_{U_{A_i}}$, timestamp, authority identification AS_A, etc. are stored in the blockchain created by the domain. The chaincode of the insert algorithm of the cuckoo filter is called to add the user's identity and authentication information to the filter. At last, the link peer of the blockchain calls the chaincode *SendCrossMessage* deployed in the supervision chain to generate a new transaction record on the supervision chain for storing the authentication information.

Intra-domain authentication. Upon receiving the authentication request from user U_{A_i}, the authentication server AS_A first calls the lookup algorithm of the cuckoo filter to check whether the certificate $Cert_{U_{A_i}}$ exists in the filter, i.e., U_{A_i} is not revoked. Then, AS_A authenticates U_{A_i} according to the authentication mechanism of Domain A.

Cross-domain authentication. If a user U_{A_i} in Domain A wants to be authenticated by the authentication server AS_B in Domain B, they execute an interactive protocol as follows:

(1) U_{A_i} sends the authentication request to AS_B along with its identity and certificate $Cert_{U_{A_i}}$

(2) AS_B checks whether the user U_{A_i} has been revoked by calling the chaincode of the lookup algorithm of the cuckoo filter. If the check passes, go to the next step. Otherwise, the authentication stops

(3) U_{A_i} passes the authentication by AS_A and the authentication information of U_{A_i} is written to the blockchain

(4) AS_B sends the information request to the supervision chain by calling the chaincode *GetCrossMessage* described in Algorithm 1 to get the authentication information of U_{A_i}

(5) AS_B sends a random challenge cha to user U_{A_i}

(6) U_{A_i} uses the private key to sign the challenge and sends the signature $\mathrm{Sig}(\mathrm{cha})$ to AS_B

(7) AS_B verify the signature $\mathrm{Sig}(\mathrm{cha})$ using the public key of U_{A_i}

(8) The authentication is completed

Figure 4 Shows the workflow of the cross-domain authentication.

Revocation. If the authentication server AS_A needs to revoke a user U_{A_i}, it calls the delete algorithm of the cuckoo filter to remove the fingerprint information of U_{A_i}'s certificate, and finally the filter is updated in the blockchain. AS_A also needs to complete the revocation process according to the authentication mechanism of Domain A.

3.3. Security and Performance Analysis

3.3.1. Security. Due to the unforgeability of the blockchain, the user's authentication information on blockchains is hard

```
Procedure (s *SmartContract) GetCrossMessage(ctx contractapi.TransactionContextInterface, channel, chaincode, user1, user2 string) (string, error)
    params := []string{"GetMessage", user1, user2}
    invokeArgs: = make([][]byte, len(params))
    For i, arg: = range params {invokeArgs[i] = []byte(arg)}
    response := ctx.GetStub().InvokeChaincode(chaincode, invokeArgs, channel)
    If response.Status! = shim.OK then
        Return shim.Error(err.Error())
    End if
    users := user1 + ":" + user2
    messages := Messages{Users: users, Message: string(response.Payload)}
    msgJSON, err: = json.Marshal(messages)
    If err! = nil then
        Return shim.Error(err.Error())
    End if
    Err = ctx.GetStub().PutState(users, msgJSON)
    If err! = nil then
        Return shim.Error(err.Error())
    End if
    Return string(response.Payload), nil
End procedure
```

Algorithm 1: The key chaincode about crossing chain.

```
Procedure (s *SmartContract) InitCFilter(ctx contractapi.TransactionContextInterface) string
    var cf CFilter
    configure(&cf)
    cf.Buckets = make([]bucket, cf.Size, cf.Size)
    i := range cf.Buckets
    cf.Buckets[i] = make([]fingerprint, cf.BSize, cf.BSize)
    cfJSON, _: = json.Marshal(&cf)
    err := ctx.GetStub().PutState("cuckfilter", cfJSON)
    If err! = nil then
        Return shim.Error(err.Error())
    End if
    Return string(cfJSON)
End procedure
```

Algorithm 2: The chaincode about creating the cuckoo filter.

to tamper and forge, which effectively ensures the integrity of data and the validity of the user's identity in the cross-domain authentication. In the traditional cross-domain authentication scheme based on PKI, Certificate Authority (CA) is vulnerable to attacks. In our scheme, the user's certificate is stored on the blockchain and cannot be tampered so that it is more resistant to denial of service attacks and other attacks affecting system availability. For the privacy of the user's authentication information stored in the blockchain, we utilize Hyperledger Fabric multiple channels mechanism to separate the information between different channels. Only nodes in the same channel can share the data [24].

3.3.2. Performance. We mainly analyze the time overheads costed when checking whether the user is revoked while obtaining the user's authentication information in the cross-domain authentication. Table 1 Summarizes the basic information of our experiment.

The test blockchain network runs on two hosts belonging to two different channels. Each host contains a total of eight nodes, four MSPs, and three sorting services belonging to four domains and being located in the same channel.

(i) Time overhead of checking whether the user is revoked. This is the execution time of the chaincode of the lookup algorithm of the cuckoo filter. We send 1000 transactions to the blockchain network and observe the changes of average latency and throughput (TPS) of the system under different transaction sending rates as shown in Figure 5

(ii) Time overhead of obtaining the user's authentication information. This is the execution time of the chaincode of *GetCrossMessage* described in Algorithm 1. We send 5000 transactions to the blockchain network and observe the changes of average latency

```
Procedure func (s *SmartContract) Insert(ctx contractapi.TransactionContextInterface, id string, item string) error
  cfJSON, err: = ctx.GetStub().GetState(id)
  If err! = nil then
     Return shim.Error(err.Error())
  End if
  var cf CFilter
  err = json.Unmarshal(cfJSON, &cf)
  If err! = nil then
     Return shim.Error(err.Error())
  End if
  f := fprint([]byte(item), cf.FpSize, Hashfn)
  j: = hashfp([]byte(item))
  k := (j XOR hashfp(f)) % cf.Size
  If cf.Buckets[j].insert(f) || cf.Buckets[k].insert(f) then
     cf.Count++
  Else
     i: = [2]uint{j, k}[rand.Intn(2)]

     For n := uint(0); n<cf.Kicks; n++ do
        f = cf.Buckets[i].swap(f)
        i = (i XOR hashfp(f)) % cf.Size

        If cf.Buckets[i].insert(f) then
           cf.Count++
        End if
     End for
  End if
  cfJSON, err = json.Marshal(&cf)
  If err! = nil then
     Return shim.Error(err.Error())
  End if
  CF_: = CF{id: Id, cfilter: cfJSON}
  err = ctx.GetStub().PutState(cf_.id, cf_.cfilter)
  If err! = nil then
     Return shim.Error(err.Error())
  End if
  Return nil
End procedure
```

Algorithm 3: The chaincode about the insert algorithm of the cuckoo filter.

```
Procedure func (s *SmartContract) Lookup(ctx contractapi.TransactionContextInterface, id string, item string) bool
  cfJSON, err: = ctx.GetStub().GetState(id)
  If err! = nil then
     Return shim.Error(err.Error())
  End if
  var cf CFilter
  err = json.Unmarshal(cfJSON, &cf)
  If err! = nil then
     Return shim.Error(err.Error())
  End if
  f := fprint([]byte(item), cf.FpSize, Hashfn)
  j := hashfp([]byte(item)) % cf.Size
  k := (j XOR hashfp(f)) % cf.Size
  Return cf.Buckets[j].lookup(f) || cf.Buckets[k].lookup(f)
End procedure
```

Algorithm 4: The chaincode about the lookup algorithm of the cuckoo filter.

```
Procedure func (s *SmartContract) Delete(ctx contractapi.TransactionContextInterface, id string, item string) error
  cfJSON, err: = ctx.GetStub().GetState(id)
  If err! = nil then
     Return shim.Error(err.Error())
  End if
  var cf CFilter
  err = json.Unmarshal(cfJSON, &cf)
  If err! = nil then
     Return shim.Error(err.Error())
  End if
  f := fprint([]byte(item), cf.FpSize, Hashfn)
  j := hashfp([]byte(item)) % cf.Size
  k := (j XOR hashfp(f)) % cf.Size
  If cf.Buckets[j].remove(f) || cf.Buckets[k].remove(f) then
     cf.Count–
  End if
  cfJSON, err = json.Marshal(&cf)
  If err! = nil then
     Return shim.Error(err.Error())
  End if
  cf_ := CF{id: id, cfilter: cfJSON}
  err = ctx.GetStub().PutState(cf_.id, cf_.cfilter)
  If err! = nil then
     Return shim.Error(err.Error())
  End if
  Return nil
End procedure
```

Algorithm 5: The chaincode about the delete algorithm of the cuckoo filter.

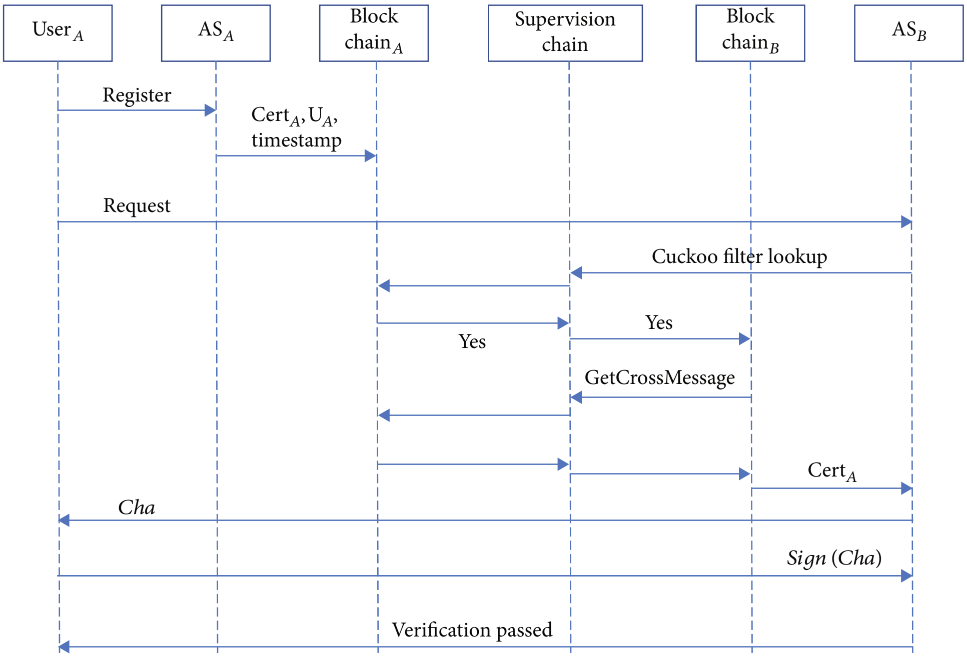


Figure 4: Process of the cross-domain authentication.

Table 1: Basic information.

CPU series	Intel(R) Core(TM) i7-8700
RAM	16.0 GB (15.8 GB is available)
Operating system	Ubuntu 20.04.2 LTS
Software version	Hyperledger fabric 2.4
Environment	Go version: Go 1.16.7 linux/amd64 Docker version 20.10.8

and throughput (TPS) of the system under different transaction sending rates which is shown in Figure 6

4. Cross-Domain Authorization

4.1. The Mapping Based Solution. The key issue that should be solved in cross-domain authorization is how to convert an access control policy in the authorization domain to that in the access domain. In the IoT and Energy Internet scenarios, different domains may apply different access control mechanisms such as role-based access control (RBAC) [25, 26], access control list (ACL), attribute-based access control (ABAC) [27, 28], and capability-based access control (CapBAC) [29]. Moreover, even if the same access control mechanism is used, the access control policy may be different. For example, the roles of two domains applying the RBAC mechanism may be different. To make the authorization by the user's own domain be accepted by other domains, it is necessary to solve the problem of interoperability of access control policies between heterogeneous domains.

A straightforward solution is the mapping-based approach. In the initial phase, mappings are done between different authorization bases (user groups in ACL, roles in RBAC, attributes in ABAC) within domains and the mapping results are recorded in the blockchain. In the following, we briefly describe the access control policy interoperability method based on roles mapping between domains using RBAC mechanism, while the mapping methods between other authorization bases are similar which are omitted.

Role-based access control mechanism includes user set USERS, role set ROLES, and permission set PERMS. The authorization center assigns permissions to the roles, and the users obtain the permissions owned by the roles which he is assigned. Role is the core of RBAC mechanism, so the access control policy interoperability between two domains using RBAC mechanism can be achieved through role mapping. That is, mapping the role in the authorization domain to that in the access domain. The mapping should be one-way. For example, let the role sets in the authorization Domain A and the access Domain B be ROLES_A and ROLES_B, respectively. $\{Ra_1, Ra_2, \cdots\} \mapsto \{Rb_1, Rb_2, \cdots\}$, where $\{Ra_1, Ra_2, \cdots\} \subseteq \mathrm{ROLES}_A$ and $\{Rb_1, Rb_2, \cdots\} \subseteq \mathrm{ROLES}_B$ means that $\{Ra_1, Ra_2, \cdots\}$ is mapped to $\{Rb_1, Rb_2, \cdots\}$ and the mapping results is written to the blockchain. In the process of cross-domain authorization, the user performs identity authentication and obtains the corresponding role according to the above cross-chain authentication process. When a user in Domain A wants to access the resources in the access Domain B, the user's roles $\in \mathrm{ROLES}_A$ can be converted into the roles $\in \mathrm{ROLES}_B$ according to the role mapping, and finally the authorization center of the access Domain B can assign permissions to the user according to his corresponding roles $\in \mathrm{ROLES}_B$.

Although the approach of mapping between different authorization bases can achieve interoperability of access control policies between heterogeneous domains, it requires a lot of overheads of implementing the mapping by authorization centers of the domains in the initial stage. Moreover, the mapping between different authorization bases also needs to consider the tradeoff of access control granularity for all of the domains. To our knowledge, most of the IoT domains currently use RBAC and ABAC mechanisms, so a simpler approach [30] proposed to turn the access control policies in ABAC mechanism into the access control rules in RBAC mechanism can be used for access control policy interoperability when we focus on these two kinds of access control mechanisms. As a result, the access control policies of the domains using ABAC mechanism can be converted into the RBAC rules first, and then only the mapping from roles to roles can be applied globally for RBAC rule's interoperability among various domains. Thus, access control policies interoperability among heterogeneous domains can be achieved for cross-domain access control. However, in general, the existing mapping methods have low practicality in the real scenario with large scale and complex relationships.

4.2. Role and Object Class Based Access Control. Inspired by the capability-based access control (CapBAC) model [31], we propose a cross-domain access control model based on roles and object classes.

Currently, CapBAC model is used in many IoT domains. CapBAC model is an implementation of the access control matrix (ACM) model, while the access control list (ACL) model is another implementation. In the ACL model, each object is associated with an access control list, which records the access permissions of the subjects to it. Conversely, in the CapBAC model, each subject is associated with a list of capabilities (permissions) that record the subject's access permissions to the objects. As shown in Figure 7(a), from the row perspective, each subject is assigned the access rights to the objects, and from a column perspective, it is an ACL model, where O is the subject's access permissions to the object, such as readability and writability, and C is a set of context-aware information, such as time and location.

We assume that the heterogeneous domains in the Energy Internet have the same kinds of resources, such as computing resources, printer, camera, devices in smart grid, various types of energy data, and so on. By this assumption and using the access control matrix, the objects in the domains can be represented in the unified forms. Then, regardless of the access control mechanism used by the domains, we can use the same classes of the objects to link their access control policies. Figure 7(b) shows that the access control matrix in CapBAC model allows for object classification and the assignment of roles to subjects. Also, ACL, RBAC, and ABAC mechanisms can be transformed to be role and object class based mechanisms. For example,

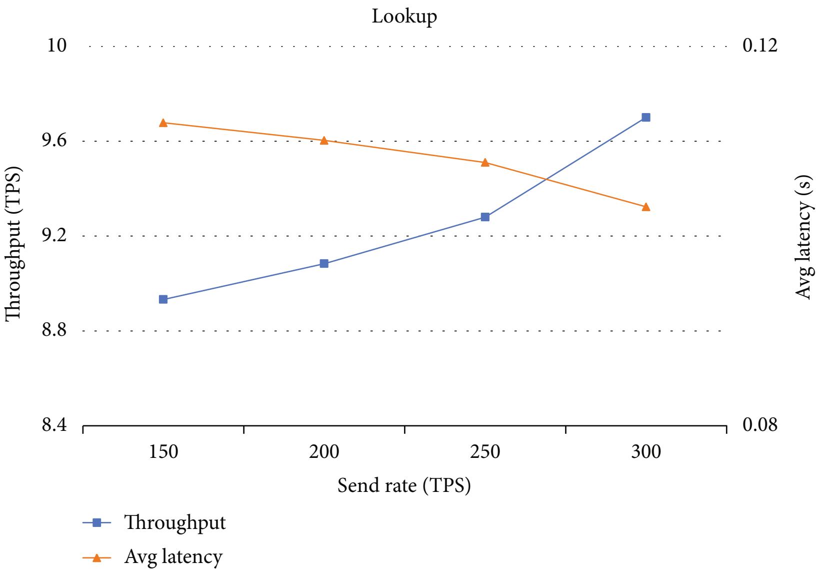


Figure 5: Performance of invoking lookup chainchode of the cuckoo filter.

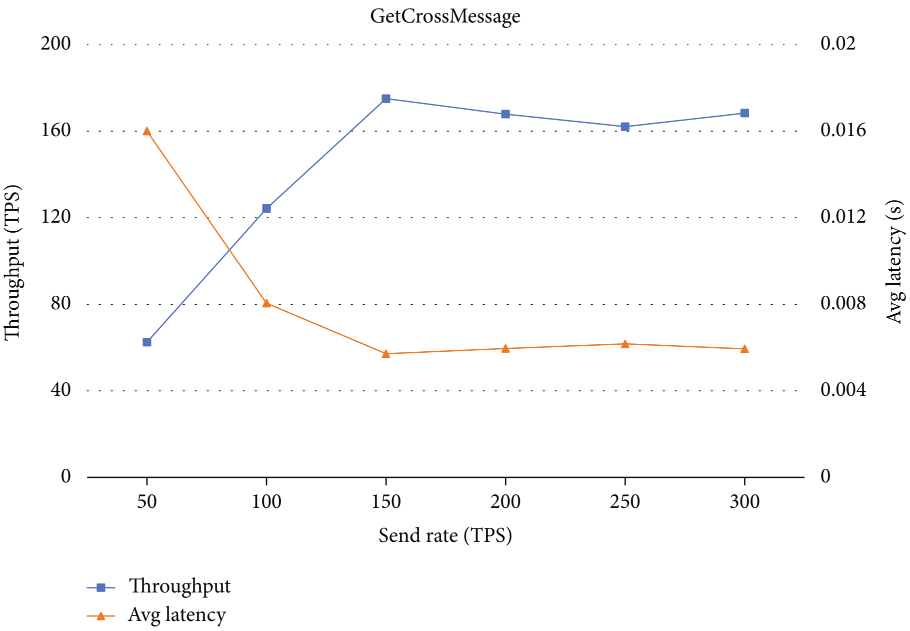
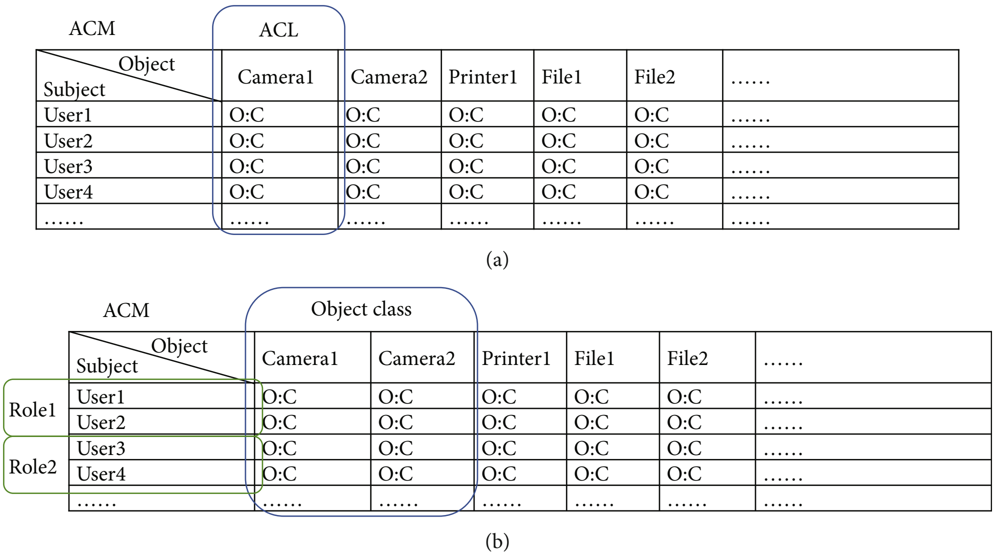


Figure 6: Invoke GetCrossMessage chaincode performance.

in the RBAC model, permissions (PRMS) represent the operational permission for roles to access the resources, which can be divided into operations and objects. The objects can be categorized to some object classes and the role-to-object access control rules can be transformed to be role-to-object class rules. In the ABAC model, both subjects and objects are represented by a set of attributes and the corresponding attribute values. Permissions consist of the object descriptors and operations, and the authorizations are defined between subject descriptors and object descriptors, or consist of the attribute conditions on the subject or object. As described above, ABAC policies can be turned to the RBAC rules, and thus turned to the role-to-object class rules.

Considering the characteristics of various kinds of access control mechanisms, we propose to ensure the interoperability of access control rules based on the object categories. In the Energy Internet scenario, we unify the objects among domains based on the resource types, treating the classes of the resources as the objects to be controlled. Each class of the resources is no longer specifically subdivided. For example, all the cameras will be treated as the same class. As a result, if a user with a role in Domain *A* can access the Camera class, he can access the cameras in Domain *B* after the authentication and authorization in Domain *A*. (Note that for some resources with special access control requirements, access control can be performed independently). In summary, we propose a cross-domain authorization mechanism based on blockchain, roles


ACM
ACL

Object / Subject	Camera1	Camera2	Printer1	File1	File2	……
User1	O:C	O:C	O:C	O:C	O:C	……
User2	O:C	O:C	O:C	O:C	O:C	……
User3	O:C	O:C	O:C	O:C	O:C	……
User4	O:C	O:C	O:C	O:C	O:C	……
……	……	……	……	……	……	……

(a)

ACM
Object class
Role1
Role2

Object / Subject	Camera1	Camera2	Printer1	File1	File2	……
User1	O:C	O:C	O:C	O:C	O:C	……
User2	O:C	O:C	O:C	O:C	O:C	……
User3	O:C	O:C	O:C	O:C	O:C	……
User4	O:C	O:C	O:C	O:C	O:C	……
……	……	……	……	……	……	……

(b)

Figure 7: From CapBAC to role and object class.

ACM

Object class / Role	Camera	Tablet	Printer	FileClass1	FileClass2	……
Role1	O:C	O:C	O:C	O:C	O:C	……
Role2	O:C	O:C	O:C	O:C	O:C	……
Role3	O:C	O:C	O:C	O:C	O:C	……
Role4	O:C	O:C	O:C	O:C	O:C	……
……	……	……	……	……	……	……

Figure 8: Example of role and object class based access control model.

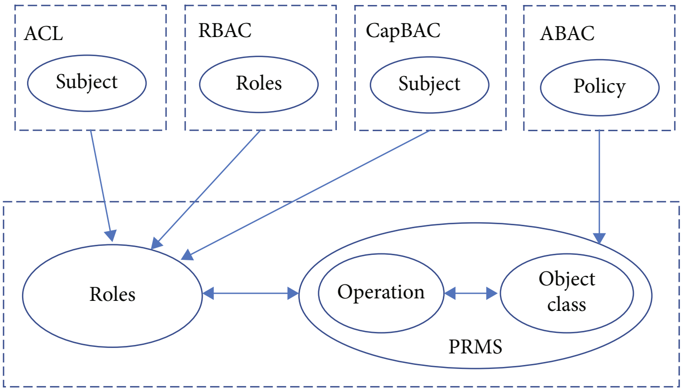


Figure 9: Role and object class based cross-domain authorization.

and object categories. In the initial phase, the authorization centers of each domain negotiate the object classes and the role classes according to the access control methods in their respective domains. First, the objects are categorized to various classes which is a many-to-many mapping, i.e., an object can be regarded as different classes. Then the object classes are authorized to the negotiated roles, and the authorization rules are written to the blockchain. As shown in Figure 8, this is essentially an access control mechanism based on roles and controlled object classes.

To make cross-domain authorization, the access control policies of each heterogeneous domain can be converted into RBAC access control rules using the mapping methods described above. As shown in Figure 9, users can then be granted permissions based on the role-object class based authorization rules negotiated and written to the blockchain.

In the following, we describe the details of our cross-domain authorization scheme including initial phase, registration, intra-domain authorization, and cross-domain authorization.

Initial phase. As described above, the authorization centers of each domain negotiate the object classes, the role classes according to the access control methods in their respective domains, and transform their own access control policies to the role and object class based rules. The transformation information is written to the blockchain.

Registration. When a user registers in its own domain, the role assignment procedure is added at the end of the registration phase of the authentication process described in the Section III to complete the registration.

Intra-domain authorization. Upon receiving the authorization request from user U_{A_i}, the authentication server AS_A first calls the lookup algorithm of the cuckoo filter to check whether the certificate $Cert_{U_{A_i}}$ exists in the filter, i.e., U_{A_i} is not revoked. Then, AS_A authorizes U_{A_i} according to the access control mechanism of Domain A.

Cross-domain authorization. If a user U_{A_i} in Domain A wants to be authorized by the authorization server AS_B in Domain B, they execute an interactive protocol as follows:

(1) U_{A_i} and AS_B first complete the cross-domain authentication process as described above

(2) AS_B obtains the access control rules for U_{A_i} by coverting the role and object class based access control rules in the cross-domain authorization token to that of its own access control mechanism according to the transformation in the initial phase

(3) The authorization is completed

5. Conclusion

In this paper, based on the cross-chain technology of the blockchain and the cuckoo filter, we propose a multi-blockchain-based cross-domain authentication scheme supporting heterogeneous domains for the Energy Internet. In our scheme, a cross-chain authentication architecture is established and the users coming from different domains can independently perform the authentication with the help of the deployed chaincodes. The cuckoo filter is deployed on the blockchain as chaincodes and used to effectively manage the user certificate for the user registration, query, and revocation. We analyze the security of our scheme and design experiments to analyze the performance of our scheme such as the costs of user revocation and cross-domain authentication. Experimental results show the effectiveness and feasibility of our scheme. Basd on the authentication scheme, we propose a cross-domain access control model based on roles and object classes, by assuming that the heterogeneous domains have the same kinds of resources, treating the object classes as controlled objects and then applying the role-based access control to the object classes. It should be pointed out that we only implemented our scheme on the Hyperledger Fabric platform, and we leave it a future work to implement our scheme on other blockchain platforms. Besides, we will try to construct cross-domain authentication and authorization schemes based on the trusted execution environment.

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 there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the State Grid Shandong Electric Power Company Science and Technology Project funded “Research on Smart IoT Active Security Protection and Simulation and Detection Technology for Energy Internet” (5206002000V5).

References

[1] T. Bei, D. Shi, F. Li, B. Chen, and Y. Yuan, “Research on security assessment scheme to smart grid based on digital twin,” *Shandong Electric Power*, vol. 49, pp. 25–30, 2022.

[2] A. Shamir, “Identity-based cryptosystems and signature schemes,” in *Advances in Cryptology*, CRYPTO 1984, G. R. Blakley and D. Chaum, Eds., pp. 47–53, Springer, Berlin, Heidelberg, 1985.

[3] D. Zhang, J. Le, X. Lei, T. Xiang, and X. Liao, “Exploring the redaction mechanisms of mutable blockchains: a comprehensive survey,” *International Journal of Intelligent Systems*, vol. 36, no. 9, pp. 5051–5084, 2021.

[4] N. Szabo, “Formalizing and securing relationships on public networks,” *First Monday*, vol. 2, no. 9, 1997.

[5] B. Vaidya, D. Makrakis, and H. T. Mouftah, “Security mechanism for multi-domain vehicle-to-grid infrastructure,” in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pp. 1–5, Houston, TX, USA, 2011.

[6] W. Libing, J. Wang, K.-K. R. Choo, Y. Li, and D. He, “An efficient provably-secure identity-based authentication scheme using bilinear pairings for ad hoc network,” *Journal of Information Security and Applications*, vol. 37, pp. 112–121, 2017.

[7] X. Cao, W. Kou, and D. Xiaoni, “A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges,” *Information Sciences*, vol. 180, no. 15, pp. 2895–2903, 2010.

[8] S. Benzarti and B. Triki, “Drone authentication using id-based signcryption in LoRaWAN network,” in *Intelligent Systems Design and Applications*, ISDA 2019. Advances in Intelligent Systems and Computing, P. Siarry, K. Ma, and A. Kaklauskas, Eds., pp. 205–216, Springer, Cham, 2021.

[9] M. Shen, H. Liu, L. Zhu et al., “Blockchain-assisted secure device authentication for cross-domain industrial iot,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 942–954, 2020.

[10] C. Fromknecht, D. Velicanu, and S. Yakoubov, “A decentralized public key infrastructure with identity retention,” Cryptology ePrint Archive, 2014, Report 2014/803.

[11] A. Ouaddah, A. A. El Kalam, and A. A. Ouahman, “Fairaccess: a new blockchain-based access control framework for the internet of things,” *Security and cCommunication Networks*, vol. 9, no. 18, pp. 5943–5964, 2016.

[12] W. Wang, N. Hu, and X. Liu, “Blockcam: a blockchain-based cross-domain authentication model,” in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 896–901, Guangzhou, China, 2018.

[13] H. Wang, S. Wanyan, G. Shuxian, and Y. Jia, "Research on blockchain technology in power system," *Shandong Electric Power*, vol. 46, pp. 8–12, 2019.

[14] Y. Yao, X. Chang, J. Misic, V. B. Misic, and L. Li, "BLA: blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3775–3784, 2019.

[15] S. Guo, F. Wang, N. Zhang, F. Qi, and X. Qiu, "Master-slave chain based trusted cross-domain authentication mechanism in IoT," *Journal of Network and Computer Applications*, vol. 172, article 102812, 2020.

[16] H. Zhang, X. Chen, X. Lan, H. Jin, and Q. Cao, "BTCAS: a blockchain-based thoroughly cross-domain authentication scheme," *Journal of Information Security and Applications*, vol. 55, article 102538, 2020.

[17] Y. Hao and L. Qing, "Block chain based distributed resources participating in power market transactions under ubiquitous power internet of things," *Shandong Electric Power*, vol. 47, pp. 42–48, 2020.

[18] G. Ali, N. Ahmad, Y. Cao et al., "xDBAuth: Blockchain based cross domain authentication and authorization framework for internet of things," *IEEE Access*, vol. 8, pp. 58800–58816, 2020.

[19] X. Zhu, J. Zheng, B. Ren, X. Dong, and Y. Shen, "Microthingschain: blockchain-based controlled data sharing platform in multi-domain iot," *Journal of Networking and Network Applications*, vol. 1, no. 1, pp. 19–27, 2021.

[20] X. Ma, W. Ma, and X. Liu, "A cross domain authentication scheme based on blockchain technology," *Acta Electronica Sinica*, vol. 46, no. 11, pp. 2571–2579, 2018.

[21] "Hyperledger fabric," https://www.hyperledger.org/use/fabric.

[22] R. Pagh and F. F. Rodler, "Cuckoo hashing," *Journal of Algorithms*, vol. 51, no. 2, pp. 122–144, 2004.

[23] B. Fan, D. G. Andersen, M. Kaminsky, and M. D. Mitzenmacher, "Cuckoo filter: practically better than bloom," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pp. 75–88, New York, NY, USA, 2014.

[24] C. Ma, X. Kong, Q. Lan, and Z. Zhou, "The privacy protection mechanism of hyperledger fabric and its application in supply chain finance," *Cybersecur*, vol. 2, 2019.

[25] B. Gwak, J.-H. Cho, D. Lee, and H. Son, "TARAS: Trust-aware role-based access control system in public internet-of-things," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 74–85, New York, NY, USA, 2018.

[26] M. Amoon, T. Altameem, and A. Altameem, "RRAC: role based reputed access control method for mitigating malicious impact in intelligent IoT platforms," *Computer Communications*, vol. 151, pp. 238–246, 2020.

[27] H. Nasiraee and M. Ashouri-Talouki, "Anonymous decentralized attribute-based access control for cloud-assisted IoT," *Future Generation Computer Systems*, vol. 110, pp. 45–56, 2020.

[28] M. Gupta, F. M. Awaysheh, J. Benson, M. Alazab, F. Patwa, and R. Sandhu, "An attribute-based access control for cloud enabled industrial smart vehicles," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4288–4297, 2021.

[29] Y. Nakamura, Y. Zhang, M. Sasabe, and S. Kasahara, "Capability-based access control for the internet of things: an ethereum blockchain-based scheme," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Waikoloa, HI, USA, 2019.

[30] G. Batra, V. Atluri, J. Vaidya, and S. Sural, "Enabling the deployment of ABAC policies in RBAC systems," in *Data and Applications Security and Privacy XXXII. DBSec 2018*, Lecture Notes in Computer Science, F. Kerschbaum and S. Paraboschi, Eds., pp. 51–68, Springer, Cham, 2018.

[31] Y. Liu, L. Qinghua, S. Chen et al., "Capability-based IoT access control using blockchain," *Digital Communications and Networks*, vol. 7, no. 4, pp. 463–469, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 6086686, 10 pages
https://doi.org/10.1155/2023/6086686

Research Article

Security Authentication Protocol for Massive Machine Type Communication in 5G Networks

Junfeng Miao,[1] **Zhaoshun Wang**,[1] **Mei Wang**,[2] **Xiao Feng**,[3,4] **Nan Xiao**,[1] **and Xiaoxue Sun**[1]

[1]*The School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijjing 100083, China*
[2]*Shandong University, Shandong 250100, China*
[3]*Beijing University of Posts and Telecommunications, Beijing 100876, China*
[4]*State Grid Information Telecommunication Group Co.Ltd, Beijing 102211, China*

Correspondence should be addressed to Xiao Feng; fengxiao@bupt.edu.cn

Received 5 July 2022; Revised 29 September 2022; Accepted 24 November 2022; Published 6 April 2023

Academic Editor: Chenglu Jin

As one of the three major applications of 5G, massive machine type communication (mMTC) is mainly oriented to network access scenarios for massive devices. mMTC focuses on solving the problem that traditional mobile communication cannot well support the Internet of Things and vertical industry applications. According to the current 3GPP standard, these massive devices still use the traditional authentication process to realize mutual authentication with 5G core network, which brings a lot of communication and computing overhead. In addition, privacy protection will also be threatened in the authentication process. In order to alleviate the signaling congestion during authentication and solve the insecurity in authentication, this paper proposes a group authentication scheme for mMTC. Due to the characteristics of low power consumption and massive connection, the scheme mainly adopts lightweight encryption operation to avoid the computational burden of equipment and server. We verify the security of our scheme by using BAN logic to formally analyze the scheme. Then, through informal analysis, our proposed scheme can not only avoid signaling blocking and provide mutual authentication but also resist various possible attacks. Through performance evaluation, it is proved that our scheme has better efficiency.

1. Introduction

With the deepening of 5G technology research, ITU-R formally defined massive machine type communication (mMTC) as one of the three major 5G application scenarios in 2015 [1]. With its huge advantages over 4G in performance indicators such as peak rate, air interface delay, and spectrum resources, 5G can meet hundreds of millions of massive IoT terminal network performance requirements, promote the deep integration of 5G and IoT, and form a mMTC business scenario [2]. From the concept definition of mMTC, hundreds of millions of terminal devices are deployed and applied to the needs of massive data acquisition and transmission [3]. Massive connections and small amount of data are one of the main characteristics of the typical mMTC mode. At the same time, it has the advantages of 5G network high speed, low delay, and other network performance advantages [4].

In the mMTC business scenario, a large number of terminal devices, 5G key technologies, etc., meet the needs of digital and diversified business in terms of coverage, number of devices, and network performance [5]. At the same time, it also brings network security challenges to the mMTC business scenario. The mMTC business scenario introduces 5G key technologies such as virtualization and network slicing to drive the mMTC business scenario network to a virtualized and service-oriented transition. At the same time, in the ubiquitous connection scenario, a large number of diversified terminals are easy to be used by attacks, and they lead to the threat of network attacks [6].

As a typical application scenario under the 5G Internet of Things architecture, mMTC has become the focus of many researchers and the cornerstone of building a global Internet of Things to realize the interconnection of all things. mMTC is mainly aimed at the Internet of Things aiming at sensing and data acquisition. Its goal is simply to enable more machine type communication user equipment to connect to the network. The Third Generation Partnership Project (3GPP) defines the secure access process of mMTC device [7, 8]. However, it also faces many problems. Firstly, there are too many information of header transmitted between MTC device and base station in the process of random access, resulting in low data transmission efficiency. Secondly, the number of MTC devices is much larger than the number of time-frequency resources that the system can provide. The serious mismatch between the two will lead to serious equipment access collision and increase the access delay of MTC devices and excessive access energy consumption. Therefore, it is necessary to reduce the signaling interaction in the random-access process and the average delay in the access process and then improve the utilization efficiency of time-frequency resources and the data transmission efficiency of MTC device. From LTE network to 5G networks, the number of users increases exponentially. But in mMTC communication scenario, the secure access scheme still adopts 3GPP standard authentication protocol and key agreement (EAP-AKA) [9]. Therefore, when the mMTC device roams to the 5G network, serious signaling congestion and security issues may occur [10]. The inspiration of this paper is based on [11–17], which proposes a lightweight security authentication protocol based on Barrel Shifter Physical Unclonable Function (BS-PUF) for mMTC in 5G network. The protocol allows the service network to authenticate a group of devices at the same time, so as to reduce the number of signaling transmission and communication delay through the home network. The main contributions of this paper are as follows:

(1) Under the background that 5G networks have a large number of MTC devices, in order to reduce the computation overhead and communication delay, we aggregate the authentication messages on leader MTC into a message and send it to the server for authentication, which improves the authentication efficiency

(2) We propose a lightweight security authentication scheme. Our scheme is based on lightweight encryption primitives

(3) Here, we first use BAN logic to verify the correctness and safety of the scheme. Then, we use informal security analysis to analyze the related security requirements achieved by our scheme and compare it with the security functions of other related schemes later

(4) Finally, in the performance evaluation, we analyze that our scheme has less computation overhead and communication overhead. Therefore, our scheme has good security and efficiency in the process of mMTC device authentication

The remaining chapters of our article are listed below. In Chapter 2, we review related research work. In Chapter 3, we mainly introduce the relevant knowledge of the scheme. In Chapter 4, we mainly describe our proposed scheme in detail. In Chapter 5, we prove and analyze the security of the scheme. In Chapter 6, we evaluated the performance of the solution. Finally, in Chapter 7, we summarize the work of the full text.

2. Related Work

So far, many researchers have proposed a lot of research on group MTC authentication in LTE networks. With the continuous development and popularization of 5G network, many scholars also put forward the research on group MTC authentication for 5G network.

In [18], Lai et al. proposed a lightweight group authentication protocol based on aggregated messages in LTE networks. This protocol performed group authentication on MTC devices, reduced the overhead of identity verification, and effectively avoided signaling congestion in the network. Cao et al. [19] proposed a group-based access authentication scheme using aggregated signature technology. This scheme could enable a large number of MTC devices to be authenticated by the network and establish corresponding session keys, respectively. Zhang et al. [20] proposed a group-based security authentication protocol in roaming scenarios. The protocol had a dynamic group key generation and update method, and it also avoided the blockage caused by a large number of MTC devices. Cao et al. [21] proposed an efficient group-based anonymous handover protocol. The protocol could adapt to roaming scenarios in LTE-A networks and could effectively reduce signaling costs and communication costs and protect user privacy. Li et al. [22] proposed an identity verification and key agreement scheme based on a secret sharing scheme in MTC scenarios. This scheme realized distributed authentication and dynamically updated access strategy. Cao et al. [23] proposed a secure and efficient authentication scheme based on multisignature and aggregated message authentication code technology. This solution could implement a simple authentication process and switch between different scenarios and had relatively good security. These schemes were mainly for LTE networks.

Cao et al. [11] proposed a group-based handover authentication and reauthentication protocol in 5G networks. This protocol was suitable for mMTC devices roaming to a new network, and the signaling overhead and bandwidth consumption were less than other protocols.

Basudan [12] proposed a lightweight and efficient mMTC group authentication protocol in 5G networks. The protocol was based on bilinear mapping and aggregation without certificates and realized mutual authentication, session keys, and confidentiality. Cao et al. [13] proposed a secure and efficient authentication scheme for a large number of devices in 5G networks. This scheme could not only resist a large number of protocol attacks but also could update group members and realize privacy protection. Lai et al. [9] proposed a group-based secure lightweight authentication and key protocol for machine-to-machine communication. The scheme could resist various attacks and provide the required security requirements. Cao et al. [14] proposed a lightweight and secure access authentication protocol based on extended chaotic mapping. This protocol was aimed at two types of equipment. One was ordinary user equipment, and the other was mMTC equipment. And the protocol implemented functions such as mutual authentication and anonymity protection. These schemes were mainly for 5G networks, but some schemes had large computation and communication overhead.

3. Preliminaries

3.1. System Model. As shown in Figure 1, the system model mainly includes 5G access network and 5G core network [4, 14, 24].

The 5G access network is mainly composed of MTC devices and wireless networks. The wireless network includes 5G next-generation radio access network (NG-RAN) and non-3GPP access network, which provide with data network access and communication services for devices. In 5G core network, access and mobility management function (AMF) can provide all functions related to users and control plane session management and can authenticate through security anchor function (SEAF). Authentication Server Function (AUSF) and Unified Data Management (UDM) provide authentication and user data management services for users. When connecting to the network through NG-RAN, the user authenticates with AUSF through SEAF/AMF. When connecting to the network through non-3GPP access network, the user establishes a security association through IKEv2 (Internet Key Exchange Protocol version 2) in the non-3GPP access interworking function (N3IWF) and then performs the authentication process through AMF/AUSF. In addition, 5G core network also provides session management function (SMF) and user plane function (UPF).

3.2. Security Model. The protocol security analysis method mainly focuses on whether there are loopholes in protocol interaction, that is, the Dolev-Yao model [25]. In the Dolev-Yao model, Dolev and Yao believe that the knowledge and capabilities of protocol attackers cannot be ignored in protocol security certification. The specific capabilities are as follows:

(1) The attacker can control the whole communication channel

(2) Attackers can establish connections with devices and execute security authentication and key agreement protocols by constructing masquerade nodes

(3) Attackers can eavesdrop, store, forge, modify, and replay messages transmitted on the channel

3.3. Security Requirements. In order to eliminate possible security threats and ensure that mMTC devices can communicate securely, the authentication protocol we designed should meet the following security goals:

(1) *Identity Authentication*. The communication entities authenticate each other to ensure the legitimacy of the authentication entities

(2) *Session Key Security*. The communication entity negotiates the secure session key, and the attacker cannot obtain the session key

(3) *Identity Anonymity and Unlinkability*. In the whole authentication process, the user identity information must be hidden, and the attacker cannot associate its identity information with the public information of the channel

(4) *Forward Security*. This goal ensures that even if the session key is leaked, the previous session key cannot be calculated from the key, which is irrelevant to each other. The security of session key is guaranteed

(5) *Antiattack Ability*. The proposed scheme can resist existing protocol attacks, including replay attack and forgery attack

(6) *Avoid Authentication Signaling Congestion*. When a large number of users make access requests at the same time, it can simplify the authentication process, reduce the authentication delay, avoid signaling congestion, and finally ensure the smooth progress of the whole authentication system

3.4. Barrel Shifter Physical Unclonable Function. Physical Unclonable Function (PUF) is a group of miniature delay circuits, which extracts the differences in the chip manufacturing process to obtain a group of input and output called stimulus-response pairs. The relationship between stimulus and response is only determined by certain physical differences in the device. Due to the differences in the chip manufacturing process, it has a non-reproducible characteristic [15].

In 2018, Guo et al. [16] proposed a Barrel Shifter Physical Unclonable Function (BS-PUF) based on reversible and commutativity. It is defined as follows:

Property 1: reversible

Given a reversible keyed PUF, the value x and the key K, calculate $\mathrm{PUF}(K, x) = y \Rightarrow \mathrm{PUF}^{-1}(K, y) = x$, where PUF^{-1} is the reverse calculation on the same PUF.

Property 2: commutativity

Given two commutative PUF_1 and PUF_2, for BS-PUF, such the commutative PUF not only has logical commutativity but also physical commutativity, so $\mathrm{PUF}_1(\mathrm{PUF}_2(x)) = \mathrm{PUF}_2(\mathrm{PUF}_1(x))$ can be calculated.

4. Proposed Scheme

Based on research [11–23], this paper proposes a lightweight security authentication scheme. This solution enables the mMTC devices to communicate securely through the session key in the 5G networks. Table 1 lists the main notations used here.

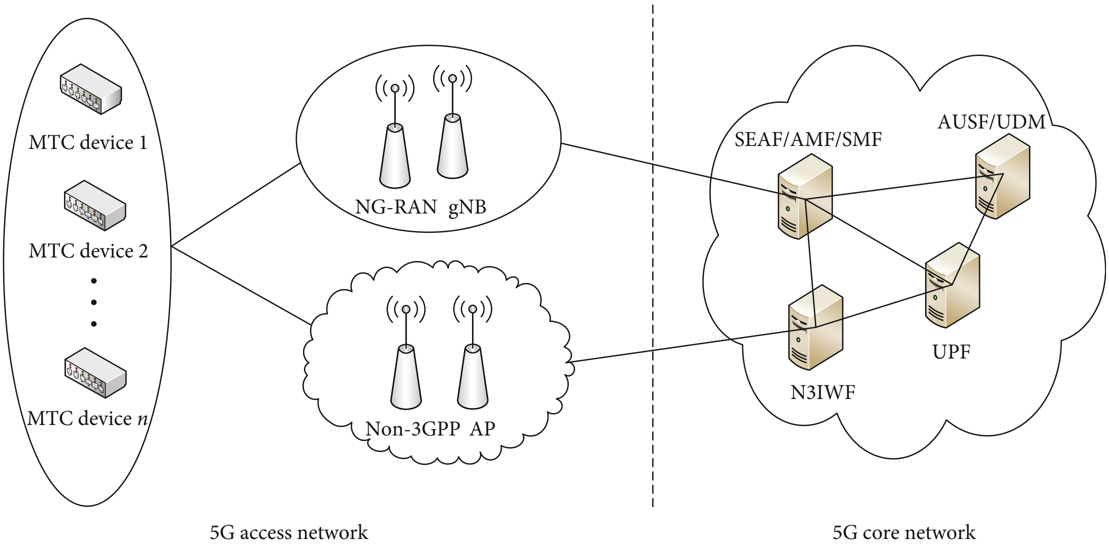


Figure 1: System model.

Table 1: Notations.

Notations	Definitions
TRC	The trusted registration center
MTC_i	Machine type communication device
AUSF/UDM	Authentication server function/unified data management
SEAF/AMF	Security anchor function/access and mobility management function
$H(\bullet)$	A one-way secure hash function
$\mathrm{PUF}_i(\bullet)$	Physical Unclonable Function
$\oplus$	Exclusive-OR operation
$\|\|$	Concatenation operation
T_x	The timestamp
GID	The group identity
TID_i	The temporary identity
s	The system master key
MAC_x	Message authentication code

4.1. System Setup. In order to better design the access authentication protocol for mMTC device and facilitate the security analysis of the protocol, in the scheme, it is assumed that each user device and 5GC network node can perform BS-PUF. In this initialization phase, the trusted registration center (TRC) is a trusted entity responsible for registering MTC device. TRC selects the master key $s \in Z_q^*$ and a one-way secure hash function $H : \{0,1\}^* \longrightarrow Z_q^*$. Then, TRC publishes system parameters $\{H(\bullet)\}$. Here, we merge TRC and AUSF/UDM. Each MTC device first registers with TRC and returns relevant parameters to the user device through the secure channel. According to the Diameter protocol [4] formulated by 3GPP organization, it can be seen that the communication between AUSF/UDM and SEAF/AMF uses the wired channel between backbone networks for transmission. Therefore, we believe that the communication channel between AUSF/UDM and SEAF/AMF is safe. In addition, for mMTC devices in the same range, we select a device leader MTC_n based on the functions of the mMTC device including computing capabilities and communication capabilities. As shown in Figure 2, it shows the specific authentication details of our scheme.

4.2. Registration. In the registration phase, each device MTC_i registers with the TRC through a secure channel. Firstly, MTC_i randomly selects a random value X_i, calculates $\mathrm{PK}_{\mathrm{MTC}_i} = \mathrm{PUF}_{\mathrm{MTC}_i}(X_i)$, and then sends the identity ID_i, X_i, and $\mathrm{PK}_{\mathrm{MTC}_i}$ to TRC through the secure channel. When TRC receives the values sent by MTCD_i, it randomly selects the value e_i, calculates the temporary identity $\mathrm{TID}_i = \mathrm{PUF}_{\mathrm{TRC}}(s, e_i)$, $\mathrm{PK}_i = \mathrm{PUF}_{\mathrm{TRC}}(X_i)$, $A_i{=}H(s, e_i)$, stores $(\mathrm{ID}_i$ $,\mathrm{PK}_{\mathrm{MTC}_i})$ in the database, and then sends the message $(\mathrm{TID}_i$, PK_i, and $A_i)$ to MTCD_i through the secure channel.

4.3. Access Authentication

(1) First, the device MTC_i in the group generates a random number X_i^{new}; calculates the secret value $K_{\mathrm{MTC}_i} = \mathrm{PUF}_{\mathrm{MTC}_i}(\mathrm{PK}_i)$, $\mathrm{PK}_{\mathrm{MTC}_i}^{\mathrm{new}} = \mathrm{PUF}_{\mathrm{MTC}_i}(X_i^{\mathrm{new}})$, $\mathrm{HID}_i = A_i \oplus \mathrm{ID}_i$, and $M_{\mathrm{MTC}_i} = (H(A_i, \mathrm{ID}_i) \| K_{\mathrm{MTC}_i}) \oplus (\mathrm{PK}_{\mathrm{MTC}_i}^{\mathrm{new}} \| X_i^{\mathrm{new}})$; and generates a verification message $\mathrm{MAC}_i = H(\mathrm{ID}_i, \mathrm{TID}_i, K_{\mathrm{MTC}_i}, \mathrm{PK}_{\mathrm{MTC}_i}^{\mathrm{new}}, X_i^{\mathrm{new}})$. Then, MTC_i sends the message $\{\mathrm{TID}_i, \mathrm{HID}_i, M_{\mathrm{MTC}_i}, \mathrm{MAC}_i\}$ to MTC_n

(2) Upon receiving the messages sent by the group members, MTC_n performs the same operation as MTC_i. And it generates the current timestamp T_{MTC_n} and the corresponding group identity GID and calculates $\mathrm{TGID} = \mathrm{GID} \oplus H(A_n, \mathrm{ID}_n, K_{\mathrm{MTC}_n})$ and $\mathrm{MAC}_L = H($ $\mathrm{MAC}_1, \mathrm{MAC}_2 \bullet\bullet\bullet, \mathrm{MAC}_n, T_{\mathrm{MTC}_n}, \mathrm{GID})$. Finally, MTC_n

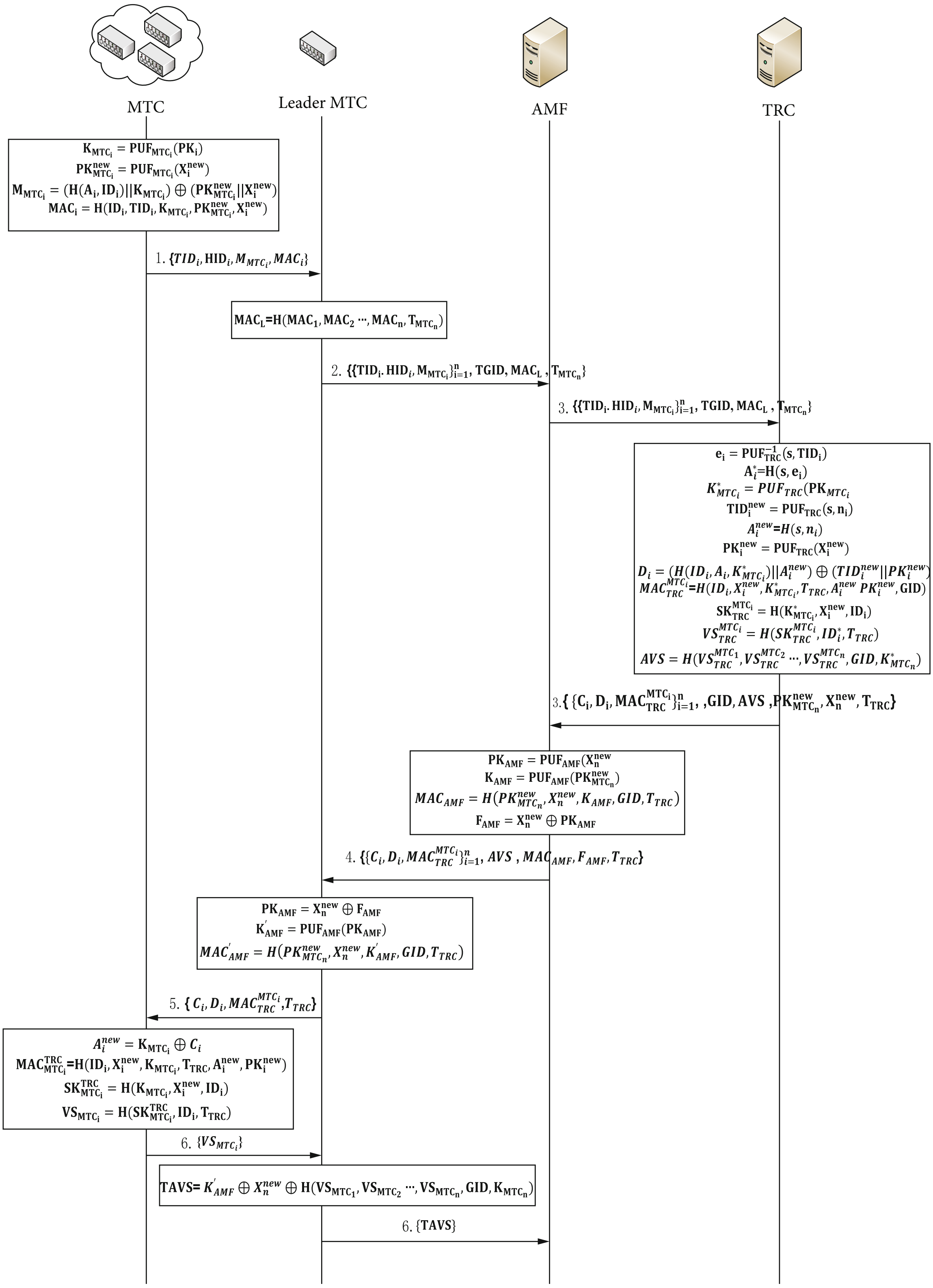

MTC
Leader MTC
AMF
TRC
$K_{MTC_i} = PUF_{MTC_i}(PK_i)$
$PK^{new}_{MTC_i} = PUF_{MTC_i}(X^{new}_i)$
$M_{MTC_i} = (H(A_i, ID_i)||K_{MTC_i}) \oplus (PK^{new}_{MTC_i}||X^{new}_i)$
$MAC_i = H(ID_i, TID_i, K_{MTC_i}, PK^{new}_{MTC_i}, X^{new}_i)$
1. $\{TID_i, HID_i, M_{MTC_i}, MAC_i\}$
$MAC_L = H(MAC_1, MAC_2 \cdots, MAC_n, T_{MTC_n})$
2. $\{\{TID_i. HID_i, M_{MTC_i}\}^n_{i=1}, TGID, MAC_L , T_{MTC_n}\}$
3. $\{\{TID_i. HID_i, M_{MTC_i}\}^n_{i=1}, TGID, MAC_L , T_{MTC_n}\}$
$e_i = PUF^{-1}_{TRC}(s, TID_i)$
$A^*_i = H(s, e_i)$
$K^*_{MTC_i} = PUF_{TRC}(PK_{MTC_i}$
$TID^{new}_i = PUF_{TRC}(s, n_i)$
$A^{new}_i = H(s, n_i)$
$PK^{new}_i = PUF_{TRC}(X^{new}_i)$
$D_i = (H(ID_i, A_i, K^*_{MTC_i})||A^{new}_i) \oplus (TID^{new}_i||PK^{new}_i)$
$MAC^{MTC_i}_{TRC} = H(ID_i, X^{new}_i, K^*_{MTC_i}, T_{TRC}, A^{new}_i\ PK^{new}_i, GID)$
$SK^{MTC_i}_{TRC} = H(K^*_{MTC_i}, X^{new}_i, ID_i)$
$VS^{MTC_i}_{TRC} = H(SK^{MTC_i}_{TRC}, ID^*_i, T_{TRC})$
$AVS = H(VS^{MTC_1}_{TRC}, VS^{MTC_2}_{TRC} \cdots, VS^{MTC_n}_{TRC}, GID, K^*_{MTC_n})$
3. $\{\{C_i, D_i, MAC^{MTC_i}_{TRC}\}^n_{i=1}, ,GID, AVS ,PK^{new}_{MTC_n}, X^{new}_n, T_{TRC}\}$
$PK_{AMF} = PUF_{AMF}(X^{new}_n$
$K_{AMF} = PUF_{AMF}(PK^{new}_{MTC_n})$
$MAC_{AMF} = H(PK^{new}_{MTC_n}, X^{new}_n, K_{AMF}, GID, T_{TRC})$
$F_{AMF} = X^{new}_n \oplus PK_{AMF}$
4. $\{\{C_i, D_i, MAC^{MTC_i}_{TRC}\}^n_{i=1}, AVS , MAC_{AMF}, F_{AMF}, T_{TRC}\}$
$PK_{AMF} = X^{new}_n \oplus F_{AMF}$
$K'_{AMF} = PUF_{AMF}(PK_{AMF})$
$MAC'_{AMF} = H(PK^{new}_{MTC_n}, X^{new}_n, K'_{AMF}, GID, T_{TRC})$
5. $\{C_i, D_i, MAC^{MTC_i}_{TRC}, T_{TRC}\}$
$A^{new}_i = K_{MTC_i} \oplus C_i$
$MAC^{TRC}_{MTC_i} = H(ID_i, X^{new}_i, K_{MTC_i}, T_{TRC}, A^{new}_i, PK^{new}_i)$
$SK^{TRC}_{MTC_i} = H(K_{MTC_i}, X^{new}_i, ID_i)$
$VS_{MTC_i} = H(SK^{TRC}_{MTC_i}, ID_i, T_{TRC})$
6. $\{VS_{MTC_i}\}$
$TAVS = K'_{AMF} \oplus X^{new}_n \oplus H(VS_{MTC_1}, VS_{MTC_2} \cdots, VS_{MTC_n}, GID, K_{MTC_n})$
6. $\{TAVS\}$

Figure 2: Authentication process of the proposed protocol.

sends the message $\{\{\mathrm{TID}_i.\mathrm{HID}_i, M_{\mathrm{MTC}_i}\}_{i=1}^{n}, \mathrm{TGID}, \mathrm{MAC}_L, T_{\mathrm{MTC}_n}\}$ to AMF

(3) On receiving the messages, AMF sends the message $\{\{\mathrm{TID}_i, \mathrm{HID}_i, M_{\mathrm{MTC}_i}\}_{i=1}^{n}, \mathrm{TGID}, \mathrm{MAC}_L, T_{\mathrm{MTC}_n}\}$ to TRC

(4) When TRC receives the message from AMF, it first verifies whether the timestamp T_{MTC_n} is within the legal range. If it is within the legal scope, TRC calculates $e_i = \mathrm{PUF}_{\mathrm{TRC}}^{-1}(s, \mathrm{TID}_i)$, $A_i^* = H(s, e_i)$, and $\mathrm{ID}_i^* = A_i^* \oplus \mathrm{HID}_i$. TRC queries the database to verify whether the identity ID_i^* is legal. If the verification is legal, TRC gets $\mathrm{PK}_{\mathrm{MTC}_i}$ and calculates $K_{\mathrm{MTC}_i}^* = \mathrm{PUF}_{\mathrm{TRC}}(\mathrm{PK}_{\mathrm{MTC}_i})$, $\mathrm{GID} = \mathrm{TGID} \oplus H(A_n^*, \mathrm{ID}_n^*, K_{\mathrm{MTC}_n}^*)$, $(\mathrm{PK}_{\mathrm{MTC}_i}^{\mathrm{new}} \| X_i^{\mathrm{new}}) = (H(A_i^*, \mathrm{ID}_i^*) \| K_{\mathrm{MTC}_i}^*) \oplus M_{\mathrm{MTC}_i}$. Then, it calculates $\mathrm{MAC}_i'{=}H(\mathrm{ID}_i^*, \mathrm{TID}_i, K_{\mathrm{MTC}_i}^*, \mathrm{PK}_{\mathrm{MTC}_i}^{\mathrm{new}}, X_i^{\mathrm{new}})$. TRC calculates MAC_L' $=H(\mathrm{MAC}_1', \mathrm{MAC}_2'\bullet\bullet\bullet,\mathrm{MAC}_n', T_{\mathrm{MTC}_n}, \mathrm{GID})$ and verifies whether MAC_L' and MAC_L are equal. If they are equal, then the group MTC devices are certified. If they are not equal, there are illegal devices in the group. TRC selects random value n_i and timestamp T_{TRC}; calculates $\mathrm{TID}_i^{\mathrm{new}} = \mathrm{PUF}_{\mathrm{TRC}}(s, n_i)$, $A_i^{\mathrm{new}} = H(s, n_i)$, $\mathrm{PK}_i^{\mathrm{new}} = \mathrm{PUF}_{\mathrm{TRC}}(X_i^{\mathrm{new}})$, $C_i = K_{\mathrm{MTC}_i}^* \oplus A_i^{\mathrm{new}}$, and $D_i = (H(\mathrm{ID}_i, \mathrm{A}_i^*, K_{\mathrm{MTC}_i}^*) \| A_i^{\mathrm{new}}) \oplus (\mathrm{TID}_i^{\mathrm{new}} \| \mathrm{PK}_i^{\mathrm{new}})$; and updates value $(\mathrm{ID}_i, \mathrm{PK}_{\mathrm{MTC}_i}^{\mathrm{new}})$, stored in the database. Then, TRC generates verification message $\mathrm{MAC}_{\mathrm{TRC}}^{\mathrm{MTC}_i} = H(\mathrm{ID}_i^*, X_i^{\mathrm{new}}, K_{\mathrm{MTC}_i}^*, T_{\mathrm{TRC}}, A_i^{\mathrm{new}}, \mathrm{PK}_i^{\mathrm{new}})$ and the session key $\mathrm{SK}_{\mathrm{TRC}}^{\mathrm{MTC}_i} = H(K_{\mathrm{MTC}_i}^*, X_i^{\mathrm{new}}, \mathrm{ID}_i^*)$. TRC generates verification value $\mathrm{VS}_{\mathrm{TRC}}^{\mathrm{MTC}_i} = H(\mathrm{SK}_{\mathrm{TRC}}^{\mathrm{MTC}_i}, \mathrm{ID}_i^*, T_{\mathrm{TRC}})$ and aggregates the verification values to obtain $\mathrm{AVS} = H(\mathrm{VS}_{\mathrm{TRC}}^{\mathrm{MTC}_1}, \mathrm{VS}_{\mathrm{TRC}}^{\mathrm{MTC}_2}\bullet\bullet\bullet, \mathrm{VS}_{\mathrm{TRC}}^{\mathrm{MTC}_n}, \mathrm{GID}, K_{\mathrm{MTC}_n}^*)$. Finally, TRC sends message $\{\{C_i, D_i, \mathrm{MAC}_{\mathrm{TRC}}^{\mathrm{MTC}_i}\}_{i=1}^{n}, \mathrm{GID}, \mathrm{AVS}, \mathrm{PK}_{\mathrm{MTC}_n}^{\mathrm{new}}, X_n^{\mathrm{new}}, T_{\mathrm{TRC}}\}$ to AMF

(5) After receiving the message sent from TRC, AMF verifies whether the timestamp T_{TRC} is within the legal range. If the verification is legal, it stores the group identity GID and AVS; calculates $\mathrm{PK}_{\mathrm{AMF}} = \mathrm{PUF}_{\mathrm{AMF}}(X_n^{\mathrm{new}})$, $K_{\mathrm{AMF}} = \mathrm{PUF}_{\mathrm{AMF}}(\mathrm{PK}_{\mathrm{MTC}_n}^{\mathrm{new}})$, $\mathrm{MAC}_{\mathrm{AMF}} = H(\mathrm{PK}_{\mathrm{MTC}_n}^{\mathrm{new}}, X_n^{\mathrm{new}}, K_{\mathrm{AMF}}, \mathrm{GID}, T_{\mathrm{TRC}})$, and $F_{\mathrm{AMF}} = X_n^{\mathrm{new}} \oplus \mathrm{PK}_{\mathrm{AMF}}$; and forwards the message $\{\{C_i, D_i, \mathrm{MAC}_{\mathrm{TRC}}^{\mathrm{MTC}_i}\}_{i=1}^{n}, \mathrm{AVS}, \mathrm{MAC}_{\mathrm{AMF}}, F_{\mathrm{AMF}}, T_{\mathrm{TRC}}\}$ to MAC_n

(6) MAC_n receives the message sent and verifies whether the timestamp T_{TRC} is within the legal range. If the verification is legal, it calculates $\mathrm{PK}_{\mathrm{AMF}} = X_n^{\mathrm{new}} \oplus F_{\mathrm{AMF}}$, $K_{\mathrm{AMF}}' = \mathrm{PUF}_{\mathrm{MTC}_n}(\mathrm{PK}_{\mathrm{AMF}})$, and $\mathrm{MAC}_{\mathrm{AMF}}' = H(\mathrm{PK}_{\mathrm{MTC}_n}^{\mathrm{new}}, X_n^{\mathrm{new}}, K_{\mathrm{AMF}}', \mathrm{GID}, T_{\mathrm{TRC}})$. It verifies whether $\mathrm{MAC}_{\mathrm{AMF}}'$ and $\mathrm{MAC}_{\mathrm{AMF}}$ are equal. If they are equal, it verifies AMF. Then, MAC_n calculates $A_n^{\mathrm{new}} = K_{\mathrm{MTC}_n} \oplus C_n$, $(\mathrm{TID}_n^{\mathrm{new}} \| \mathrm{PK}_n^{\mathrm{new}}) = (H(\mathrm{ID}_n, A_n, K_{\mathrm{MTC}_n}) \| A_n^{\mathrm{new}}) \oplus D_n$, and $\mathrm{MAC}_{\mathrm{MTC}_n}^{\mathrm{TRC}} = H(\mathrm{ID}_n, X_n^{\mathrm{new}}, K_{\mathrm{MTC}_n}, T_{\mathrm{TRC}}, A_n^{\mathrm{new}}, \mathrm{PK}_n^{\mathrm{new}})$. If the generated value $\mathrm{MAC}_{\mathrm{MTC}_n}^{\mathrm{TRC}}$ and the received value $\mathrm{MAC}_{\mathrm{TRC}}^{\mathrm{MTC}_n}$ are equal, then it verifies the server TRC and updates the device parameters at the same time. MTC_n generates the session key $\mathrm{SK}_{\mathrm{MTC}_n}^{\mathrm{TRC}} = H(K_{\mathrm{MTC}_n}, X_n^{\mathrm{new}}, \mathrm{ID}_n)$ and the verification value $\mathrm{VS}_{\mathrm{MTC}_n} = H(\mathrm{SK}_{\mathrm{MTC}_n}^{\mathrm{TRC}}, \mathrm{ID}_n, T_{\mathrm{TRC}})$. Finally, MTC_n forwards message $\{C_i, D_i, \mathrm{MAC}_{\mathrm{TRC}}^{\mathrm{MTC}_i}, T_{\mathrm{TRC}}\}$ to MAC_i

(7) When receiving a message from MAC_n, MTC_i verifies whether the received timestamp T_{TRC} is legal. If the timestamp T_{TRC} is legal, MTC_i calculates $A_i^{\mathrm{new}} = K_{\mathrm{MTC}_i} \oplus C_i$, $(\mathrm{TID}_i^{\mathrm{new}} \| \mathrm{PK}_i^{\mathrm{new}}) = (H(\mathrm{ID}_i, A_i, K_{\mathrm{MTC}_i}) \| A_i^{\mathrm{new}}) \oplus D_i$, and $\mathrm{MAC}_{\mathrm{MTC}_i}^{\mathrm{TRC}} = H(\mathrm{ID}_i, X_i^{\mathrm{new}}, K_{\mathrm{MTC}_i}, T_{\mathrm{TRC}}, A_i^{\mathrm{new}}, \mathrm{PK}_i^{\mathrm{new}})$. If the generated value $\mathrm{MAC}_{\mathrm{MTC}_i}^{\mathrm{TRC}}$ and the received value $\mathrm{MAC}_{\mathrm{TRC}}^{\mathrm{MTC}_i}$ are equal, then it verifies the server TRC and updates the device parameters at the same time. MTC_i generates the session key $\mathrm{SK}_{\mathrm{MTC}_i}^{\mathrm{TRC}} = H(K_{\mathrm{MTC}_i}, X_i^{\mathrm{new}}, \mathrm{ID}_i)$ and the verification value $\mathrm{VS}_{\mathrm{MTC}_i} = H(\mathrm{SK}_{\mathrm{MTC}_i}^{\mathrm{TRC}}, \mathrm{ID}_i, T_{\mathrm{TRC}})$. Finally, the message $\{\mathrm{VS}_{\mathrm{MTC}_i}\}$ is sent to MTC_n

(8) On receiving the message sent by the group members, MTC_n calculates TAVS= $K_{\mathrm{AMF}}' \oplus X_n^{\mathrm{new}} \oplus H(\mathrm{VS}_{\mathrm{MTC}_1}, \mathrm{VS}_{\mathrm{MTC}_2}\bullet\bullet\bullet,\mathrm{VS}_{\mathrm{MTC}_n}, \mathrm{GID}, K_{\mathrm{MTC}_n})$ and sends it to AMF

(9) AMF receives the message and calculates $\mathrm{AVS}^* = K_{\mathrm{AMF}} \oplus X_n^{\mathrm{new}} \oplus \mathrm{TAVS}$. Then, it compares AVS with the received AVS^*. If they are equal, the correctness of the generated session key is verified

Finally, MTC_i communicates through the session key.

5. Security Evaluation

5.1. Security Proof Based on BAN Logic

5.1.1. BAN Logic Rules. In this paper, BAN logic is used to formally analyze the proposed authentication scheme. BAN logic [26] is a formal analysis tool based on knowledge and belief.

5.1.2. Verification. Here, we formally verify our scheme. First, we idealize the scheme.

(1) The messages involved in our scheme are idealized

$\mathrm{Mes}_1 : \mathrm{MTCD_i} \longrightarrow \mathrm{TRC} : <\mathrm{TID}_i, \mathrm{HID}_i, M_{\mathrm{MTC}_i}, \mathrm{MAC}_i>_{K_{\mathrm{MTC}_i}}$

$\mathrm{Mes}_2 : \mathrm{TRC} \longrightarrow \mathrm{MTCD}_i : <C_i, D_i, \mathrm{MAC}_{\mathrm{TRC}}^{\mathrm{MTC}_i}, T_{\mathrm{TRC}}>_{K_{\mathrm{MTC}_i}}$

(2) Formal description of initial state

$A_1 : \mathrm{MTCD}_i| \equiv \mathrm{MTCD}_i \overset{K_{\mathrm{MTC}_i}}{\leftrightarrow} \mathrm{TRC}$

$A_2 : \mathrm{TRC}| \equiv \mathrm{MTCD}_\mathrm{i} \overset{K^*_{\mathrm{MTC}_i}}{\leftrightarrow} \mathrm{TRC}$

$A_3 : \mathrm{TRC}|\equiv\#(X_i^{\mathrm{new}})$

$A_4 : \mathrm{TRC}| \equiv \mathrm{MTCD}_i \Rightarrow <\mathrm{TID}_i, \mathrm{HID}_i, M_{\mathrm{MTC}_i}, \mathrm{MAC}_i >$

$A_5 : \mathrm{TRC}| \equiv \mathrm{MTCD}_i \Rightarrow \mathrm{MTCD}_\mathrm{i} \overset{\mathrm{SK}_{\mathrm{TRC}}^{\mathrm{MTC}_i}}{\leftrightarrow} \mathrm{TRC}$

$A_6 : \mathrm{MTCD}_i|\equiv\#(T_{TRC})$

$A_7 : \mathrm{MTCD}_i| \equiv \mathrm{TRC} \Rightarrow <\mathrm{TID}_i, \mathrm{HID}_i, M_{\mathrm{MTC}_i}, \mathrm{MAC}_i >$

$A_8 : \mathrm{MTCD}_i| \equiv \mathrm{TRC} \Rightarrow \mathrm{MTCD}_i \overset{\mathrm{SK}_{\mathrm{TRC}}^{\mathrm{MTC}_i}}{\leftrightarrow} \mathrm{TRC}$

(3) The ultimate goal of the scheme

In this section, our scheme needs to meet the following goals:

$G_1 : \mathrm{MTCD}_i| \equiv \mathrm{MTCD}_\mathrm{i} \overset{\mathrm{SK}^{\mathrm{TRC}}_{\mathrm{MTC}_i}}{\leftrightarrow} \mathrm{TRC}$

$G_2 : \mathrm{MTCD}_i|\equiv\mathrm{TRC}| \equiv \mathrm{MTCD}_\mathrm{i} \overset{\mathrm{SK}^{\mathrm{TRC}}_{\mathrm{MTC}_i}}{\leftrightarrow} \mathrm{TRC}$

$G_3 : \mathrm{TRC}| \equiv \mathrm{MTCD}_i \overset{\mathrm{SK}_{\mathrm{TRC}}^{\mathrm{MTC}_i}}{\leftrightarrow} \mathrm{TRC}$

$G_4 : \mathrm{TRC}|\equiv\mathrm{MTCD}_i| \equiv \mathrm{MTCD}_\mathrm{i} \overset{\mathrm{SK}_{\mathrm{TRC}}^{\mathrm{MTC}_i}}{\leftrightarrow} \mathrm{TRC}$

(4) Logical reasoning

According to the message Mes_1 sent by MTC_i to TRC, it can be concluded that

$S_1 : \mathrm{TRC}\triangleleft<\mathrm{TID}_i, \mathrm{HID}_i, M_{\mathrm{MTC}_i}, \mathrm{MAC}_i>_{K_{\mathrm{MTC}_i}}$

Given S_1 and A_2, from the message meaning rule, we can get

$S_2 : \mathrm{TRC}\ |\equiv\mathrm{MTCD}_i| \sim \mathrm{TID}_i, \mathrm{HID}_i, M_{\mathrm{MTC}_i}, \mathrm{MAC}_i$

From S_1, A_3 and the freshness rule, we can get

$S_3 : \mathrm{TRC}| \equiv \#\{\mathrm{TID}_i, \mathrm{HID}_i, M_{\mathrm{MTC}_i}, \mathrm{MAC}_i\}$

From S_2,S_3, and nonce verification rule, we can get

$S_4 : \mathrm{TRC}|\equiv\mathrm{MTCD}_i| \equiv \{\mathrm{TID}_i, \mathrm{HID}_i, M_{\mathrm{MTC}_i}, \mathrm{MAC}_i\}$

From S_4, A_4, and arbitration rules, we can get

$S_5 : \mathrm{TRC}| \equiv \{\mathrm{TID}_i, \mathrm{HID}_i, M_{\mathrm{MTC}_i}, \mathrm{MAC}_i\}$

Given S_5 and $\mathrm{SK}_{\mathrm{TRC}}^{\mathrm{MTC}_i} = H(K^*_{\mathrm{MTC}_i}, X_i^{\mathrm{new}}, \mathrm{ID}_i)$, we can get

$S_6 : \mathrm{TRC}\ |\equiv\mathrm{MTCD}_i| \equiv \mathrm{MTCD}_i \overset{\mathrm{SK}_{\mathrm{TRC}}^{\mathrm{MTC}_i}}{\leftrightarrow} \mathrm{TRC}$

According to S_6, A_5, and the arbitration rule, we can get

$S_7 : \mathrm{TRC}| \equiv \mathrm{MTCD}_i \overset{\mathrm{SK}_{\mathrm{TRC}}^{\mathrm{MTC}_i}}{\leftrightarrow} \mathrm{TRC}$

According to the message Mes_2 sent by TRC to MTCD_i, we can get:

$S_8 : \mathrm{MTCD}_i\triangleleft<C_i, D_i, \mathrm{MAC}_{\mathrm{TRC}}^{\mathrm{MTC}_i}, T_{\mathrm{TRC}}>_{K_{\mathrm{MTC}_i}}$

Given S_8 and A_1, from the message meaning rule, we can get

$S_9 : \mathrm{MTCD}_i|\equiv\mathrm{TRC}| \sim \{C_i, D_i, \mathrm{MAC}_{\mathrm{TRC}}^{\mathrm{MTC}_i}, T_{\mathrm{TRC}}\}$

According to S_9, A_6, and the freshness rule, we can get

$S_{10} : \mathrm{MTCD}_i| \equiv \#(C_i, D_i, \mathrm{MAC}_{\mathrm{TRC}}^{\mathrm{MTC}_i}, T_{\mathrm{TRC}})$

From S_9, S_{10}, and the nonce verification rule, we can see

$S_{11} : \mathrm{MTCD}_i|\equiv\mathrm{TRC}| \equiv \{C_i, D_i, \mathrm{MAC}_{\mathrm{TRC}}^{\mathrm{MTC}_i}, T_{\mathrm{TRC}}\}$

From S_{11}, A_7, and the arbitration rule, we can get

Table 2: Computation overhead.

Protocol	Computation overhead
LPPA [13]	$(2n + 2m)T_{D/E} + (12n + 8m)T_H$
LSAA [14]	$16nT_{CM} + 4nT_{E/D} + 6nT_H$
Our scheme	$(6n + 3m)T_{\mathrm{PUF}} + (14n + 7m)T_H$

Table 3: Communication overhead.

Protocol	Leader MTC
LPPA [13]	$640n + 1280m$
LSAA [14]	$1152n + 384m$
Our scheme	$1056n - 256m$

$S_{12} : \mathrm{MTCD}_i| \equiv \{\mathrm{TID}_i, \mathrm{HID}_i, M_{\mathrm{MTC}_i}, \mathrm{MAC}_i\}$

From S_{12} and $\mathrm{SK}^{\mathrm{TRC}}_{\mathrm{MTC}_i} = H(K_{\mathrm{MTC}_i}, X_i^{\mathrm{new}}, \mathrm{ID}_i)$, we can see

$S_{13} : \mathrm{MTCD}_i|\equiv\mathrm{TRC}| \equiv \mathrm{MTCD}_i \overset{\mathrm{SK}^{\mathrm{TRC}}_{\mathrm{MTC}_i}}{\leftrightarrow} \mathrm{TRC}$

According to S_{13}, A_8, and the arbitration rule, we can get

$S_{14} : \mathrm{MTCD}_i| \equiv \mathrm{TRC} \overset{\mathrm{SK}_{\mathrm{TRC}}^{\mathrm{MTC}_i}}{\leftrightarrow} \mathrm{TRC}$

Through S_6, S_7, S_{13}, and S_{14}, we can see that our scheme reaches the goals.

5.2. Security Analysis. The security of our scheme is mainly analyzed from the aspects of identity authentication, session key security, resistance to attacks, and so on.

(1) *Identity Authentication.* In our scheme, communication entities use message authentication codes to verify their legitimacy. Because the generated message verification code includes the secret value generated by BS-PUF, the security of the verification message is guaranteed

(2) *Session Key Security.* Each MTC device negotiates a session key with the server. The corresponding session key is generated through the secret value generated by the BS-PUF and other parameters, ensuring the security of the session key

(3) *Identity Anonymity and Unlinkability.* In our scheme, each MTC device communicates with the server through pseudonym $\mathrm{TID}_i = \mathrm{PUF}_{\mathrm{TRC}}(s, e_i)$, and the real identity is encrypted as $M_{\mathrm{MTC}_i} = H_1(A_i, T_{\mathrm{MTC}_i}) \oplus \mathrm{ID}_i$. After receiving the pseudonym TID_i and M_{MTC_i}, the server obtains the real identity through calculation. Because the real identity of the device can be obtained only through the calculation of the server, the anonymity of the device is guaranteed. Because the temporary identity of each MTC device in the scheme changes and the generated messages use random numbers and time stamps, the messages transmitted in the network are different, and the attacker cannot distinguish that the two messages are sent by the same device

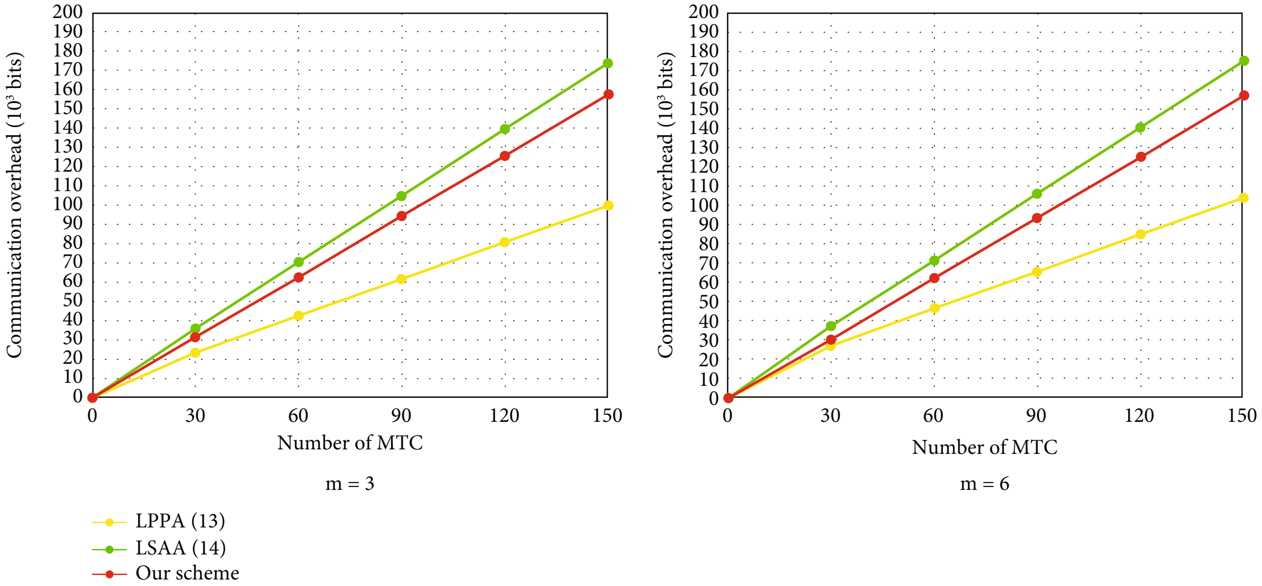


Figure 3: Communication overhead between different protocols.

(4) *Forward Security*. Each MTC device negotiates with the server to generate a corresponding session key through the secret value and random number generated. Because the secret value and random number generated for each authentication are different, the security of the session key is guaranteed. Even if the session key is leaked, it will not affect the previously generated session keys

(5) *Antiattack Ability*. In the communication process of our scheme, each MTC device ensures the freshness of messages by using time stamps, so it can effectively avoid replay attacks. In the process of message verification, our scheme uses the message authentication code. Because the message authentication code is generated by the secret value and other parameters generated, it is difficult for the attacker to generate the correct message authentication code, so it can effectively avoid man in the middle attack. In our scheme, because the real identity is encrypted, it is difficult for the attacker to extract the user identity from the message, so it is difficult to impersonate a legitimate user for communication. In the authentication process, since the secret value $K^*_{\mathrm{MTC}_i}$ can only be generated by the server, the attacker cannot generate this value for verification, so it is difficult for the attacker to impersonate the server

(6) *Avoid Authentication Signaling Congestion*. Our scheme uses aggregation message authentication technology to aggregate a group of MTC device request messages into one request message. Here, we complete the message aggregation in leader MTC, reduce the signaling computation and communication overhead, and send it to the server for authentication. Our scheme effectively simplifies the authentication process, reduces the authentication delay, and avoids signaling congestion

6. Performance Analysis

In this section, we mainly analyze the performance of our scheme from two aspects: computation overhead and communication overhead. Here, we mainly compare the schemes similar to our scheme.

6.1. Computation Overhead. By calculating the time of various encryption operations, we analyze the computation overhead of the protocol. In this paper, we omit the lightweight operations including XOR operations and concatenation operations. Here, $T_{D/E}$ represents the time to calculate symmetric encryption or decryption, T_H represents the time to calculate one-way hash, and T_{CM} represents the time to calculate an extended chaotic map. In addition, we refer to [17] to obtain $T_H \approx 1.6 T_{\mathrm{PUF}}$. The computation overhead of relevant schemes is obtained, as shown in Table 2.

Therefore, we can see that our scheme has obvious advantages in terms of computation overhead.

6.2. Communication Overhead. Here, we evaluate the communication overhead of our scheme by comparing similar schemes. We define the size of different authentication messages. In this article, we refer to standards [27, 28]. Assume that the random number, hash value, and device identity size are 128 bits. The size of the time stamp is 32 bits. The size of the chaotic map is 128 bits. In the scheme of [17], we define the size of PUF to be 128 bits. According to the size of the defined message, we obtain the size of the communication overhead of the comparison schemes. Because of different schemes, the number of server entities communicating is different. Therefore, for the sake of fairness, we mainly compare the communication overhead of the group leader MTC device in Table 3.

Figure 3 shows the comparison results of different m values and changes in the number of devices. We can see that

[13] has small communication overhead, but it has security vulnerabilities. Therefore, our scheme has obvious advantages in terms of communication overhead and security.

7. Conclusion

Due to the signaling congestion and security problems encountered for mMTC communication in 5G networks, we propose a mMTC group authentication scheme. The scheme is based on lightweight encryption operation, which reduces the computational burden of equipment and server, and ensures the security of the scheme. Then, security verification of the proposed scheme is carried out through BAN logic and informal security analysis. The verification results show that our scheme has strong security in the process of encryption and authentication and can resist most known attacks. The data analysis shows that the proposed scheme has great improvement in communication overhead and computation overhead compared with the existing schemes. In the future research work, we will start to study the authentication scheme based on group signature. With the development of 5G communication technology, a more efficient scheme is designed to meet the requirements of lightweight and security.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This work was supported by the 2020 Industrial Technology Foundation Public Service Platform Project (grant number 2020-0105-2-1).

References

[1] M. Series, “Framework and overall objectives of the future development of IMT for 2000 and beyond,” ITU-R M. 2083, 2015.

[2] G. Wunder, Č. Stefanović, P. Popovski, and L. Thiele, “Compressive coded random access for massive MTC traffic in 5G systems,” in *2015 49th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, 2017.

[3] D. T. Wiriaatmadja and K. W. Choi, “Hybrid random access and data transmission protocol for machine-to-machine communications in cellular networks,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 33–46, 2015.

[4] 3rd Generation Partnership Project (3GPP)TS33.501-f10, “Technical specification group services and system aspects,” Security architecture and procedures for 5G system, V.15.0.0, 2018.

[5] W. Zhan and L. Dai, “Massive random access of machine-to-machine communications in LTE networks: modeling and throughput optimization,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2771–2785, 2018.

[6] M. A. Ferrag, L. Maglaras, A. Argyriou, D. Kosmanos, and H. Janicke, “Security for 4G and 5G cellular networks: a survey of existing authentication and privacy-preserving schemes,” *Journal of Network & Computer Applications*, vol. 101, pp. 55–82, 2018.

[7] C. Lai, H. Li, X. Li, and J. Cao, “A novel group access authentication and key agreement protocol for machine- type communication,” *Transactions on Emerging Telecommunications Technologies (ETT)*, vol. 26, no. 3, pp. 414–431, 2015.

[8] J. Puneet, H. Peter, and Z. Haris, “Machine type communications in 3GPP systems,” *IEEE Communications Magazine*, vol. 50, no. 11, pp. 28–35, 2012.

[9] C. Lai, R. Lu, D. Zheng, H. Li, and X. Shen, “Toward secure large-scale machine-to-machine communications in 3GPP networks: challenges and solutions,” *IEEE Communications Magazine*, vol. 53, no. 12, pp. 12–19, 2015.

[10] J. Cao, M. Ma, H. Li et al., “A survey on security aspects for 3GPP 5G networks,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 170–195, 2020.

[11] J. Cao, M. Ma, H. Li, Y. Fu, and X. Liu, “EGHR: efficient group-based handover authentication protocols for mMTC in 5G wireless networks,” *Journalof Network and Computer Application*, vol. 102, pp. 1–16, 2018.

[12] 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.

[13] J. Cao, M. Ma, and H. Li, “LPPA: lightweight privacy-preservation access authentication scheme for massive devices in fifth Generation (5G) cellular networks,” *International Journal of Communication Systems*, vol. 32, no. 3, article e3860, 2019.

[14] J. Cao, Z. Yan, R. Ma, Y. Zhang, Y. Fu, and H. Li, “LSAA: a lightweight and secure access authentication scheme for both UE and mMTC devices in 5G networks,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5329–5344, 2020.

[15] P. Gope, J. Lee, and T.-Q. S. Quek, “Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2831–2843, 2018.

[16] Y. Guo, T. Dee, and A. Tyagi, “Barrel shifter physical unclonable function based encryption,” *Cryptography*, vol. 2, no. 3, p. 22, 2018.

[17] T. F. Lee and W. Y. Chen, “Lightweight fog computing-based authentication protocols using physically unclonable functions for Internet of Medical Things,” *Journal of Information Security and Applications*, vol. 59, no. 4, article 102817, 2021.

[18] 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 *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 832–837, Atlanta, GA, USA, 2013.

[19] J. Cao, M. Ma, and H. Li, “GBAAM: group-based access authentication for MTC in LTE networks,” *Security and Communication Networks*, vol. 8, pp. 3282–3299, 2015.

[20] Y. Zhang, J. Chen, H. Li, J. Cao, and C. Lai, “Group-based authentication and key agreement for machine-type communication,” *International Journal of Grid and Utility Computing*, vol. 5, no. 2, pp. 87–95, 2014.

[21] J. Cao, H. Li, and M. Ma, "GAHAP: a group-based anonymity handover authentication protocol for MTC in LTE-A networks," in *IEEE International Conference on Communications*, London, UK, 2015.

[22] 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.

[23] J. Cao, H. Li, M. Ma, and F. Li, "UPPGHA: uniform privacy preservation group handover authentication mechanism for mMTC in LTE-A networks," *Security and Communication Networks*, vol. 2018, Article ID 6854612, 16 pages, 2018.

[24] J. Miao, Z. Wang, X. Miao, and L. Xing, "A secure and efficient lightweight vehicle group authentication protocol in 5G networks," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 4079092, 12 pages, 2021.

[25] D. Dolev and A. Yao, "On the security of public-key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[26] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 23, no. 5, pp. 1–13, 1989.

[27] National Institute of Standards and Technology, "Special publication 800-57: recommendation for key management part 1: general (revision 4)," NIST.SP.800-57pt1r4, 2016.

[28] "Federal information processing standards publication digital signature standard (DSS)," FIPS PUB 186-4, 2013.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7050573, 9 pages
https://doi.org/10.1155/2022/7050573

Research Article

A Novel Approach Based on Generative Adversarial Network for Interference Detection in Wireless Communications

Shoushuai He,[1] Lei Zhu,[1] Changhua Yao,[2] Weijun Zeng,[1] and Zhen Qin[1]

[1]*College of Communications Engineering, Army Engineering University, Nanjing 210007, China*
[2]*School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China*

Correspondence should be addressed to Lei Zhu; zhulei_paper@126.com

Received 23 June 2022; Accepted 24 August 2022; Published 21 September 2022

Academic Editor: Chenglu Jin

With the rapid growth of wireless devices, the communication environment gets complex. The detection of interference or unauthorized signals can improve spectrum efficiency, which is a key technology for limited spectrum resources. Traditional detection methods analyze the parameter characteristics of the received signal. But it is difficult to detect interference with the same time and frequency as the original signal by those feature engineering. As a classical problem in deep learning, anomaly detection is usually solved by supervised learning. But a more challenging situation is to detect unknown or invisible anomalies. It means that the number of abnormal samples is insufficient and the data is highly biased toward the normal samples. In this paper, a wireless communication interference detection algorithm based on generative adversarial network (GAN) is proposed. In the semi-supervised learning scenario, the algorithm detects the time-frequency overlapped interference by the reconstruction strategy. The generator adopts the encoder-decoder-encoder architecture. In the training process, the model jointly learns the data distribution of normal samples by minimizing the distance in both the signal space and the latent space. In the inference phase, a large distance metric implies an abnormal sample. Experiments on simulated communication datasets show the superiority of the proposed algorithm.

1. Introduction

Wireless communications enrich life and facilitate production. The development of information technology undoubtedly poses a challenge to communication security. Electromagnetic spectrum is the transmission medium of wireless communications, and the spectrum resources are limited. In actual communications, different wireless services use corresponding frequency bands. Since no more communication signals can be included in a specific frequency band, spectrum resources get more scarce and valuable. However, the increasing number of new users and wireless devices makes the communication environment complex. In this case, communication signals transmitted in the channel are easily overlapped by interference signals with the same frequency. The decoding of these received signals leads to many errors and further affects the communication behavior. In order to prevent this situation, it is necessary to allocate spectrum resources reasonably and detect time-frequency overlapped interference timely [1]. On this basis, interference signals can also be separated for further analysis and identification. A common and key problem to be solved in wireless communications is the guarantee of communication quality, so interference detection is essential.

Supervised methods of deep learning have achieved encouraging performance in various computer vision tasks, but they heavily rely on large labeled datasets. In some practical applications, the samples of a specific category may not be enough to construct the model effectively. For instance, it is often difficult to obtain a large amount of training data that causes security threats, and the data may change with external factors. However, the task of interference detection is to deal with this challenging situation. In other words, the model only trains on normal samples, then identifies abnormal samples that are not fully available and different from normal distribution. In addition, the receiver may be unknown about the operating parameters of the communication system. Therefore, the model needs to detect the interference in the signal with limited prior information.

Many researches have proposed anomaly detection models for different application fields [2]. As a dominant method for unsupervised and semi-supervised problems, generative adversarial network (GAN) [3] was initially introduced by Goodfellow et al. and also applied to anomaly detection. GAN jointly trains a pair of networks: a generator and a discriminator. The former simulates high-dimensional data by latent vectors to approximate the original distribution, while the latter distinguishes between generated samples and real samples. The generator is a network similar to the decoder, and the discriminator is a typical classification network. They compete with each other during training to learn the features of the original data.

Inspired by GANomaly [4], an interference detection architecture for wireless communication signals is proposed in this paper, which includes an adversarial training framework. The algorithm adopts deep learning method and extracts features end-to-end. It is only trained on normal samples and is suitable for learning representation from time-series data. The received signals are directly used as inputs to the algorithm. In the training phase, they only contain a set of normal samples. Based on the reconstruction strategy, the algorithm detects whether interference exists in the received signal. It captures the distribution of training data by joint learning in both signal space and latent space, where latent space helps to learn data features and simplify data representation. The generator works on the pipeline of encoder-decoder-encoder. Specifically, the original signal is mapped to a low-dimensional vector; then, the vector is used to reconstruct the generated signal, and finally, the signal is mapped to its latent representation. In this way, the detection effect under noisy conditions is improved. Experimental results on simulated communication datasets show that the proposed algorithm has better detection performance for time-frequency overlapped interference. And even in the case of low signal-to-noise ratio (SNR), the algorithm is still effective.

The main contents and specific sections of this paper are arranged as follows: in Section 2, the research status of interference detection in wireless communications and some related methods are introduced. In Section 3, the scenario of time-frequency overlapped interference and the detection algorithm based on generative adversarial network are proposed. Then, in Section 4, the experimental results verify that the algorithm can solve the interference detection problem and achieve better detection effect. Finally, this paper is concluded in Section 5.

2. Related Work

Anomaly detection has attracted extensive attention in various fields for a long time, such as network intrusion [5], video surveillance [6], financial fraud [7], and disease monitoring [8]. The related research in wireless communications is also common [9], in which interference detection and interference source localization are particularly important.

Anti-interference technology plays an active role in wireless communications [10]. It has always been a promising research direction in civil and military applications and has got a lot of valuable research results. As a key procedure in anti-interference, interference detection can find the existence of interference signals and provide necessary support for anti-interference. In other words, its main task is to detect whether the original signal is overlapped by additional interference during transmission. Further, it can obtain the parameters of the interference signal, such as type, power, and frequency. Through the feedback of this information, the anti-interference system can take corresponding measures to suppress interference signals, then reduce the bit error rate of decoding at the receiver. In addition, spectrum monitoring in cognitive radio is an important aspect of wireless communications [11], which is closely related to interference detection.

In recent years, the research on interference detection in wireless communications has gradually increased, and some effective methods have been proposed. Most of them analyze signals according to domain transformation or statistical characteristics [12]. Traditional methods use priori information for calculation, thus face some difficulties. And the proliferation of wireless devices also brings them great challenges. In the actual wireless environment, there are many interference signals with small power and the same frequency as the original signal. When the original signals are overlapped by such interference signals, their characteristics in time domain and frequency domain just change slightly. Therefore, the above methods are obviously difficult to detect time-frequency overlapped interference.

With the vigorous development of deep learning, related applications have emerged in many fields, such as speech enhancement [13, 14] and image denoising [15–17]. Deep learning is actually a complex neural network. The improvement of its performance is attributed to the increase of network layers, the optimization of network structure, and the expansion of training data. Different from traditional methods, deep learning is independent of artificial features. It directly takes the original data as input for training and automatically extracts the corresponding features. Through the calculation of multiple neurons, the network continuously adjusts the parameter values to update the output results. As a typical representative, convolutional neural network (CNN) has the outstanding advantages of sparse connection and weight sharing. Compared with the fully connected network, it simplifies the calculation process and reduces the number of parameters. As the complexity of the network gets decreased, the training of the network gets accelerated. Signal processing based on deep learning focuses on one-dimensional data with periodicity [18]. They no longer need to analyze signal models and can be expanded to new scenarios by collecting data samples. The above researches not only provide a theoretical basis for the application of deep learning in wireless communications but also provide a factual basis for neural networks to extract signal features.

Interference detection in wireless communications is not exactly the same as general anomaly detection [19]. On the one hand, the type of interference is complex and variable. Interference may be quite different for various wireless applications. In this case, it is difficult to obtain the label of the interference signal or even the interference signal itself. It is no longer an ordinary classification problem, so supervised training is invalid. On the other hand, there is a lot of noise in the wireless environment. Noise is usually irregular, so neural networks cannot learn useful features. The existence of noise seriously affects the detection accuracy. Therefore, classical anomaly detection methods may not be applicable. However, interference detection in wireless communications has some similarities with anomaly detection in speech and image. In particular, both communication signals and speech signals are time-series data with correlation. Hence, deep learning methods in speech and image can be adopted to design interference detection algorithms, in which communication signals can be directly used as inputs to extract features.

Recently, some deep learning methods based on reconstruction strategy have been applied to anomaly detection [20–22]. Their purpose is to construct a network in the training phase and perform the reconstruction task for normal samples. But in the test phase, the network causes poor reconstructioneffect for abnormal samples due to the distribution difference. In addition, the current research also focuses on adversarial training [23, 24] and especially explores the potential of generative adversarial network [25, 26]. The initial task of GAN is to produce realistic images. The generator attempts to generate samples similar to the training data from the Gaussian distribution, while the discriminator needs to decide whether the generated samples are real or fake. Several improved methods have been proposed to solve its problem of unstable training, such as the use of Wasserstein loss [27].

In summary, the existing research strongly supports the prospect of GAN for interference detection in wireless communications. In this paper, an interference detection algorithm based on generative adversarial network is proposed. The generator is designed as the structure of encoder-decoder-encoder, so as to jointly learn the feature representation in the signal and latent space.

3. Proposed Approach

3.1. Problem Description. The problem of anomaly detection is defined as follows: given a dataset, it contains a large number of normal samples for training and a relatively small number of abnormal samples for testing. In the training phase, the model learns the data distribution of normal samples and optimizes its parameters. In the test phase, the model determines whether the input sample is abnormal by calculating the anomaly score. Since the model is aimed at minimizing the anomaly score, a large value indicates that the input sample may be abnormal. The anomaly score is universal to detect invisible anomalies different from normal distribution. Therefore, it is necessary to train a semi-supervised algorithm for anomaly detection, where the data is highly biased toward a specific category. Then, the existence of abnormal samples is determined by thresholding the anomaly score.

For interference detection in wireless communications, the situation considered in this paper is that the interference signal and the transmitted signal are the same in time and frequency. Traditional methods analyze the statistical information of received signals, such as spectrum and power. But in the actual wireless environment, the parameters of the interference signal may be unknown. So these spectrum analysis methods may fail to detect such time-frequency overlapped interference. Especially when the SNR is low, the interference signal may be submerged in noise and the original signal due to the small power. So it is difficult to judge the existence of the interference signal through the change of power. In addition, the characteristics of the interference signal in the time domain and frequency domain may be similar to those of the original signal, which also makes detection difficult. All the detection of interference signals mentioned in this paper is based on this relatively difficult situation.

3.2. Model Establishment. In order to solve the above problems, an interference detection algorithm based on generative adversarial network is proposed in this paper. As a generative model that has been widely used in the field of image and speech, GAN contains two components: generator G and discriminator D. G generates data, while D measures the difference between the generated data and the labeled data then provides feedback. They compete and cooperate with each other to jointly control the output of the model.

$$\min_G \max_D V_{\mathrm{GAN}}(D, G) = E_{x \sim p_X}[\log D(x)] + E_{z \sim p_Z}[\log (1 - D(G(z)))]. \tag{1}$$

The proposed algorithm adopts the reconstruction strategy to deal with the time-frequency overlapped interference in wireless signals. It can effectively learn the correlation of time-series data and accurately extract the features of transmitted signals. The algorithm can avoid the influence of noise and achieve better detection effect even in the case of low SNR. Figure 1 depicts the overall architecture of the proposed model, which includes three subnetworks.

The first one is an autoencoder, which acts as the generator G of the model. The encoder G_E learns the compressed representation of the input signal, while the decoder G_D reconstructs the input signal. Specifically, G_E consists of convolution, batch normalization, and leaky ReLU activation. The input signal x forward-passes through G_E and obtains the bottleneck feature z. G_D adopts the structure of generator in deep convolutional generative adversarial network (DCGAN), which consists of transposed convolution, batch normalization, ReLU activation, and tanh activation. The latent vector z forward-passes through G_D and expands to the generated signal x'. The dimension of each layer in the network is gradually changed, and the length of the output signal is equal to that of the input signal. In a word, the generator reconstructs the signal x into x' by z.

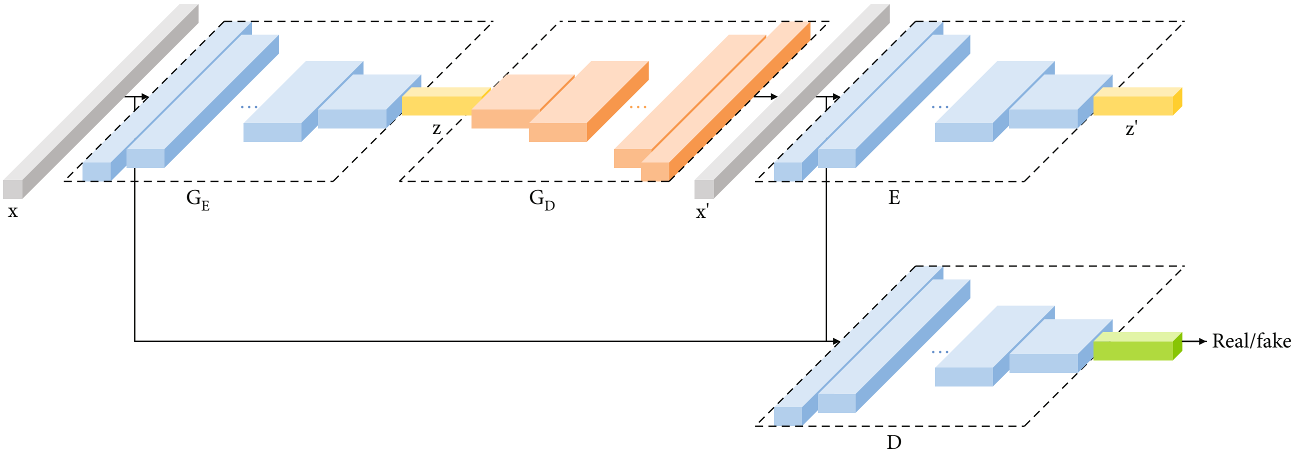


Figure 1: Architecture of the proposed model.

The second one is the encoder E, which compresses the signal x' reconstructed by G. Compared with G_E, it has the same structure but different parameters. E compresses x' to obtain its feature representation z', which has the same dimension as z for consistency comparison.

The third one is the discriminator D, whose goal is to classify x and x' as real or fake. It adopts the structure of discriminator in DCGAN.

3.3. Network Training. Theoretically, when an abnormal signal forward-passes in the generator G, G_D fails to complete the reconstruction through the latent vector mapped by G_E. Because G is only modeled based on normal samples during training, its parameters are not suitable to generate abnormal samples. For the same reason, the reconstructed abnormal sample also causes the encoder E to produce a feature representation different from that of the normal sample. When such dissimilarity occurs in the latent space, the model classifies the input signal as abnormal signal. As shown in Figure 2, three loss functions are established to optimize the subnetworks.

The first one is encoding loss. It can guide the model to encode the generated signal from the normal signal. It minimizes the distance between the bottleneck feature z of the original signal and the encoding feature z' of the generated signal. But for abnormal signals, it fails to do so in the latent space.

$$L_{\text{enc}} = E_{x\sim p_X} \|G_E(x) - E(G(x))\|_2. \tag{2}$$

The second one is reconstruction loss. It can optimize G according to the context information of the input data, then force it to generate an approximately real signal. The distance between the original signal and the generated signal reflects the effect of G. Since the L_1 metric has been proven to produce fewer fuzzy results, it is selected to calculate the distance.

$$L_{\text{rec}} = E_{x\sim p_X} \|x - G(x)\|_1. \tag{3}$$

The third one is feature matching loss. Following the trend of current methods, feature matching loss is applied to adversarial learning in the model, which is proved to reduce the instability of GAN in training. Different from the original method, G is updated based on the internal representation of D, rather than the final output of D. The feature matching loss makes it possible for the generated samples to deceive the discriminator. It is calculated according to the distance between the feature representation of the original sample and the generated sample.

$$L_{\text{fea}} = E_{x\sim p_X} \|f(x) - f(G(x))\|_2. \tag{4}$$

In general, the objective function of the generation phase is defined as the following equation, where λ and μ are hyperparameters that balance the three parts.

$$L_G = L_{\text{enc}} + \lambda L_{\text{rec}} + \mu L_{\text{fea}}. \tag{5}$$

As described above, G and E are optimized according to the encoding of normal samples during training. So in this paper, the anomaly score is defined as the following form. It is expected to be large if x is an abnormal sample.

$$S(x) = \|G_E(x) - E(G(x))\|_2. \tag{6}$$

4. Experiment

4.1. Setup. In order to evaluate the proposed interference detection algorithm, a set of simulated communication data is used in the experiment. The normal samples in the dataset are divided into two parts. The training set consists of 80% normal samples, while the test set consists of the remaining 20% combined with all abnormal samples. The total number of normal samples and abnormal samples is 10000 and 2000, respectively. During the experiment, the original signal and interference signal are modulated by BPSK, QPSK, 8PSK, 16QAM, and 32QAM. Three typical scenarios are set to represent the general situation, which are QPSK-BPSK, 16QAM-BPSK, and 16QAM-QPSK. In QPSK-BPSK scenario, the QPSK modulation signal is used as the original

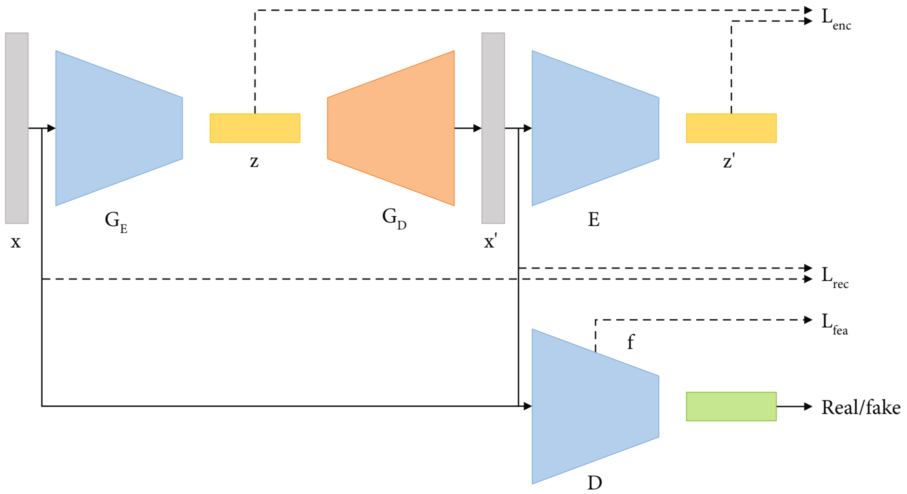


Figure 2: Loss functions of the proposed model.

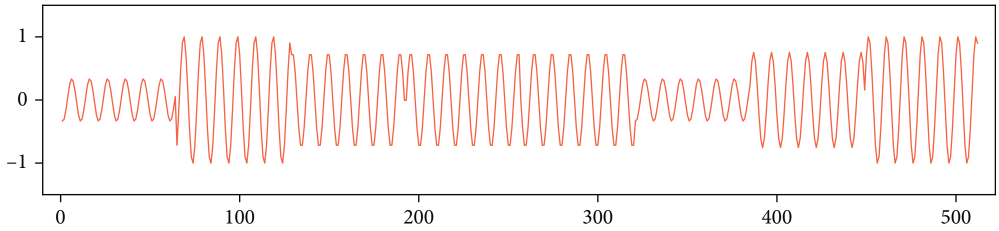


Figure 3: Original transmitted signal.

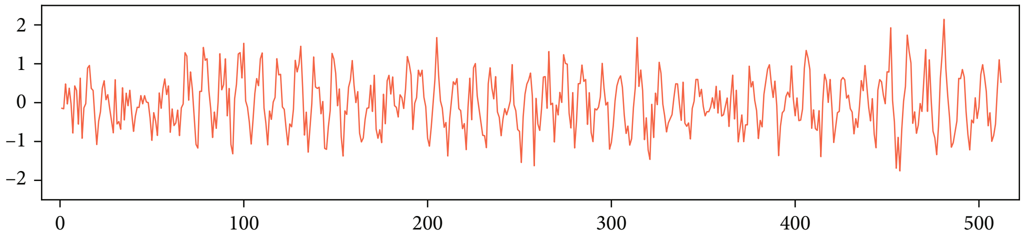


Figure 4: Normal received signal.

signal, while the BPSK modulation signal is used as the interference signal with the same frequency. In 16QAM-BPSK and 16QAM-QPSK scenarios, the generality of the proposed algorithm for different original signals and different interference signals is comprehensively verified. Moreover, the original signals in normal samples and abnormal samples are different, so as to reflect the adaptability of the algorithm. All signals are downsampled at 2 MHz. And each sample is adjusted to a fixed length of 512 sampling points.

The following figures depict the transmitted signal and the corresponding received signal when noise or interference exists in the channel, where the SNR is 2 dB. It is a 16QAM-QPSK scenario, in which a 16QAM modulation signal is overlapped by a QPSK modulation signal with the same frequency. Figure 3 depicts the original transmitted signal without noise and interference. Figure 4 depicts the normal received signal that contains only noise. Figure 5 depicts the abnormal received signal, which contains both noise and interference with the same frequency as the transmitted signal. Since the overlapped interference is not obvious in the mixture of noise and original signal, it is difficult to judge whether the interference exists by observing the waveform. The proposed algorithm directly takes the received signal as input to obtain the detection results without prior knowledge of the communication system, which makes it superior to traditional methods.

The adversarial training in the proposed model is based on the standard DCGAN without using additional skills to improve the training process. The model adopts Adam optimizer. The initial learning rate is set to 0.0002. The momentums are set to 0.5 and 0.999, respectively. The discriminator is optimized based on the binary cross entropy loss, and the generator is updated based on the equations mentioned above. The model is trained for 10 epochs on the simulated

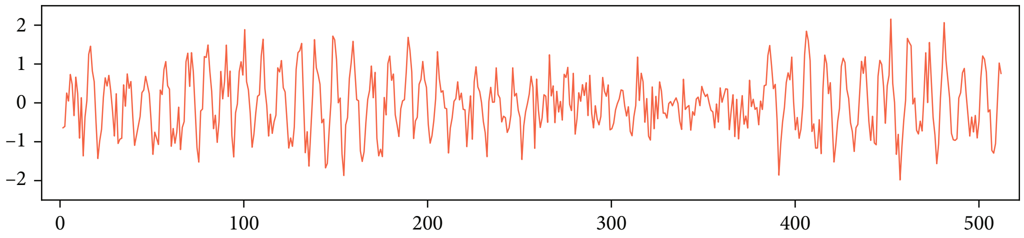


Figure 5: Abnormal received signal.

Table 1: AUC values for different SNR in QPSK-BPSK scenario.

Methods	-16	-14	-12	-10	-8	-6	-4	-2	0	2
IAE	0.517	0.583	0.639	0.672	0.742	0.819	0.869	**0.934**	**0.956**	**0.968**
IGAN	0.566	0.612	0.649	0.692	0.765	0.814	0.865	0.893	0.909	0.920
Proposed	**0.569**	**0.623**	**0.659**	**0.709**	**0.781**	**0.831**	**0.874**	0.900	0.911	0.934

Table 2: AUC values for different SNR in 16QAM-BPSK scenario.

Methods	-16	-14	-12	-10	-8	-6	-4	-2	0	2
IAE	0.549	0.567	0.618	0.718	0.763	0.848	0.886	0.896	0.920	0.929
IGAN	0.577	0.599	0.615	0.715	0.768	0.824	0.887	0.917	0.937	0.943
Proposed	**0.594**	**0.645**	**0.673**	**0.758**	**0.811**	**0.855**	**0.900**	**0.920**	**0.940**	**0.947**

Table 3: AUC values for different SNR in 16QAM-QPSK scenario.

Methods	-16	-14	-12	-10	-8	-6	-4	-2	0	2
IAE	0.506	0.554	0.639	0.698	0.771	0.820	0.881	0.909	0.920	0.929
IGAN	0.547	0.612	0.639	0.723	0.776	0.839	0.891	0.920	0.935	0.952
Proposed	**0.559**	**0.628**	**0.650**	**0.746**	**0.802**	**0.858**	**0.901**	**0.923**	**0.940**	**0.955**

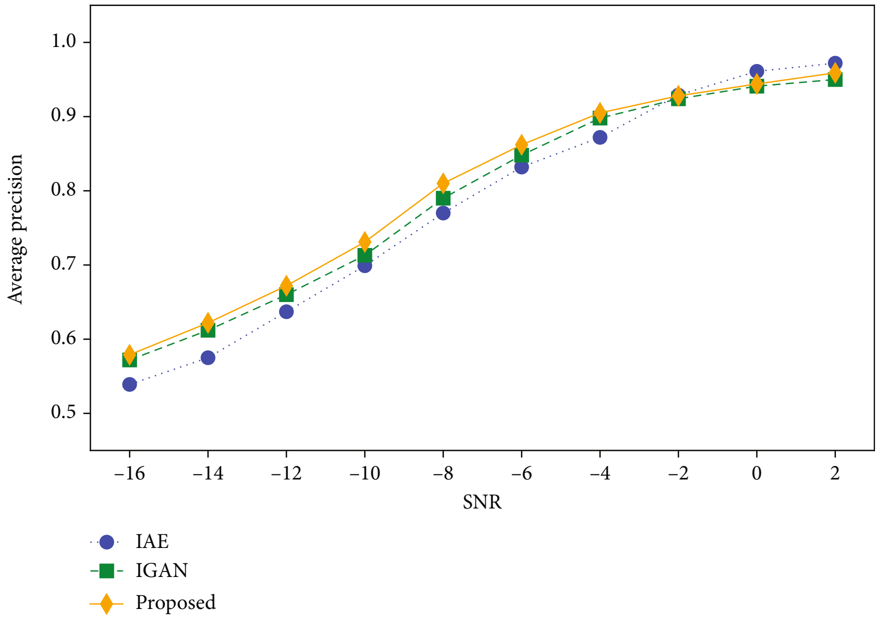


Figure 6: AP trends for different SNR in QPSK-BPSK scenario.

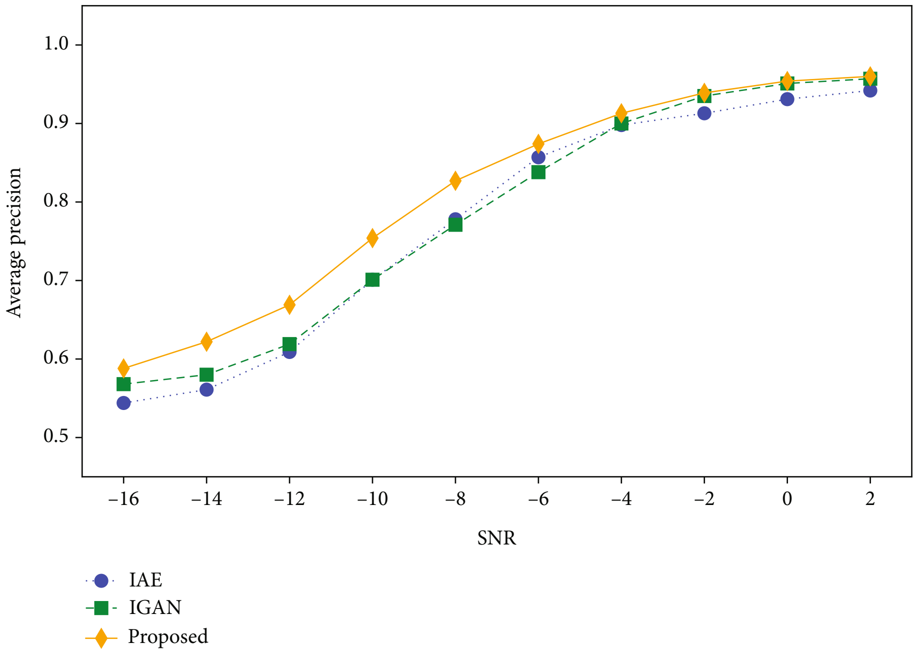


Figure 7: AP trends for different SNR in 16QAM-BPSK scenario.

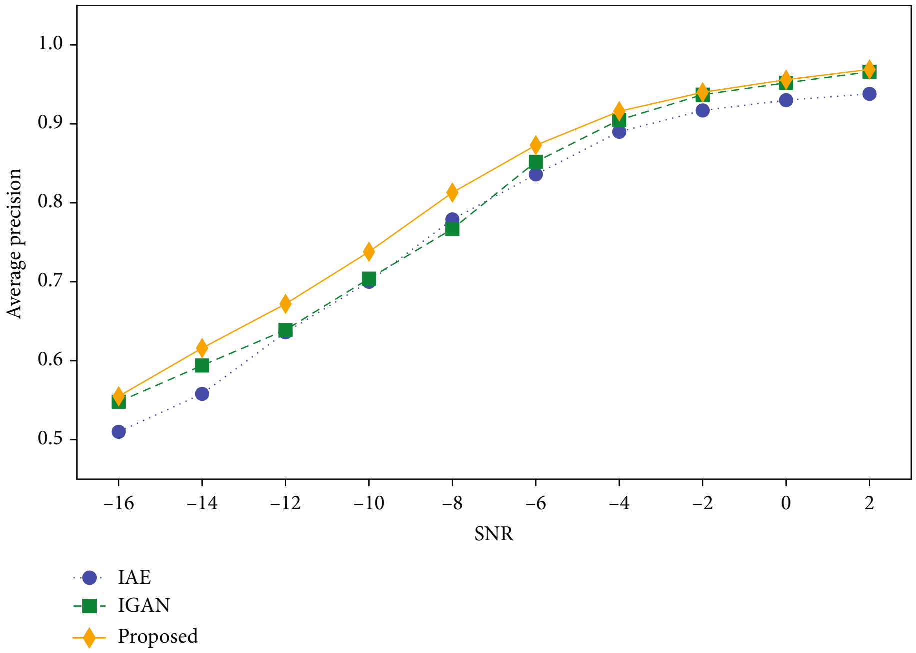


Figure 8: AP trends for different SNR in 16QAM-QPSK scenario.

dataset. All these parameters are determined by a large number of experiments to make the model produce good results.

4.2. Results. In order to investigate the detection performance of the proposed algorithm for time-frequency overlapped interference, one method based on improved autoencoder (IAE) [28] and another method based on improved generative adversarial network (IGAN) [29] are compared in this paper, which are commonly used for anomaly detection in deep learning.

Due to the particularity of interference detection, the area under curve (AUC) for receiver operating characteristic (ROC) is calculated in the experiment. ROC curve reflects the trade-off between true positive rate (TPR) and false positive rate (FPR). Each point on the curve represents the TPR-FPR value corresponding to different thresholds. Tables 1–3 list the experimental results obtained in three scenarios, so as to visually display the detection performance of the three models under different SNR conditions. It can be seen that the proposed algorithm provides higher AUC compared with other methods. Especially in the case of low SNR, the detection effect is also improved. In QPSK-BPSK scenario, the difference between the original signal and the interference signal is relatively small, but the proposed algorithm is still feasible.

Besides, the average precision (AP) is also used as an evaluation measure, the higher the better. Figures 6–8 show that the proposed algorithm achieves efficiency improvement among all methods. It can be concluded that the model fails to generate abnormal samples. Moreover, the distance in both signal space and latent space provides sufficient support for the algorithm to resist serious noise and detect sudden interference.

In summary, the above experimental results indicate that the proposed algorithm has generalization ability and performs better than other competitive methods for interference detection.

5. Conclusion

In wireless communications, signals are always affected by noise and interference during transmission, which leads to large errors in the decoding phase at the receiver. In order to improve communication quality, interference detection is an essential process. It is an important research direction to design an interference detection algorithm suitable for various communication scenarios. In other fields, the research on anomaly detection has made great progress. Interference detection in wireless communications can be regarded as a special anomaly detection, which identifies whether interference exists in the signal. In addition, deep learning has been proved to have strong feature extraction ability. Inspired by the combination of anomaly detection and deep learning, a wireless communication interference detection algorithm based on generative adversarial network is proposed in this paper. It uses the reconstruction strategy to detect time-frequency overlapped interference. The model optimizes the parameters in the way of adversarial training, and the generator adopts the structure of encoder-decoder-encoder. It can overcome the influence of noise and improve the detection accuracy in the case of low SNR. The experimental results on the simulated communication dataset show that the proposed algorithm performs better than the competitive methods based on deep learning and effectively solves the problem of interference detection.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61702543 and No. 61971439), the Natural Science Foundation of Jiangsu Province (No. BK20191329), the China Postdoctoral Science Foundation (No. 2019T120987), and the Startup Foundation for Introducing Talent of Nanjing University of Information Science and Technology (No. 2020r100).

References

[1] Q. Wu, Z. Sun, and X. Zhou, "Interference detection and recognition based on signal reconstruction using recurrent neural network," in *2019 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, Waikoloa, HI, USA, 2019.

[2] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: a survey," *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, 2016.

[3] I. J. Goodfellow, J. Pouget-Abadie, and M. Mirza, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 2672–2680, 2014.

[4] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: semi-supervised anomaly detection via adversarial training," in *Computer Vision–ACCV*, pp. 622–637, Springer, Cham, 2019.

[5] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[6] B. R. Kiran, D. M. Thomas, and R. Parakkal, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," *Journal of Imaging, Multidisciplinary Digital Publishing Institute*, vol. 4, no. 2, p. 36, 2018.

[7] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Generation Computer Systems*, vol. 55, pp. 278–288, 2016.

[8] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," *Information Processing in Medical Imaging*, pp. 146–157, 2017.

[9] B. Li, S. Zhao, R. Zhang, Q. Shi, and K. Yang, "Anomaly detection for cellular networks using big data analytics," *IET Communications*, vol. 13, no. 20, pp. 3351–3359, 2019.

[10] C. Zhang, H. Dang, Y. Xiong, and T. Yan, "Spread spectrum algorithm resistance to wideband non-stationary interference," *The Journal of Engineering*, vol. 2019, no. 21, pp. 7747–7751, 2019.

[11] P. Thakur, A. Kumar, S. Pandit, G. Singh, and S. N. Satashia, "Spectrum monitoring in heterogeneous cognitive radio network: how to cooperate?," *IET Communications*, vol. 12, no. 17, pp. 2110–2118, 2018.

[12] A. Saad, B. Staehle, and Y. Chen, "On interference detection using higher-order statistics," in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, pp. 943–947, Cambridge, UK, 2015.

[13] B. Xia and C. Bao, "Wiener filtering based speech enhancement with weighted denoising auto-encoder and noise classification," *Speech Communication*, vol. 60, pp. 13–29, 2014.

[14] L. Sun, J. Du, L. R. Dai, and C. H. Lee, "Multiple-target deep learning for LSTM-RNN based speech enhancement," in *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*, pp. 136–140, San Francisco, CA, USA, 2017.

[15] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[16] M. T. McCann, K. H. Jin, and M. Unser, “Convolutional neural networks for inverse problems in imaging: a review,” *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 85–95, 2017.

[17] T. Meinhardt, M. Moeller, C. Hazirbas, and D. Cremers, “Learning proximal operators: using denoising networks for regularizing inverse imaging problems,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1799–1808, 2017.

[18] J. Zhang, S. Hu, Z. Du, W. Wu, Y. Gao, and J. Cao, “Deep learning-based digital signal modulation identification under different multipath channels,” *IET Communications*, vol. 15, no. 15, pp. 1950–1962, 2021.

[19] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, “Autoencoder-based network anomaly detection,” in *2018 Wireless Telecommunications Symposium (WTS)*, pp. 1–5, Phoenix, AZ, USA, 2018.

[20] Y. Zhao, B. Deng, C. Shen, Y. Liu, H. Lu, and X. S. Hua, “Spatio-temporal auto encoder for video anomaly detection,” in *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1933–1941, 2017.

[21] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, “Deep-cascade: cascading 3D deep neural networks for fast anomaly detection and localization in crowded scenes,” *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1992–2004, 2017.

[22] Y. S. Chong and Y. H. Tay, “Abnormal event detection in videos using spatiotemporal autoencoder,” in *Advances in Neural Networks*, pp. 189–196, Springer, 2017.

[23] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe, “Training adversarial discriminators for cross-channel abnormal event detection in crowds,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1896–1904, Waikoloa, HI, USA, 2019.

[24] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, 2017.

[25] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative adversarial networks: an overview,” *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training GANs,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 2234–2242, 2016.

[27] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of Wasserstein GANs,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5769–5779, 2017.

[28] Z. Cheng, S. Wang, P. Zhang, S. Wang, X. Liu, and E. Zhu, “Improved autoencoder for unsupervised anomaly detection,” *International Journal of Intelligent Systems*, vol. 36, no. 12, pp. 7103–7125, 2021.

[29] X. Y. Wang, J. J. Yang, L. Zhang, Q. N. Lu, and M. Huang, “Spectrum monitoring of radio digital video broadcasting based on an improved generative adversarial network,” *Radio Science*, vol. 56, no. 8, p. e2021RS007270, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8015896, 17 pages
https://doi.org/10.1155/2022/8015896

Research Article

Proving Simulink Block Diagrams Correct via Refinement

Wei Zhang[1] **Quan Sun**[2] **Chao Wang**[1] **and Zhiming Liu**[1,3]

[1] *College of Computer and Information Science, Southwest University, Chongqing 400715, China*
[2] *College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China*
[3] *School of Software, Northwestern Polytechnical University, Xi'an 710129, China*

Correspondence should be addressed to Quan Sun; quansun@nuaa.edu.cn and Zhiming Liu; zliu@nwpu.edu.cn

Received 22 June 2022; Accepted 30 July 2022; Published 19 September 2022

Academic Editor: Chenglu Jin

Simulink is a well-known block diagram-based tool for modular design and multidomain simulation of Cyber-Physical Systems (CPS). However, the simulation by Simulink cannot completely cover the state space or behavior of a target system, which would not ensure the correctness of the developed block diagrams in Simulink. In this work, we present a contract-based method, which supports compositional reasoning and refinement, for proving the correctness of Simulink block diagrams with *discrete-time* and *continuous-time* dynamic behavior. We use the assume-guarantee contract as a specification language. The Simulink block diagrams are correct in the sense that if the block diagrams satisfy the formal specifications of the system being modeled. To prove the correctness of a block diagram, we first define semantics for Simulink block diagrams. We study three composition operators, i.e., serial, parallel, and algebraic loop-free feedback with multistep delays. We present a satisfaction relation between the block diagram and contract and present a refinement relation between the contracts. We prove that if the Simulink block diagram satisfies the composition contract and the composition contract refines the system specifications, the block diagram is correct relative to the system specifications. Furthermore, we demonstrate the effectiveness of our method via a real-world case study originating from the control system of a reservoir. Our method can also provide an idea to verify whether the designed CPS is planted with a logic bomb by attackers.

1. Introduction

Cyber-Physical Systems (CPS) are engineering systems where functionalities emerge from the network interaction of physical and computational processes [1]. Designing CPS correctly and efficiently is a critical challenge for computer science and industry. Model-based design (MBD) [2] provides virtual system integration and a visual approach to develop models for CPS. Bugs in the model can be identified and corrected at an early stage of the design process when no hardware is available. Such a method is considered as an effective solution to design CPS correctly.

Simulink [3] is a graphical modeling language for model-based design (MBD). Currently, Simulink greatly appeals to CPS engineers since it captures the dynamic behavior of the modeled system. A Simulink block diagram consists of blocks connected via wires. The blocks (from Simulink library, a set of predefined blocks that can assemble block diagrams of systems with drag-and-drop mouse operations) represent different parts of a system being modeled, and wires indicate the communication between the blocks. The blocks have input and output ports that receive the input signals and send the output signals. The signals are the functions of time that can be *continuous-time* or *discrete-time*. Hence, the Simulink block diagrams can be classified based on the time: contain only *discrete-time* blocks, *continuous-time* blocks, or a mixture of *discrete-time* and *continuous-time* blocks. Our work focuses on the Simulink block diagrams containing only *discrete-time* blocks and *continuous-time* blocks, which we call *discrete-time* Simulink block diagrams or *continuous-time* Simulink block diagrams.

Simulink supports the design, modeling, simulation, and test of CPS. The test for Simulink block diagrams is based on numerical simulation. One of the drawbacks of numerical simulation is that it does not completely cover a target system's state space or behavior. In addition, a logic bomb [4] maliciously inserted into Simulink by attackers can persistently change the behavior. In safety-critical systems, an error could lead to incorrect analysis results and thus result in property damage, even significant injury or death. Formal methods can rigorously prove that all possible behaviors satisfy a specific formal specification, thus ensuring correctness. By "correctness," we mean that all possible behaviors of the Simulink block diagram satisfy the given formal specification of a system to be modeled.

A number of methods have been reported in the literature. To the best of our knowledge, some existing solutions only focus on Simulink block diagrams with *discrete-time* behaviors, e.g. [5–11]. A common approach for tackling *continuous-time* Simulink block diagrams is to discretize the *continuous-time* dynamical behavior [12, 13]. However, the discretization of continuous systems reduces the accuracy of the verification of continuous dynamics. The contract supports compositional reasoning and refinement, enabling hierarchical design and verification of complex systems by decomposed system-level specification into the block-level specification to provide implementations correctly. Based on this advantage, we employ contract as formal specification to specify the observable trajectory of *discrete-time* and *continuous-time* blocks and present a contract-based refinement technique for proving Simulink block diagrams' correctness.

To prove the correctness of Simulink block diagrams, we first define formal semantics for Simulink block diagrams. We consider the blocks in the library to be units and call them elementary blocks. The elementary block expresses the time-dependent (*continuous-time* or *discrete-time*) relationships between the inputs, internal states, and outputs. Thus, we define the *elementary block* as a dynamic system that can model both *continuous-time* and *discrete-time* blocks. Based on this definition, we define the observable trajectory for the blocks, i.e., the evolution of the value of input-output variables over time. We formulate the wire as *unilateral connection* (i.e., the relations of output and input between connected blocks) for communication. Then, to construct the Simulink block diagrams, we define three basic composition operators, namely, serial, parallel, and algebraic loop-free feedback composition. Moreover, we study the algebraic loop-free feedback composition containing multistep delays. Similar works [13, 14] only considered the ones with unit delay.

We then present a contract-based refinement technique to prove the correctness of the Simulink block diagrams, as shown in Figure 1. Our purpose is to prove that the block diagram satisfies the system specification. To this end, we introduce a mid-level called composition contract (i.e., the composition of contracts corresponding to the blocks that construct the Simulink block diagrams) between low-level block diagrams and high-level system specifications for composition. The approach is divided into two stages. Firstly, we define the satisfaction relation that relates block to contract and verify that the satisfaction relation is preserved by composition, i.e., if blocks satisfy their contract, respectively, their composition satisfies the contract composition. Secondly, we define the refinement relation between contracts and prove that the composition of contracts refines the system specifications. The block diagram is correct as long as we prove that the block diagram satisfies the composition contract and the composition contract refines the system specification to imply that the block diagram satisfies the system specification.

1.1. Contributions. Contributions of this paper are summarized as follows:

(i) We define the formal semantics for Simulink block diagrams from the viewpoint of dynamic systems to precisely express the trajectory of the Simulink block diagrams with *discrete-time* and *continuous-time*

(ii) Under the semantics, we propose a contract-based refinement technique mentioned above for proving the correctness of the Simulink block diagrams with *discrete-time* and *continuous-time* blocks

(iii) We demonstrate the effectiveness of our method through a case study of the control system of a reservoir that is modelled with Simulink block diagrams

1.2. Organization. The remainder of this paper is organized as follows. Section 2 reviews the related works. Section 3 introduces the notations and notions used in our work. Section 4 presents the semantics of the Simulink block diagrams. Section 5 proposes the contract-based refinement method for proving the correctness of Simulink diagram blocks. Section 6 demonstrates the effectiveness of our method with a case study. Section 7 concludes the paper and proposes future works.

2. Related Works

In recent years, there are a range of methods to analyze and verify Simulink.

There exist some works that translate Simulink into other formal modeling languages. For example, Tripakis et al. [5] translated the Simulink block diagrams to a synchronous dataflow language, Lustre. Since the Lustre has a *discrete-time* semantics, the work [5] only handled *discrete-time* Simulink block diagrams. In [6], Cavalcanti et al. presented a semantics for *discrete-time* Simulink blocks diagrams called Circus. The work [6] was based on existing tools that generate CSP and Z specifications from *discrete-time* block diagrams. It only translated *discrete-time* Simulink blocks diagrams, Simulink block diagrams with *continuous-time* were not considered.

Chen and Dong [15, 16] presented the method to automatically transformed Simulink diagrams with *discrete-time* and *continuous-time* into Timed Interval Calculus (TIC) models. This method applied the Prototype Verification System (PVS) to validate that TIC fulfils requirements. These works were the first attempt to model Simulink block diagrams with *continuous-time*. The work [17] presented an operational semantics for Simulink's simulation engine that formally defines the numerical simulation result, including *discrete-time* and *continuous-time*. Zou et al. [18]

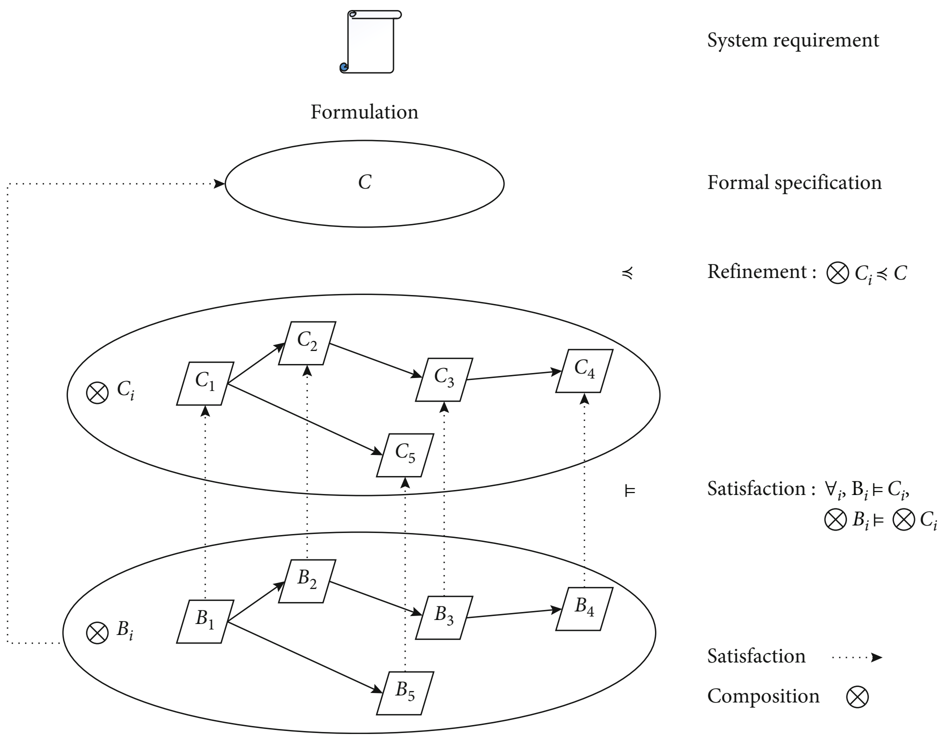


FIGURE 1: An overview of contract-based correctness proof methods for Simulink block diagrams, where the dashed line and $\vDash$: satisfaction; $\preccurlyeq$: refinement; $\otimes$: composition.

automatically translated Simulink block diagrams into Hybrid Communicating Sequential Processes (HCSP) and showed how the translated HCSP models are verified using the Hybrid Hoare Logic Verifier. In [19], it showed how different Simulink blocks can be expressed in the synchronous language Ze$'$lus, which extends a language Lustre with ODEs and zero-crossing events. The main difference between the articles [15–19] and our work is intentions. Our main goal here is not to translate the Simulink block diagrams to other formalisms, nor to define the semantics of Simulink's engine, but to directly define the trajectory of the Simulink block diagrams from the point of view of dynamic systems and provide a compositional and refinement technology to prove the correctness of Simulink block diagrams with *discrete-time* and *continuous-time*. On the other hand, we have three basic composition operators, i.e., serial, parallel, and algebraic loop-free feedback composition with multistep delays, which more facilitate the composition in construction. However, the work [18] could verify hybrid system, which is not considered in our work.

The contract-based approaches for verifying the correctness of Simulink block diagrams were also widely studied in the literature. Bostro et al. [7] showed definitions of contract and refinement using the action systems for Simulink models, while refinement provides a framework for reasoning about implementation correctness. However, this work only focuses on *discrete-time* Simulink block diagrams. Ye et al. [10, 11] defined a theoretical reasoning framework for Simulink block diagrams using Unifying Theories of Programming (UTP). The main idea of these papers is to translate each block or subsystem to a design, and the hierarchical connections of blocks are mapped to a variety of compositions of designs, and verify some properties. However, these papers only handled *discrete-time* Simulink block diagrams. In our work, we provide a compositional and refinement technology to prove the correctness of Simulink block diagrams with *discrete-time* and *continuous-time*. Dragomir et al. [13] recently presented a Refinement Calculus of Reactive Systems (RCRS) toolset for compositional formal modeling and reasoning about discrete and continuous reaction systems. RCRS is a *discrete-time* framework. The continuous systems can be modeled by discretizing time. However, the discretization of continuous systems reduces the accuracy of the verification of continuous dynamics. Our approach differs because we can directly represent and theoretically verify the correctness of *discrete-time* and *continuous-time* Simulink block diagrams. Moreover, we study the algebraic loop-free feedback composition with multistep delays.

3. Preliminary

In this section, we introduce some notations and notions that will be used in our work. We denote the set of natural numbers by $\mathcal{N}$, i.e., $\{0, 1, 2, \cdots\}$, the set of positive integers by $\mathcal{N}^+$, i.e., $\{1, 2, \cdots\}$, the set of positive real numbers $\mathcal{R}^+$, and the set of nonnegative real numbers by $\mathcal{R}_0^+$. We denote the set of integers between 1 and n by $[n] = \{1, \cdots, n\} \subset \mathcal{N}^+$. We denote vectors by bold fonts, and their components are indexed from 1 to n; for example, $\mathbf{x} = (x_1, \cdots, x_n)$, and x_i is the *i-th* component of $\mathbf{x}$, $i \in [n]$.

Following the definitions of vector addition and scalar multiplication in vector spaces, for two vectors $\mathbf{x} = (x_1, x_2)$ and $\mathbf{y} = (y_1, y_2)$, the vector addition can be expressed as $\mathbf{x} + \mathbf{y} = (x_1 + y_1, x_2 + y_2)$ and the scalar multiplication can be expressed as $k(x_1, x_2) = (kx_1, kx_2)$, where $k \in F$ and F is a field.

We fix a *time axis* $\mathscr{R}_0^+$. The *continuous-time* domain T_c is a subset of $\mathscr{R}_0^+$ with a left endpoint equal to 0. The right endpoint may be open or closed. For any $\tau \in \mathscr{R}^+$, the *discrete-time* domain is a set $T_\tau = \{t_k | t_k = k\tau, k \in \mathscr{N}\}$, where the time instant t_k is called *sample time point*, $t_0 = 0$ is the initial time, and t_k keeps increasing at every iteration, i.e., $\forall k \in \mathscr{N}, t_k < t_{k+1}$.

We denote the n-dimensional real-valued vector space by $\mathscr{R}^n$, where $n \in \mathscr{N}^+$. A *signal* is a function from a time-domain T to $Z \subseteq \mathscr{R}^n$, i.e., $\mathbf{z}(\cdot)$: $T \mapsto Z$ (or $\mathbf{z}$ for short). We use Z^T to denote a set of all signals $\mathbf{z}$. A *continuous-time* signal is a signal defined over a *continuous-time* domain T_c. A *discrete-time* signal is a signal that is defined at the *discrete-time* domain T_τ. The values of the *discrete-time* signal update at each t_k and remain constant in the intervals $\lceil k\tau, (k+1)\tau), k \in \mathscr{N}$.

4. Simulink Block Diagrams and Semantics

In this section, we present the semantics of Simulink block diagrams. We will start by giving a brief introduction to Simulink block diagrams, and we highlight the features relevant to our work. For more details, we refer the reader to [3].

4.1. Introduction of Simulink Block Diagrams. A Simulink block diagram is a graphical representation of a dynamic system. The Simulink block diagrams are composed of blocks and wires. The blocks can be either *elementary block*s provided by the Simulink library or composition blocks made up of *elementary blocks* or other composition blocks. An example that implements the relationship between the vehicle's power, resistance, and speed is shown in Figure 2. It consists of Constant, Subsystem, and Scope, where Constant and Scope are *elementary blocks*, and the composition block Subsystem comprises four *elementary blocks* Gain1, Subtract, Integrator, and Gain2.

In its most general form, the *elementary block* has *Inputports* and *Outputports* that receive the input signals and send the output signals. For some special blocks, the absence of *Inputports* or *Outputports* is also allowed. We will explain the details later. An *elementary block* is either *stateful* or *stateless*. We say a block is *stateful* if the output of this block depends on its inputs and internal states (i.e., memory). We say a block is *stateless* if the output depends only on its inputs. Wires transmit signals in the direction indicated by the arrow. It must transmit signals from *Outputports* of one block to *Inputports* of another block in terms of its sample times. An exception to this is that one wire can be drawn from another. This sends the original signal to two (or more) target blocks. Wire communication is instant. That is, when a block outputs the value to a wire, all blocks connected to that wire will see the new value simultaneously.

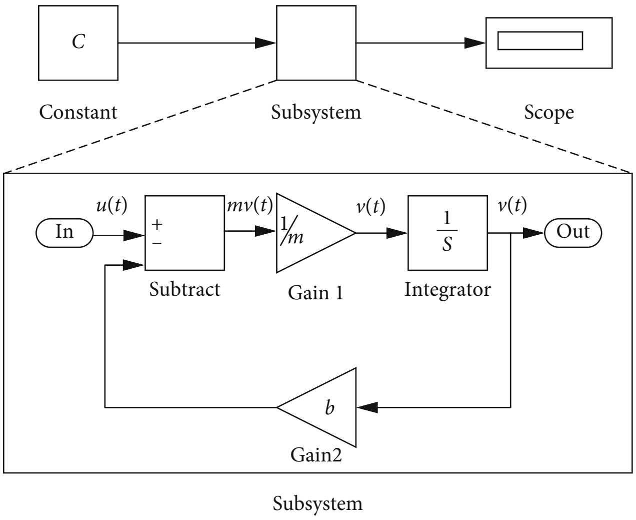


Figure 2: A Simulink block diagram.

To improve the modeling capabilities of Simulink, each *elementary block* contains some user-tunable parameters. One of the significant parameters is sample time that indicates the rate at which the block executes in simulation. According to the sample time, the blocks are divided into two main categories: continuous and discrete blocks. The sample time comprises two parameters: *sample time period* τ and *initial time offset* θ. For the continuous blocks, the *sample time period* $\tau = 0$. For the discrete blocks, the *sample time period* τ is always greater than zero and less than the simulation end time and θ less than or equal to τ. Since the default value of the initial offset is 0, unless otherwise mentioned, we let the initial offset θ be 0. As an example, suppose that the time unit is seconds, let the *sample time period* of a block be 0.02 s, and then the block updates methods (update, derivative or output) each 0.02 s. The Simulink block diagrams can be single rate where all blocks run with the same period or multirate where blocks run on different periods. This work considers single rate *discrete-time* Simulink block diagrams and *continuous-time* Simulink block diagrams with $\tau = 0$.

There are three basic composition operators in Simulink: (i) Serial composition is the composition that the output of the source block is connected to the input of the target block. (ii) Parallel composition is that two blocks are "stacked on top of each other" without any wires between the two blocks. (iii) Feedback composition is that the output of a block connects to one of its inputs. Other forms of composition can be assembled from these three basic composition operations and wires.

4.2. The Semantics of Simulink Block Diagrams. In this subsection, we formally define the semantics for Simulink block diagrams. We focus on the semantics of *elementary block*, composition semantics for composition operators, and the semantics of communication between blocks.

As introduced in 4.1, an *elementary block* has *Inputports* which receive the input signals, internal state, and *Outputports* that send the output signals. It describes a mathematical relationship between inputs, outputs, and internal states to capture dynamic behavior. We define an *elementary block* as a dynamical system.

Definition 1 (An elementary block). An *elementary block* B is a tuple $(\mathbf{x}, \mathbf{s}, \mathbf{y}, \varphi, f)$, where

(i) $\mathbf{x} : T \mapsto X$ is an input signal, $\mathbf{x}(t) \in X \subseteq \mathscr{R}^n$, $\mathbf{x}(t_0)$ is the initial value of the input signal

(ii) $\mathbf{y} : T \mapsto Y$ is an output signal, $\mathbf{y}(t) \in Y \subseteq \mathscr{R}^m$, $\mathbf{y}(t_0)$ is the initial value of the output signal

(iii) $\mathbf{s} : T \mapsto S$ is an internal state signal, $\mathbf{s}(t) \in S \subseteq \mathscr{R}^p$, $\mathbf{s}$ (t_0) is the initial value of the internal state signal

(iv) φ is the transition function of internal state

(v) f is the output function, i.e., $\mathbf{y}(t) = f(\mathbf{x}(t), \mathbf{s}(t))$

The above definition can express both *discrete-time* blocks and *continuous-time* blocks. We say that a block is continuous if it operates on *continuous-time* signals. We say a block is discrete if it operates on *discrete-time* signals. Every block must define its output function and may define initialize, update, or derivative function to realize the corresponding function. For the *discrete-time* blocks, we denote $\varphi = \varphi_u$, i.e., $\mathbf{s}(t_{k+1}) = \varphi_u(\mathbf{x}(t_k), \mathbf{s}(t_k))$, which refers to the update function of internal state. For the *continuous-time* blocks, we denote $\varphi = \varphi_d$, i.e., $\dot{\mathbf{s}}(t) = \varphi_d(\mathbf{x}(t), \mathbf{s}(t))$, which refers to the derivative function of internal state. The input or internal state of a block can be empty, respectively. In that case, we denote the input or internal state by $\mathbf{x} = \mathbf{0}$ or $\mathbf{s} = \mathbf{0}$ and use a symbol "-" to denote the transition function of the internal state. If a block has no internal state, we say the block is *stateless*.

Some *elementary blocks* mentioned in our article are shown in Table 1 Three simple examples are shown below.

Example 1 (Unitdelay). An example of a stateful *discrete-time* elementary block is the Unitdelay block. The Unitdelay expresses that the current output value of this block is equal to the value of the current internal state, and the value of the next internal state is equal to the current input value. It can be represented as $B_u = (\mathbf{x}_u, \mathbf{s}_u, \mathbf{y}_u, \mathbf{s}_u(t_{k+1}) = \mathbf{x}_u(t_k), \mathbf{y}_u(t_k) = \mathbf{s}_u(t_k))$, where $t_k \in T_\tau$.

Example 2 (Multistep delays). A block with multistep delays is denoted as $B = (\mathbf{x}, \mathbf{s}, \mathbf{y}, \varphi, f)$, where $\mathbf{y}(t_{n\cdot k}) = \mathbf{s}(t_{n\cdot k})$, and $\mathbf{s}$ $(t_{n\cdot(k+1)}) = \mathbf{x}(t_{n\cdot k})$, where n means that the block outputs the input of this block after n sample periods, $n \in \mathscr{N}^+, k \in \mathscr{N}$.

Example 3 (Integrator). An example of a stateful *continuous-time* elementary block is the Integrator. The block models the relations, $\dot{\mathbf{s}}(t) = \mathbf{x}(t)$ with $y(t) = s(t)$. The Integrator can be represented as $B_I = (\mathbf{x}_I, \mathbf{s}_I, \mathbf{y}_I, \dot{\mathbf{s}}_I(t) = \mathbf{x}_I(t), \mathbf{y}_I(t) = \mathbf{s}_I(t))$, where $t \in \mathscr{R}_0^+$.

We use the observable trajectory to model the evolution of the input-output variables.

Definition 2 (The observable trajectory of the *discrete-time* elementary block). Let $B = (\mathbf{x}, \mathbf{s}, \mathbf{y}, \varphi_u, f)$ be a *discrete-time* elementary block. An observable trajectory Tr of B is a set $\{(\mathbf{x}, \mathbf{y}) \colon T_\tau \mapsto X \times Y | \exists \mathbf{s} \mathbf{y}(t_k) = f(\mathbf{s}(t_k), \mathbf{x}(t_k)) \wedge \mathbf{s}(t_{k+1}) = \varphi_u(\mathbf{x}$ $(t_k), \mathbf{s}(t_k)), k \in \mathscr{N}\}$, where $\mathbf{x} : T_\tau \mapsto X \subseteq \mathscr{R}^n$ is the input trajectory and $\mathbf{y} : T_\tau \mapsto Y \subseteq \mathscr{R}^m$ *is* the output trajectory.

Definition 3 (The observable trajectory of the *continuous-time* elementary block). Let $B = (\mathbf{x}, \mathbf{s}, \mathbf{y}, \varphi_d, f)$ be a *continuous-time* elementary block. An observable trajectory Tr of B is a set $\{(\mathbf{x}, \mathbf{y}) \colon T_c \mapsto X \times Y | \exists \mathbf{s} \mathbf{y}(t) = f(\mathbf{s}(t), \mathbf{x}(t)) \wedge \dot{s}(t) = \varphi_d(\mathbf{x}(t), \mathbf{s}(t))\}$, where $\mathbf{x} : T_c \mapsto X \subseteq \mathscr{R}^n$ is the input trajectory and $\mathbf{y} : T_c \mapsto Y \subseteq \mathscr{R}^m$ is the output trajectory.

The wires connect some *Outputports* of a block to some *Inputports* of other block for communicating. For any blocks, $B_1 = (\mathbf{x}_1, \mathbf{s}_1, \mathbf{y}_1, \varphi_1, f_1)$ and $B_2 = (\mathbf{x}_2, \mathbf{s}_2, \mathbf{y}_2, \varphi_2, f_2)$, where $\mathbf{x}_1 = (x_{1,1}, \cdots, x_{1,n})$, $\mathbf{y}_1 = (y_{1,1}, \cdots, y_{1,m})$, $\mathbf{x}_2 = (x_{2,1}, \cdots, x_{2,n}$ $)$, and $\mathbf{y}_2 = (y_{2,1}, \cdots, y_{2,m})$. We model the wires between B_1 and B_2 as a relation, called unilateral connection. A unilateral connection is a set of variables pair $(y_{1,j}, x_{2,i})$, for $j \in [$ $m]$ and $i \in [n]$, where the former of each pair is the output variable that comes from B_1, the latter of each pair is the input variable that comes from B_2. We define the unilateral connection as follows:

Definition 4 (Unilateral connection). Given two blocks B_1 $= (\mathbf{x}_1, \mathbf{s}_1, \mathbf{y}_1, \varphi_1, f_1)$ and $B_2 = (\mathbf{x}_2, \mathbf{s}_2, \mathbf{y}_2, \varphi_2, f_2)$, we define a unilateral connection from B_1 to B_2 (and vice versa) as a relation $\rho = \{(y_{1,j}, x_{2,i}) | j \in [m], i \in [n]\}$ satisfying that:

(i) $y_{1,j}$ is the *j-th* component of $\mathbf{y}_1$, for all $j \in [m]$

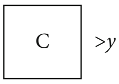
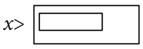
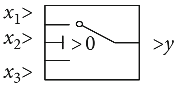
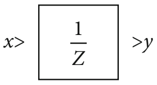
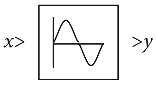
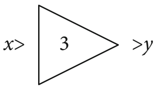
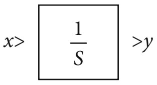
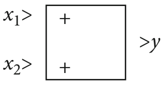
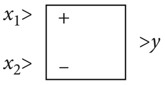
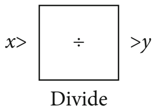
(ii) $x_{2,i}$ is the *i-th* component of $\mathbf{x}_2$, for all $i \in [n]$

(iii) $y_{1,j}(t) = x_{2,i}(t)$, for all $t \in T$

Note that not all the blocks can be connected. Given a source block $B_1 = (\mathbf{x}_1, \mathbf{s}_1, \mathbf{y}_1, \varphi_1, f_1)$ and a target block $B_2 =$ $(\mathbf{x}_2, \mathbf{s}_2, \mathbf{y}_2, \varphi_2, f_2)$, if we connect the output variables of the source block with input variables of the target block, variable names and their types (the sets of values that a variable can take) impose some constraints: First, the names of the input and output variables should not conflict. Second, the types should match and $Y_1 \subseteq X_2$, where Y_1 is the type of $\mathbf{y}_1$ and X_2 is the type of $\mathbf{x}_2$. Third, one output port can connect to many input ports, but an input port can connect to at most one output ports.

For readability, we use $\{\rho_1, \cdots, \rho_n\} \subseteq \rho$ to denote a set of specific unilateral connection (determined by a relation as defined in Definition 4). Note that, since $\rho = \{(y_{1,j}, x_{2,i}) | j \in$

Table 1: The representation of some *elementary blocks* and their semantics.

Library	Description	Elementary block	Semantics
Source	Constant Constant value		$B = (\mathbf{0}, \mathbf{0}, \mathbf{y}, -, \mathbf{y}(t) = c)$
Sinks	Display System output		$B = (\mathbf{x}, \mathbf{0}, \mathbf{y}, -, \mathbf{y}(t) = \mathbf{x}(t))$
Signal routing	Switch Conditional statement		$B = (\mathbf{x}, \mathbf{0}, \mathbf{y}, -, f)$ $\mathbf{y}(t) = \begin{cases} \mathbf{x}_1(t) & \mathbf{x}_2(t) > 0, \\ \mathbf{x}_3(t) & \mathbf{x}_2(t) \leq 0 \end{cases}$
Discrete	Unitdelay Discrete-time delay		$B = (\mathbf{x}, \mathbf{s}, \mathbf{y}, \varphi, f)$ $\varphi : \mathbf{s}(t_{k+1}) = \mathbf{x}(t_k)$ $f : \mathbf{y}(t_k) = \mathbf{s}(t_k)$
Discrete/continuous	Sine wave Discrete-time		$B = (\mathbf{x}, \mathbf{0}, \mathbf{y}, -, y(t_k) = \sin x(t_k))$
Math operations	Gain Math operation		$B = (\mathbf{x}, \mathbf{0}, \mathbf{y}, -, \mathbf{y}(t) = 3\mathbf{x}(t))$
Continuous	Integrator Continuous-time		$B = (\mathbf{x}, \mathbf{s}, \mathbf{y}, \varphi, f)$ $\varphi : \dot{\mathbf{s}}(t) = \mathbf{x}(t), f : y(t) = s(t)$
Math operations	Sum Math operation		$B = (\mathbf{x}, \mathbf{0}, \mathbf{y}, -, f), \mathbf{x} = (x_1, x_2)$ $\mathbf{y}(t) = \mathbf{x_1}(t) + \mathbf{x_2}(t))$
Math operations	Subtraction Math operation		$B = (\mathbf{x}, \mathbf{0}, \mathbf{y}, -, f), \mathbf{x} = (x_1, x_2), \mathbf{y}(t) = x_1(t) - x_2(t))$
Math operations	Divide Math operation		$B = \left(\mathbf{x}, \mathbf{0}, \mathbf{y}, -, \mathbf{y}(t) = \frac{1}{x(t)}\right)$

$[m], i \in [n]\}$ is a relation, we can also have $\rho(y_{1,j}) = \{x_{2,i} | (y_{1,j}, x_{2,i}) \in \rho, j \in [m], i \in [n]\}$.

4.3. Composition. A Simulink block diagram is a composition block constructed by *elementary blocks* according to composition operators. We will define the semantics of three basic composition operators in the following, namely, serial, parallel, and algebraic loop-free feedback composition with multistep delays. We first consider the serial composition.

We can compose blocks B_1 and B_2 to form a serial composition when there is a *unilateral connection* between B_1 and B_2 and the *unilateral connection* satisfies the connection rules, as shown in Figure 3(a).

Definition 5 (Serial composition). Given blocks $B_1 = (\mathbf{x}_1, \mathbf{s}_1, \mathbf{y}_1, \varphi_1, f_1)$ and $B_2 = (\mathbf{x}_2, \mathbf{s}_2, \mathbf{y}_2, \varphi_2, f_2)$, a serial composition $B_1 ; B_2$ *of* B_1 and B_2 is defined as a block $(\mathbf{x}, \mathbf{s}, \mathbf{y}, f, \varphi)$ if there exists a unilateral connection and

$$
\begin{aligned}
&\mathbf{x} \coloneqq \mathbf{x}_1, \\
&\mathbf{s} \coloneqq (\mathbf{s}_1, \mathbf{s}_2), \\
&\mathbf{y} \coloneqq \mathbf{y}_2, \\
&\varphi \coloneqq (\varphi_1(\mathbf{x}_1, \mathbf{s}_1), \varphi_2(\mathbf{x}_2, \mathbf{s}_2)), \\
&f \coloneqq (f_1 ; \rho ; f_2)(\mathbf{x}, \mathbf{s}) = f_2(\rho(f_1(\mathbf{x}_1, \mathbf{s}_1)), \mathbf{s}_2).
\end{aligned}
$$

The definition of the serial composition of *elementary blocks* coincides with the definition of an *elementary block* (i.e., Definition 1), and therefore a composition can be considered a block.

Example 4 (Serial composition). In Figure 3(b), the stateless block Gain takes input $\boldsymbol{x}_1$, computes $k \in \mathscr{R}$ times of $\mathbf{x_1}$, and returns $\mathbf{y}_1$ as output, where $\mathbf{x}_1 : \mathscr{R} \mapsto \mathscr{R}$ and $\mathbf{y}_1 : \mathscr{R} \mapsto \mathscr{R}$.

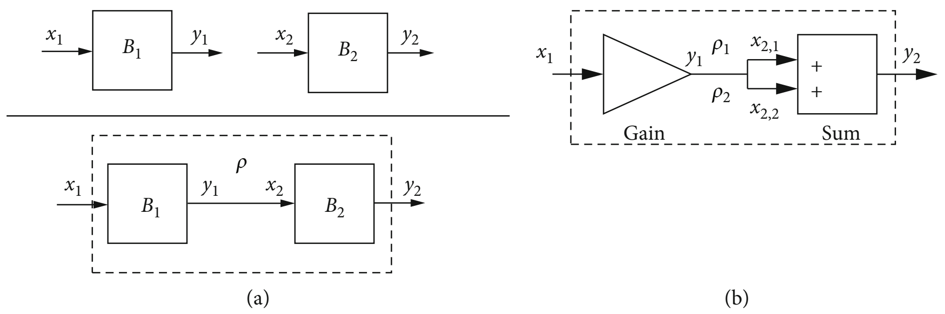


Figure 3: (a) Graphical representation of the serial composition of blocks, where $\rho = \{(y_1, x_2)\}$ is a *unilateral connection* between B_1 and B_2, the dashed box indicates a composition of blocks. (b) Graphical representation of an example of serial composition, where $\rho = \{(y_1, x_{2,1})$ $, (y_1, x_{2,2})\}$ is a *unilateral connection* between Gain and Sum.

Another stateless block Sum has the input $\boldsymbol{x}_2 = (x_{2,1}, x_{2,2})$ and output $\boldsymbol{y}_2(t) = x_{2,1}(t) + x_{2,2}(t)$, where $\mathbf{x}_2 : \mathscr{R} \mapsto \mathscr{R} \times \mathscr{R}$ and $\mathrm{y}_2 : \mathscr{R} \mapsto \mathscr{R}$, $\forall t \in \mathscr{R}$. *Let* ρ be the unilateral connection between Gain and Sum. Then, we can express these blocks as follows:

$$\begin{aligned}\text{Gain} &\coloneqq (\mathbf{x}_1, \mathbf{0}, \mathbf{y}_1, -, \mathbf{y}_1(t) = \text{gain}(x_1(t), \mathbf{0})),\\ \rho &\coloneqq \{(y_1, x_{2,1}), (y_1, x_{2,2})\},\\ \text{Sum} &\coloneqq (\mathbf{x}_2, \mathbf{0}, \mathbf{y}_2, -, y_2(t) = \text{add}(\mathbf{x}_2(t), \mathbf{0})),\end{aligned}$$

where $y_2(t) = \text{add}(\mathbf{x_2}(\mathbf{t}), \mathbf{0}) = x_{2,1}(t) + x_{2,2}(t)$, $y_1(t) = \text{gain}(x_1$ $(t), \mathbf{0}) = k \cdot x_1(t)$, $y_1(t) = x_{2,1}(t), y_1(t) = x_{2,2}(t)$. According to Definition 5, we compute the output function of the serial composition as follows:

$$\begin{aligned}f(\mathbf{x}_1(t), \mathbf{0}) &\coloneqq \text{add}(\rho(\text{gain}(\mathbf{x}_1(t), \mathbf{0})))\\ &= \text{add}(\rho(\mathbf{y}_1(t)), \mathbf{0}) = \text{add}(\mathbf{x}_2(t), \mathbf{0})\\ &= x_{2,1}(t) + x_{2,2}(t) = 2k \cdot x_1(t).\end{aligned}$$

Therefore, we have $\text{Gain}\,;\text{Adder} \coloneqq (\mathbf{x}_1, \mathbf{0}, \mathbf{y}_2, -, \mathbf{y}_2(t) = 2k$ $\cdot \mathbf{x_1}(t))$.

The serial composition of blocks satisfies associative laws.

Lemma 6 (Associativity). *Given blocks B_1, B_2, and B_3, we have $(B_1\,; B_2)\,; B_3 = B_1\,; (B_2\,; B_3)$.*

Proof. It is easy to verify using Definition 5. Thus, if we compose multiple blocks in serial, we can first compose two blocks, compose the result with a third one, $\cdots$, and rearranging the brackets in the expression does not change the result as long as the block's position remains the same. □

Parallel composition is a particular case of composition with connection, where the *unilateral connection* between B_1 and B_2 is an empty set. We define parallel composition as follows.

Definition 7 (Parallel composition). Given blocks $B_1 = (\boldsymbol{x}_1,$ $\boldsymbol{s}_1, \boldsymbol{y}_1, \varphi_1, f_1)$ and $B_2 = (\boldsymbol{x}_2, \boldsymbol{s}_2, \boldsymbol{y}_2, \varphi_2, f_2)$, we define the parallel composition $B_1 \| B_2$ of B_1 and B_2 as a block $(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{y}, f, \varphi)$, where

$$\begin{aligned}\mathbf{x} &\coloneqq (\mathbf{x}_1, \mathbf{x}_2),\\ \mathbf{s} &\coloneqq (\mathbf{s}_1, \mathbf{s}_2),\\ \mathbf{y} &\coloneqq (\mathbf{y}_1, \mathbf{y}_2),\\ \varphi &\coloneqq (\varphi_1 \| \varphi_2)((\mathbf{x}_1, \mathbf{s}_1), (\mathbf{x}_2, \mathbf{s}_2))\\ &= (\varphi_1(\mathbf{x}_1, \mathbf{s}_1), \varphi_2(\mathbf{x}_2, \mathbf{s}_2)),\\ f &\coloneqq (f_1 \| f_2)((\mathbf{x}_1, \mathbf{s}_1), (\mathbf{x}_2, \mathbf{s}_2))\\ &= (f_1(\mathbf{x}_1, \mathbf{s}_1), f_2(\mathbf{x}_2, \mathbf{s}_2)).\end{aligned}$$

Example 5 (Parallel composition). Consider the parallel composition shown in Figure 4(a). The block Divide models the relation $\mathbf{y}_1(t) = 1/\mathbf{x}_1(t)$, where $\mathbf{x}_1$ is the input signal and requires $x_1(t) \neq 0$, $\mathbf{y}_1$ is the output signal, and t represents time. The Sine Wave block models the relation $\mathbf{y}_2(t) = \mathbf{x}_2(t$ $)$, where $\mathbf{x}_2$ is the input variable and $\boldsymbol{y}_2$ is the output variable. Therefore, the Divide and Sine Wave can be represented as

$$\text{Divide} \coloneqq \left(\mathbf{x}_1, \mathbf{0}, \mathbf{y}_1, -, \mathbf{y}_1(t) = \frac{1}{\mathbf{x}_1(t)}\right),$$

$$\text{Sine Wave} \coloneqq (\mathbf{x}_2, \mathbf{0}, \mathbf{y}_2, -, \mathbf{y}_2(t) = \mathbf{x}_2(t)).$$

Following Definition 7, we write the parallel composition as follows:

$$\text{Divide}\|\text{Sine Wave} \coloneqq \left((\mathbf{x}_1, \mathbf{x}_2), \mathbf{0}, (\mathbf{y}_1, \mathbf{y}_2), -, \left(\mathbf{y}_1(t) = \frac{1}{\mathbf{x}_1(t)}, \mathbf{y}_2(t) = \mathbf{x}_2(t)\right)\right).$$

Note that Definition 5 defines the case that all the *Outputports* of B_1 match the *Inputports* of B_2. However, not all *Outputports* of B_1 match the *Inputports* of B_2 and not all *Outputports* of B_1 are connected to all the *Inputports* of B_2. As an example, we consider Figure 4(b). To handle this composition, we introduce a particular *elementary block Id* representing its output is identical to its input. We model the wires that connect the $y_{1,1}$ and $x_{2,2}$ as Id_1 and Id_2, respectively. We then denote the composition by $B = ((B_1 \| \text{Id}_1)\,;$ $\rho\,; (B_1 \| \text{Id}_2))$, where $\rho = \{(y_{1,1}, x_{Id_2}), (y_{1,2}, x_{2,1}), (y_{Id_1}, x_{2,2})\}$.

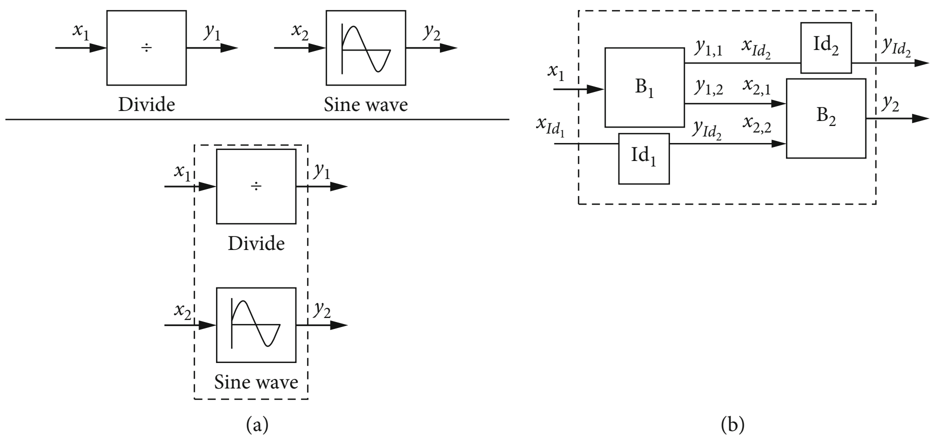


Figure 4: (a) Graphical representation of parallel composition, where the dashed box indicates a composition of blocks. (b) Graphical representation of a composition with serial and parallel, where the dashed box indicates a composition of blocks.

The parallel composition of blocks satisfies commutative law and associative law.

Lemma 8. *Given blocks B_1, B_2 and B_3, we have*

(i) $B_1 \| B_2 = B_2 \| B_1$

(ii) $(B_1 \| B_2) \| B_3 = B_1 \| (B_2 \| B_3)$

Proof. Straightforward from the Definition 7. □

We will next define the algebraic loop-free feedback composition with multistep delays. Before defining that, we first introduce the algebraic loop feedback to aid this definition. In Simulink, an algebraic loop occurs when an input port of a direct feedthrough block is driven by the output of the same block in the same time step. The direct feedthrough means that a stateless block computes its output only depending on the input value at the current time. In mathematics, an algebraic loop can be expressed as the algebraic equation $f(\mathbf{x}, \mathbf{y}) = 0$, where $\mathbf{x}$ is the input variable and $\mathbf{y}$ is the algebraic variable. Simulink solves the algebraic equation for $\mathbf{y}$ at each time instant.

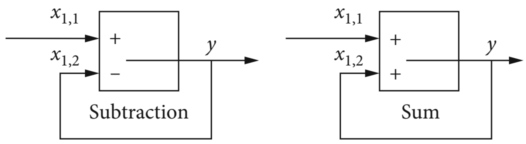
A simple example of an algebraic loop is the feedback that represents in Figure 5(a). The feedback is a Subtraction block with input $\mathbf{x} = (x_{1,1}, x_{1,2})$ and an output $\mathbf{y}$. The first element $x_{1,1}$ of input is used to communicate with the environment. The second element $x_{1,2}$ is used for feedback. The output $\mathbf{y}$ is split into two equal signals: one is for output, and another is to feed the output back into input $x_{1,2}$. Then, this loop implies that the output of the Subtraction block is an algebraic variable y that is constrained to equal the first input $x_{1,1}$ minus $x_{1,2}$, i.e., $x_{1,1}(t) - x_{1,2}(t) = y(t)$. Let $x_{1,2}(t) = y(t)$ and $x_{1,1}(t) = u(t)$, we have $u(t) - y(t) = y(t)$, and then the solution of this loop is $y(t) = u(t)/2$.

However, the algebraic loop has inherent difficulties in solving: (1) while Simulink solver solves the algebraic loop, the simulation can execute slowly. (2) Some algebraic loops have no solution (an example shown in Figure 5(b), the expression of this loop is $u(t) + y(t) = y(t)$, etc. These problems lead to algebraic loops that are undesirable. To remove the algebraic loop, according to Simulink, we connect $\mathbf{y}$ to $x_{1,2}$ by *stateful* blocks (such as Delay, Memory, or Integrator) in this feedback to break the algebraic loop. We refer to this kind of acyclic structure as algebraic loop-free feedback, as shown in Figure 6. The work [14] defined the feedback with Unitdelay. We will handle the case of multistep delays.

Definition 9 (Algebraic loop-free feedback composition with multistep delays). Let blocks $B_1 = (\mathbf{x}_1, \mathbf{s}_1, \mathbf{y}_1, \varphi_1, f_1)$ be feedback with an algebraic loop and $B_2 = (\mathbf{x}_2, \mathbf{s}_2, \mathbf{y}_2, \varphi_2, f_2)$ be a multistep delays block. Let $\rho_1 = (y_2, x_{1,2})$ and $\rho_2 = (y, x_2)$ be the unilateral connections between B_1 and B_2. We denote by $B_1 \otimes_f B_2$ the algebraic loop-free feedback composition of B_1 and B_2 with multistep delays and define $B_1 \otimes_f B_2 = (\mathbf{x}, \mathbf{s}, \mathbf{y}, \varphi, f)$, where

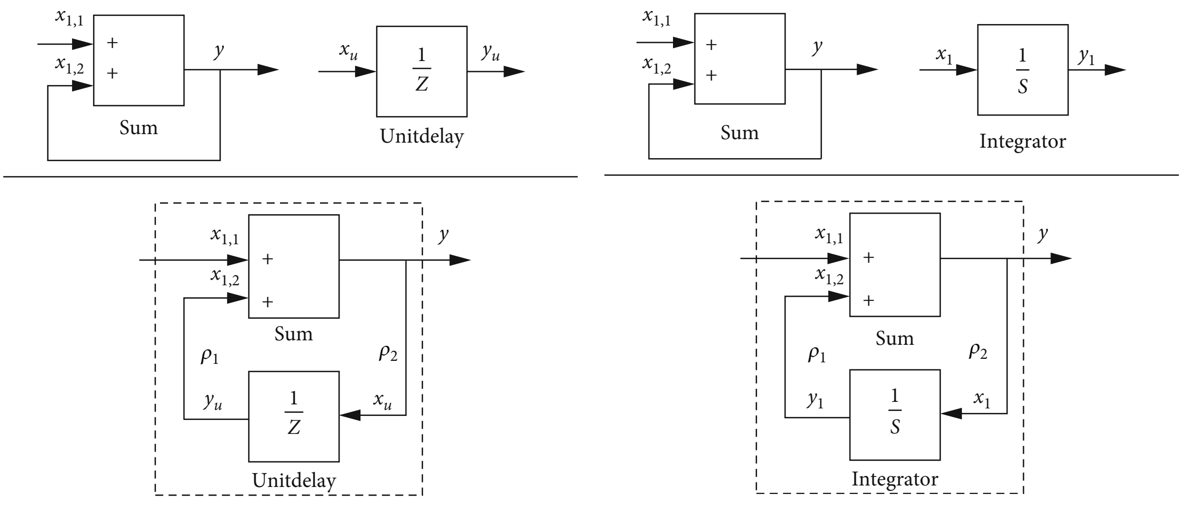
$$\begin{aligned}
\mathbf{x} &:= x_{1,1}, \\
\mathbf{s} &= (s_1, s_2), \\
\mathbf{y} &= y_1, \\
\varphi &:= (\varphi_1 \| \varphi_2)((x_1, s_1), (x_2, s_2)) \\
&\quad = (\varphi_1(x_1, s_1), \varphi_2(x_2, s_2)), \\
f &:= \begin{cases} y(t_0) & t = t_0, \\ y(t_{n\cdot k}) & t = t_{n\cdot k}, n, k \in \mathcal{N}^+. \end{cases}
\end{aligned}$$

In the definition above, the calculation process starts with the multistep delays block. When $t = t_0$, let the initial state s(t_0) $= (0, s_2(t_0))$. The output's initial value of this algebraic loop-free feedback $y(t_0) = f_1((x_{1,1}(t_0), x_{1,2}(t_0)), \mathrm{s}(t_0))$. There exists ρ_1, s.t. $x_{1,2}(t_0) = \rho_1(y_2(t_0))$, when $y_2(t_0) = f_2(x_2(t_0), s_2(t_0))$. Given a delay step $n \in \mathcal{N}^+$, when $t = t_{n\cdot k}$, where $k \in \mathcal{N}$, the output $y(t_{n\cdot k}) = f_1((x_{1,1}(t_{n\cdot k}), x_{1,2}(t_{n\cdot k})), \mathbf{s}(t_{n\cdot k}))$, where the internal state $\mathbf{s}(t_{n\cdot k}) = (0, s_2(t_{n\cdot k}))$, and $x_{1,2}(t_{n\cdot k}) = \rho_1(y_2($



(a) (b)

Figure 5: Graphical representation of feedback.



(a) (b)

Figure 6: (a) Graphical representation of algebraic loop-free feedback composition with Unitdelay. (b) Graphical representation of algebraic loop-free feedback composition with Integrator.

$t_{n\cdot k})$), where $y_2(t_{n\cdot k}) = f_2(x_2(t_{n\cdot k}), s_2(t_{n\cdot k}))$ is the output of B_2. There exists $\rho_2, s.t. x_2(t_{n\cdot k}) = \rho_2(y(t_{n\cdot(k-1)}))$ is the input of B_2, and $s_2(t_{n\cdot k}) = \varphi_2(x_2(t_{n\cdot(k-1)}), s_2(t_{n\cdot(k-1)}))$.

We next give an example to illustrate removing an algebraic loop by introducing Unitdelay, i.e., let the delay step $n = 1$.

Example 6 (An algebraic loop-free feedback composition with Unitdelay). In Figure 6(a), the Sum block Sum $= ((x_{1,1}, x_{1,2}), \mathbf{0}, \boldsymbol{y}_1, -, y_1(t) = x_{1,1}(t) + x_{1,2}(t))$ is feedback with an algebraic loop. We connect $\mathbf{y}$ *to* $x_{1,2}$ in this feedback by the Unitdelay to break the algebraic loop. The Unitdelay block $B_u = (\mathbf{x}_u, \mathbf{s}_u, \mathbf{y}_u, \mathbf{s}_u(t_{k+1}) = \mathbf{x}_u(t_k), \mathbf{y}_u(t_k) = \mathbf{s}_u(t_k))$, where $t_k \in T_\tau$. Let $\rho_1 = (y_u, x_{1,2})$ and $\rho_2 = (y, x_u)$, the ρ_1 and ρ_2 satisfy the connection rules, and then an algebraic loop-free feedback composition with Unitdelay is a block $B_1 \otimes_f B_2 = (\mathbf{x}, \mathbf{s}, \mathbf{y}, \varphi, f)$, where $\mathbf{x} = x_{1,1}, \boldsymbol{s} = (\mathbf{0}, \mathbf{s}_u), \mathbf{y} = y$, and the output function

$$y(t) = \begin{cases} y(t_0) & t = t_0, \\ y(t_k) & t = t_k, k \in \mathcal{N}^+. \end{cases}$$

When $t = t_0$, let $s_u(t_0)$ be the initial value of the internal state of the Unitdelay. We have $y_u(t_0) = s_u(t_0)$ and $x_{1,2}(t_0) = y_u(t_0)$. Hence, we get the initial output value of this composition $y(t_0) = x_{1,1}(t_0) + s_u(t_0)$.

When $t = t_k, k \in \mathcal{N}^+$, we first consider Unitdelay. Because of $x_u(t_k) = y(t_k)$, $\mathbf{s}_u(t_{k+1}) = \mathbf{x}_u(t_k), \mathbf{y}_u(t_k) = \mathbf{s}_u(t_k)$, and $x_{1,2}(t_k) = y_u(t_k)$, we have $y(t_k) = x_{1,1}(t_k) + y(t_{k-1})$.

Next, we give an example about algebraic loop-free feedback composition with Integrator.

Example 7 (An algebraic loop-free feedback composition with Integrator). An example of continuous time algebraic loop-free feedback composition with Integrator is shown in Figure 6(b). As previously mentioned, the Sum block Sum $= ((x_{1,1}, x_{1,2}), \mathbf{0}, \mathbf{y}_1, -, y_1(t) = x_{1,1}(t) + x_{1,2}(t))$ is feedback with an algebraic loop. The Integrator can be represented as $B_I = (\mathbf{x}_I, \mathbf{s}_I, \mathbf{y}_I, \dot{\mathbf{s}}_I(t) = \mathbf{x}_I(t), y_I(t) = \boldsymbol{s}_I(t))$, where $t \in \mathcal{R}_0^+$. In this composition, there exist unilateral connections $\rho_1 = (y_I, x_{1,2})$ and $\rho_2 = (y, x_I)$, and the unilateral connections satisfy the connection rules, and then an algebraic loop-free feedback composition with Integrator is a block $B_1 \otimes_f B_2 = (\mathbf{x}, \mathbf{s}, \mathbf{y}, \varphi, f)$, where $\boldsymbol{x} = x_{1,1}, \boldsymbol{s} = (\mathbf{0}, \boldsymbol{s}_I)$. Let $s_I(0)$ be the initial value of the internal state of the Integrator. We have $y_I(0) = s_I(0)$, and there is a ρ_1, s.t. $x_{1,2}(0) = y_I(0)$. Hence, we get $y(0) = x_{1,1}(0) + x_{1,2}(0) = x_{1,1}(0) + y_I(0)$.

When $t \in \mathcal{R}^+$, similarly, we first consider Integrator. Because there is a ρ_2, s.t. $x_I(t) = y(t)$, and $\dot{s}_I(t) = x_I(t), y_I(t)$

$= s_I(t)$. There exists ρ_1, s.t. $x_{1,2}(t) = y_I(t)$, we get

$$y(t) = x_{1,1}(t) + x_{1,2}(t) = x_{1,1}(t) + \left(y_I(0) + \int_0^t x_I(\delta)d\delta\right)$$

5. A Contract-Based Refinement Approach

The correctness refers to the trajectories of Simulink block diagrams that should satisfy the requirement specifications of the system. Since contracts are centered around trajectories, they are expressive and versatile enough to specify *discrete-time* and *continuous-time* blocks. We use the contract as a specification language to formalize and prove system requirements and specify the trajectories of blocks. In this section, we present a generic contract-based method for proving the correctness of Simulink block diagrams with *discrete-time* and *continuous-time* blocks.

Because the requirement specifications of a system are informal, we first formalize system requirement specifications into system contract specifications. We construct the Simulink block diagram according to the contract specification. We aim to prove that the Simulink block diagrams satisfy the system contract specification. To this end, the approach is divided into two steps. First, we modularly verify that Simulink block diagrams satisfy the composition contract (i.e., the composition of contracts corresponding to the blocks that construct the Simulink block diagrams). We associate a contract for every elementary block as a specification. After that, we define the satisfaction relation that relates block to contract and verify that the satisfaction relation is preserved by composition. That is, if blocks satisfy their contract, respectively, then the composition of blocks satisfies the composition of contracts. Second, we define the refinement relation between contracts and verify that the composition of contracts refines the system contract specifications. If the block diagram satisfies the composition contracts and the composition contracts refine the system specification, then the block diagram also satisfies the system specification. Hence, the Simulink block diagram is correct with respect to the system specification.

We further elaborate on the concepts and properties mentioned above in the following. We first define the contract specification for the block.

Definition 10 (Contract). A contract is a tuple $C = (\boldsymbol{x}, \boldsymbol{y}, \phi_a, \phi_g)$, where

(i) $\mathbf{x}, \mathbf{y}$ are the input vectors and output vectors, respectively

(ii) $\phi_a \subseteq X^T$ represents the *assumption* for the input trajectories of the block, where $X \subseteq \mathscr{R}^n$

(iii) $\phi_g \subseteq (X \times Y)^T$ represents the *guarantee* for the input-output trajectories of the block, where $Y \subseteq \mathscr{R}^m$

The contract specifies the expected trajectory for each elementary block. An example of a contract for a block Gain is shown below.

Example 8 (Contract for Gain). As an example, we consider the Gain block that represented in Example 4. A contract specification for Gain is a tuple: $C = (\boldsymbol{x}, \boldsymbol{y}, \phi_a, \phi_g)$, where

$$\phi_a := \left\{\mathbf{x} \in \mathscr{R}^T \,|\, \forall t \in T, x(t) \in \mathscr{R}\right\},$$
$$\phi_g := \left\{(\mathbf{x}, \mathbf{y}) \in (\mathscr{R} \times \mathscr{R})^T \,|\, \forall t \in T, y(t) = k \cdot x(t)\right\}.$$

Next, we will define the satisfaction relation, which relates a block to a contract by determining when a given block's trajectories satisfy the specified specification. To define satisfaction relation, we first define a projection that expresses the input trajectory to aid that definition. A projection of a set $Tr \subseteq (X \times Y)^T$ into $\mathscr{X} \subseteq X^T$ is defined as the set $Tr\downarrow_{\mathscr{X}} = \{\mathbf{x} \in \mathscr{X} \,|\, \exists \mathbf{y}(\mathbf{x}, \mathbf{y}) \in Tr\}$, where T can be either the *discrete-time* domain or the *continuous-time* domain. We are now in a position to define satisfaction relation.

Definition 11 (Satisfaction). Let $B = (\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{y}, \varphi, f)$ be a block and $C = (\boldsymbol{x}, \boldsymbol{y}, \phi_a, \phi_g)$ be a contract. We say B satisfies C, denoted by $B \vDash C$, if $Tr\downarrow_{\mathscr{X}} \subseteq \phi_a$ and $Tr \subseteq \phi_g$.

We say that B is a correct implementation of C if $B \vDash C$. We have defined the notion of contract and satisfaction relation. The primary task of the proof method is to state that the satisfaction relation is preserved by composition. In the following, we study the composition operators of contracts according to the composition operators of blocks, i.e., serial, parallel, and algebraic loop-free feedback with multistep delays. We first define the serial composition of contracts.

Definition 12 (The serial composition of contracts). Let $C_i = (\mathbf{x}_i, \mathbf{y}_i, \phi_a^i, \phi_g^i)$ be contracts for $i = 1, 2$. The serial composition of C_1 and C_2, written $C_1 \,;\, C_2$, is a contract $(\boldsymbol{x}, \boldsymbol{y}, \phi_a, \phi_g)$, where

$$\mathbf{x} = \mathbf{x}_1,$$
$$\mathbf{y} = \mathbf{y}_2,$$
$$\phi_a := \left\{\mathbf{x}_1 \in X^T \middle| \mathbf{x}_1 \in \phi_a^1 \wedge \left((\exists y_1)\left((\mathbf{x}_1, \mathbf{y}_1) \in \phi_g^1 \wedge \phi_\rho \longrightarrow \mathbf{y}_1 \in \phi_a^2\right)\right)\right\},$$
$$\phi_g := \left\{(\mathbf{x}_1, \mathbf{y}_2) \in \left(X \times Y\right)^T \middle| (\exists y_1 \exists x_2)\left((\mathbf{x}_1, \mathbf{y}_1) \in \phi_g^1 \wedge \phi_\rho \wedge (\mathbf{x}_2, \mathbf{y}_2) \in \phi_g^2\right)\right\},$$
$$\phi_\rho := \bigwedge_{\left(y_{1,j}, x_{2,i}\right) \in \rho} y_{1,j}(t) = x_{2,i}(t), \forall t \in T.$$

The serial composition preserves the satisfaction relation. That is, if blocks satisfy their contracts, respectively, then the composition of blocks satisfies the composition of contracts.

Theorem 13 (Serial composition preserves satisfaction). *If$B_1 \models C_1$ and $B_2 \models C_2$, then $B_1\,; B_2 \models C_1\,; C_2$.*

Proof. We show a proof here for blocks with *continuous-time* trajectories, and the proof for blocks with *discrete-time* trajectories is similar. Let $B_1 = (\mathbf{x}_1, \mathbf{s}_1, \mathbf{y}_1, \varphi_1, f_1)$ and $B_2 = (\mathbf{x}_2, \mathbf{s}_2, \mathbf{y}_2, \varphi_2, f_2)$ be *continuous-time* blocks. The observable trajectories of B_1 and B_2 are denoted as $Tr_1 = \{(\mathbf{x}_1, \mathbf{y}_1)\colon T \mapsto X_1 \times Y_1 | \exists \mathbf{s}_1 \mathbf{y}_1 = f(\mathbf{x}_1, \mathbf{s}_1) \wedge \dot{\mathbf{s}}_1 = \varphi_{d1}(\mathbf{x}_1, \mathbf{s}_1)\}$ and $Tr_2 = \{(\mathbf{x}_2, \mathbf{y}_2)\colon T \mapsto X_2 \times Y_2 | \exists \mathbf{s}_2 \mathbf{y}_2 = f(\mathbf{s}_2, \mathbf{x}_2) \wedge \dot{\mathbf{s}}_2 = \varphi_{d2}(\mathbf{x}_2, \mathbf{s}_2)\}$, respectively, where $X_1, X_2 \subseteq \mathscr{R}^n, Y_1, Y_2 \subseteq \mathscr{R}^m, n, m \in \mathscr{N}^+$. The projection of Tr_1 into $\mathscr{X}_1$ is $Tr_1{\downarrow_{\mathscr{X}_1}} = \{\mathbf{x}_1 | \exists y_1 (\mathbf{x}_1, \mathbf{y}_1) \in Tr_1\}$. The projection of Tr_2 into $\mathscr{X}_2$ is $Tr_2{\downarrow_{\mathscr{X}_2}} = \{\mathbf{x}_2 | \exists y_2 (\mathbf{x}_2, \mathbf{y}_2) \in Tr_2\}$.

According to Definition 5, we write $B_1\,; B_2 = (\mathbf{x}_1, \mathbf{y}_2, \mathbf{s}, f, \varphi)$. We denote the observable trajectories of $B_1\,; B_2$ by Tr. Then$Tr = \{(\mathbf{x}_1, \mathbf{y}_2) | \exists \mathbf{s} \mathbf{y}_2 = f_2(\rho(f_1(\mathbf{x}_1, \mathbf{s}_1), \mathbf{s}_2)) \wedge \dot{\mathbf{s}} = \varphi_d(\mathbf{x}, \mathbf{s})\}$. The projection of Tr into $\mathscr{X}_1$ is$Tr{\downarrow_{\mathscr{X}_1}} = \{\mathbf{x}_1 | \exists y_1 \exists x_2 \exists y_2 (\mathbf{x}_1, \mathbf{y}_1) \in Tr_1 \wedge (\mathbf{y}_1, \mathbf{x}_2) \in \rho \wedge (\mathbf{x}_2, \mathbf{y}_2) \in Tr_2\}$. So, $Tr{\downarrow_{\mathscr{X}_1}} \subseteq Tr_1{\downarrow_{\mathscr{X}_1}}$. □

Given two contracts $C_1 = (\mathbf{x}_1, \mathbf{y}_1, \phi_a^1, \phi_g^1)$ and $C_2 = (\mathbf{x}_2, \mathbf{y}_2, \phi_a^2, \phi_g^2)$, by Definition 12, we have $C_1\,; C_2 = (\mathbf{x}_1, \mathbf{y}_2, \phi_a, \phi_g)$, where $\phi_a = \{\mathbf{x}_1 \in X^T | \mathbf{x}_1 \in \phi_a^1 \wedge ((\exists y_1)((\mathbf{x}_1, \mathbf{y}_1) \in \phi_g^1 \wedge \phi_\rho \longrightarrow \mathbf{y}_1 \in \phi_a^2))\}$ and $\phi_g = \{(\mathbf{x}_1, \mathbf{y}_2) \in (X \times Y)^T | (\exists y_1 \exists x_2)((\mathbf{x}_1, \mathbf{y}_1) \in \phi_g^1 \wedge \phi_\rho \wedge (\mathbf{x}_2, \mathbf{y}_2) \in \phi_g^2)\}$.

In order to proof $B_1\,; B_2 \models C_1\,; C_2$, according to Definition 11, we only need to show that: if $Tr{\downarrow_{\mathscr{X}_1}} \subseteq \phi_a$ implies $Tr \subseteq \phi_g$. We first prove that $Tr{\downarrow_{\mathscr{X}_1}} \subseteq \phi_a$.

Since $B_1 \models C_1$, according to Definition 11, we have $Tr_1{\downarrow_{\mathscr{X}_1}} \subseteq \phi_a^1$ implies $Tr_1 \subseteq \phi_g^1$. That is, for all $x_1 \in Tr{\downarrow_{\mathscr{X}_1}} \subseteq Tr_1{\downarrow_{\mathscr{X}_1}} \subseteq \phi_a^1$, then $(\mathbf{x}_1, \mathbf{y}_1) \in \phi_g^1$. Since $B_2 \models C_2$, according to Definition 11, we have for all $x_2 \in Tr_2{\downarrow_{\mathscr{X}_2}} \subseteq \phi_a^2$, then $Tr_2 \subseteq \phi_g^2$. There exists a connection $\rho = (\mathbf{y}_1, \mathbf{x}_2)$, according to Definition 4, we have $y_1(t) = x_2(t)$. Then, $\mathbf{y}_1 \in \phi_a^2$, and by the expression of ϕ_a, we have $\mathbf{x}_1 \in \phi_a$. Thus, $Tr{\downarrow_{\mathscr{X}_1}} \subseteq \phi_a$. For all $(\mathbf{x}_1, \mathbf{y}_2) \in Tr$, we have $(\mathbf{x}_1, \mathbf{y}_1) \in \phi_g^1$, $\mathbf{y}_1 = \mathbf{x}_2$, and $(\mathbf{x}_2, \mathbf{y}_2) \in \phi_g^2$. According to the expression of ϕ_g, we have $(\mathbf{x}_1, \mathbf{y}_2) \in \phi_g$. Thus, $Tr \subseteq \phi_g$. Hence, $B_1\,; B_2 \models C_1\,; C_2$.

We now define the parallel composition of contracts.

Definition 14 (The parallel composition of contracts). Given contracts $C_1 = (\boldsymbol{x}_1, \boldsymbol{y}_1, \phi_a^1, \phi_g^1)$ and $C_2 = (\boldsymbol{x}_2, \boldsymbol{y}_2, \phi_a^2, \phi_g^2)$, we define the parallel composition of contracts as $C_1 \| C_2 = (\boldsymbol{x}, \boldsymbol{y}, \phi_a, \phi_g)$, where $\boldsymbol{x} = (x_1, x_2), \boldsymbol{y} = (y_1, y_2)$, and

$$\phi_a \coloneqq \{\mathbf{x} \in X^T | x_1 \in \phi_a^1 \wedge x_2 \in \phi_a^2\},$$

$$\phi_g \coloneqq \left\{(\mathbf{x}, \mathbf{y}) \in (X \times Y)^T | (x_1, y_1) \in \phi_g^1 \wedge (x_2, y_2) \in \phi_g^2\right\}.$$

The parallel composition also preserves the satisfaction relation. That is, if blocks satisfy their contracts, respectively, then the parallel composition of blocks satisfies the parallel composition of contracts.

Theorem 15 (Parallel composition preserves satisfaction). *If$B_1 \models C_1$ and $B_2 \models C_2$, then $B_1 \| B_2 \models C_1 \| C_2$.*

Proof. We show here a proof for blocks with *continuous-time* trajectories, and the proof for blocks with *discrete-time* trajectories is similar. Let $B_1 = (\mathbf{x}_1, \mathbf{s}_1, y_1, \varphi_1, f_1)$ and $B_2 = (\mathbf{x}_2, \mathbf{s}_2, y_2, \varphi_2, f_2)$ be blocks with *continuous-time* trajectories. The observable trajectories of B_1 and B_2 are denoted as $Tr_1 = \{(\mathbf{x}_1, \mathbf{y}_1)\colon T \mapsto X_1 \times Y_1 | \exists \mathbf{s}_1 \mathbf{y}_1 = f_1(\mathbf{x}_1, \mathbf{s}_1) \wedge \dot{\mathbf{s}}_1 = \varphi_{d1}(\mathbf{x}_1, \mathbf{s}_1)\}$ and $Tr_2 = \{(\mathbf{x}_2, \mathbf{y}_2)\colon T \mapsto X_2 \times Y_2 | \exists \mathbf{s}_2 \mathbf{y}_2 = f_2(\mathbf{s}_2, \mathbf{x}_2) \wedge \dot{\mathbf{s}}_2 = \varphi_{d2}(\mathbf{x}_2, \mathbf{s}_2)\}$, respectively, $X_1, X_2 \subseteq \mathscr{R}^n, Y_1, Y_2 \subseteq \mathscr{R}^m, n, m \in \mathscr{N}^+$. The projection of Tr_1 into $\mathscr{X}_1$ is $Tr_1{\downarrow_{\mathscr{X}_1}} = \{\mathbf{x}_1 | \exists \mathbf{y}_1 (\mathbf{x}_1, \mathbf{y}_1) \in Tr_1\}$. The projection of Tr_2 into $\mathscr{X}_2$ is the set $Tr_2{\downarrow_{\mathscr{X}_2}} = \{\mathbf{x}_2 | \exists \mathbf{y}_2 (\mathbf{x}_2, \mathbf{y}_2) \in Tr_2\}$. Following Definition 7, we write $B_1 \| B_2 = (\mathbf{x}, \mathbf{y}, \mathbf{s}, f, \varphi)$. We denote the observable trajectories of $B_1 \| B_2$ by Tr. Then $Tr = \{(\mathbf{x}, \mathbf{y}) | \exists \mathbf{s} = (s_1, s_2) \mathbf{y}_1 = f_1(\mathbf{x}_1, \mathbf{s}_1) \wedge \dot{\mathbf{s}}_1 = \varphi_{d1}(\mathbf{x}_1, \mathbf{s}_1) \wedge \mathbf{y}_2 = f_2(\mathbf{x}_2, \mathbf{s}_2) \wedge \dot{\mathbf{s}}_2 = \varphi_{d2}(\mathbf{x}_2, \mathbf{s}_2)\}$. The projection of Tr into $\mathscr{X}$ is $Tr{\downarrow_{\mathscr{X}}} = \{\mathbf{x} | \exists \mathbf{y}_1 \mathbf{y}_2 (\mathbf{x}_1, \mathbf{y}_1) \in Tr_1 \wedge (\mathbf{x}_2, \mathbf{y}_2) \in Tr_2\}$, where $\mathscr{X} \subseteq (X_1 \times X_2)^T$.

Let $C_1 = (\mathbf{x}_1, \mathbf{y}_1, \phi_a^1, \phi_g^1)$ and $C_2 = (\mathbf{x}_2, \mathbf{y}_2, \phi_a^2, \phi_g^2)$ be contracts. By Definition 14, we write $C_1 \| C_2 = (\mathbf{x}, \mathbf{y}, \phi_a, \phi_g)$, where $\mathbf{x} = (x_1, x_2), \mathbf{y} = (y_1, y_2)$, and

$$\begin{aligned} \phi_a &\coloneqq \{\mathbf{x} \in X^T | \mathbf{x}_1 \in \phi_a^1 \wedge \mathbf{x}_2 \in \phi_a^2\}, \\ \phi_g &\coloneqq \left\{(\mathbf{x}, \mathbf{y}) \in (X \times Y)^T | (\mathbf{x}_1, \mathbf{y}_1) \in \phi_g^1 \wedge (\mathbf{x}_2, \mathbf{y}_2) \in \phi_g^2\right\}. \end{aligned} \tag{1}$$

□

To proof $B_1 \| B_2 \models C_1 \| C_2$, according to Definition 11, we only need to show that if $(Tr{\downarrow_{\mathscr{X}}}) \subseteq \phi_a$ implies $Tr \subseteq \phi_g$. We first prove that $Tr{\downarrow_{\mathscr{X}}} \subseteq \phi_a$. Since $B_1 \models C_1$, according to Definition 11, we have if $(Tr_1{\downarrow_{\mathscr{X}_1}}) \subseteq \phi_a^1$ implies $Tr_1 \subseteq \phi_g^1$. That is for all $x_1 \in (Tr_1{\downarrow_{\mathscr{X}_1}}) \in \phi_a^1$, for all $(x_1, y_1) \in Tr_1$, $(x_1, y_1) \in \phi_g^1$. Since $B_2 \models C_2$, we have $(Tr_2{\downarrow_{\mathscr{X}_2}}) \subseteq \phi_a^2$ implies $Tr_2 \subseteq \phi_g^2$. That is, for all $x_2 \in (Tr_2{\downarrow_{\mathscr{X}_2}})$, then $x_2 \in \phi_a^2$, for all $(x_2, y_2) \in Tr_2$, then $(x_2, y_2) \in \phi_g^2$. Hence, by (1), we have $\mathbf{x} = (x_1, x_2) \in \phi_a$, $(\mathbf{x}, \mathbf{y}) \in \phi_g$. Hence, $B_1 \| B_2 \models C_1 \| C_2$.

The parallel composition of contract is also associative and commutative.

Lemma 16 (Associativity, commutativity). *Let C_1, C_2, and C_3 be contracts. Then*

(i) $C_1 \| C_2 = C_2 \| C_1$

(ii) $(C_1 \| C_2) \| C_3 = C_1 \| (C_2 \| C_3)$

Proof. Immediately follows from the Definition 14. □

We now define the algebraic loop-free feedback composition of contracts.

Definition 17 (The algebraic loop-free feedback composition of contracts). Let $C_1 = (\mathbf{x}_1, \mathbf{y}_1, \phi_a^1, \phi_g^1)$ and $C_2 = (\mathbf{x}_2, \mathbf{y}_2, \phi_a^2,$

$\phi_g^2)$ be the contracts. The algebraic loop-free feedback composition of C_1 and C_2 be defined as $C_1 \otimes_f C_2 = (\mathbf{x}, \mathbf{y}, \phi_a, \phi_g)$,

$$\phi_a := \left\{\mathbf{x}_{1,1} \in X^T \middle| \mathbf{x}_{1,1} \in \phi_a^1 \wedge \left((\exists y_1)(\mathbf{x}_1, \mathbf{y}_1) \in \phi_g^1 \wedge \phi_{\rho_2} \longrightarrow y_1 \in \phi_a^2\right)\right) \wedge \left((\exists \mathbf{y}_2)(\mathbf{x}_2, \mathbf{y}_2) \in \phi_g^2 \wedge \phi_{\rho_1} \longrightarrow \mathbf{y}_2 \in \phi_a^1\right)\right)\right\},$$

$$\phi_g := \left\{(\mathbf{x}_{1,1}, \mathbf{y}_1) \in (X \times Y)^T \middle| (\exists x_{1,2} \exists y_2 \exists x_2)\left((\mathbf{x}_1, \mathbf{y}_1) \in \phi_g^1 \wedge \phi_\rho \wedge (\mathbf{x}_2, \mathbf{y}_2) \in \phi_g^2\right)\right\},$$

$$\phi_\rho := \phi_{\rho_1} \wedge \phi_{\rho_2}.$$

where $\phi_{\rho_2} := \wedge_{(y_{1,j}, x_{2,i}) \in \rho_2} y_{1,j}(t) = x_{2,i}(t)$, $\phi_{\rho_1} := \wedge_{(y_{2,j}, x_{1,i}) \in \rho_1} y_{2,j}(t) = x_{1,i}(t)$, and $\mathbf{x}_1 = (x_{1,1}, x_{1,2})$.

The algebraic loop-free feedback composition also preserves the satisfaction relation.

Theorem 18 (Algebraic loop-free feedback composition preserves satisfaction). *Let B_1 be the feedback with an algebraic loop and B_2 be a block with multistep delays. Let $C_1 = (\mathbf{x}_1, \mathbf{y}_1, \phi_a^1, \phi_g^1)$ and $C_2 = (\mathbf{x}_2, \mathbf{y}_2, \phi_a^2, \phi_g^2)$ be the contracts. If $B_1 \models C_1$ and $B_2 \models C_2$, then $B_1 \otimes_f B_2 \models C_1 \otimes_f C_2$.*

Proof. It easily follows from Theorem 13 to Theorem 15. □

We next turn to the refinement relation between the contract. We follow a standard notion inspired by [20].

Definition 19 (Refinement of contracts). Let $C_1 = (\mathbf{x}, \mathbf{y}, \phi_a^1, \phi_g^1)$ and $C_2 = (\mathbf{x}, \mathbf{y}, \phi_a^2, \phi_g^2)$ be two contracts. We say C_2 refines C_1, denoted by $C_2 \preccurlyeq C_1$, *if* $\phi_a^1 \subseteq \phi_a^2$ and $\phi_g^2 \subseteq \phi_g^1$.

Refinement relaxes assumptions and reinforces guarantees, therefore, strengthening the contract. Obviously, the following refinement rule holds.

Lemma 20. *Let $C_1 = (\mathbf{x}, \mathbf{y}, \phi_a^1, \phi_g^1)$ and $C_2 = (\mathbf{x}, \mathbf{y}, \phi_a^2, \phi_g^2)$ be two contracts. Then*

(i) $C_i \preccurlyeq C_i$, for $i = 1, 2$

(ii) If $\phi_a^1 \subseteq \phi_a^2$ and $\phi_g^2 = \phi_g^1$, then $C_2 \preccurlyeq C_1$

(iii) If $\phi_g^2 \subseteq \phi_g^1$ and $\phi_a^1 = \phi_a^2$, then $C_2 \preccurlyeq C_1$

The refinement implies that every Simulink block diagram satisfies the C_2 also satisfies the C_1. This gives us the following property of correctness.

where $\mathbf{x} = x_{1,1}, \mathbf{y} = y_1$, and

Theorem 21 (Correctness). *Let $B = (\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{y}, \varphi, f)$ be a block. Let $C_1 = (\boldsymbol{x}, \boldsymbol{y}, \phi_a^1, \phi_g^1)$ and $C_2 = (\boldsymbol{x}, \boldsymbol{y}, \phi_a^2, \phi_g^2)$ be contracts. The B is said to be correct for C_2, if $(B \models C_1 \wedge C_1 \preccurlyeq C_2) \Rightarrow B \models C_2$.*

Proof. Suppose $C_1 \preccurlyeq C_2$, in terms of Definition 19, then $\phi_a^2 \subseteq \phi_a^1$, and $\phi_g^1 \subseteq \phi_g^2$. Suppose $B \models C_1$, then, for all $x \in Tr\downarrow_{\mathcal{X}} \subseteq \phi_a^1$, and $Tr \subseteq \phi_g^1 \subseteq \phi_g^2$. Hence, if $x \in Tr\downarrow_{\mathcal{X}} \subseteq \phi_a^2$, then $B \models C_2$.

Theorem 21 is essential in our contract theory since it relates the Simulink block diagram (composition block) to a system contract specification. That is, if a Simulink block diagram satisfies the composition contract, and the composition contract refines the system contract, then it also satisfies the system contract. This ensures that the Simulink block diagram is correct.

6. Case Study

We have discussed the proof approach in the previous section, and in this section, the approach that we proposed is being implemented through a real-world case study.

6.1. Problem Statement. We examine a case, a safety-critical water level control system of reservoir, to illustrate how to verify that the controller model of a reservoir originating from the farmland irrigation satisfy the safety requirements.

This reservoir mainly is used for irrigation and considers the comprehensive utilization of flood control and aquaculture. An inlet water pipe and a spillway exist on the top and bottom of the dam, respectively. The maximum dam height is 40m, the maximum is 30m, and the minimum water level is 10m. The management facilities of the reservoir are very backward. There are no special management agencies to observe and safety check the water level and no timely flood control.

We will use Simulink to model the water level control system of the reservoir and simulate the water level trajectories to monitor the water level control operation. Then we adopt the method that we presented to verify the Simulink block diagram's correctness for ensuring the water level's safety according to the reservoir's actual situation. The water level controls the opening or closing of the valves. That is, the water level is neither higher than the highest water level,

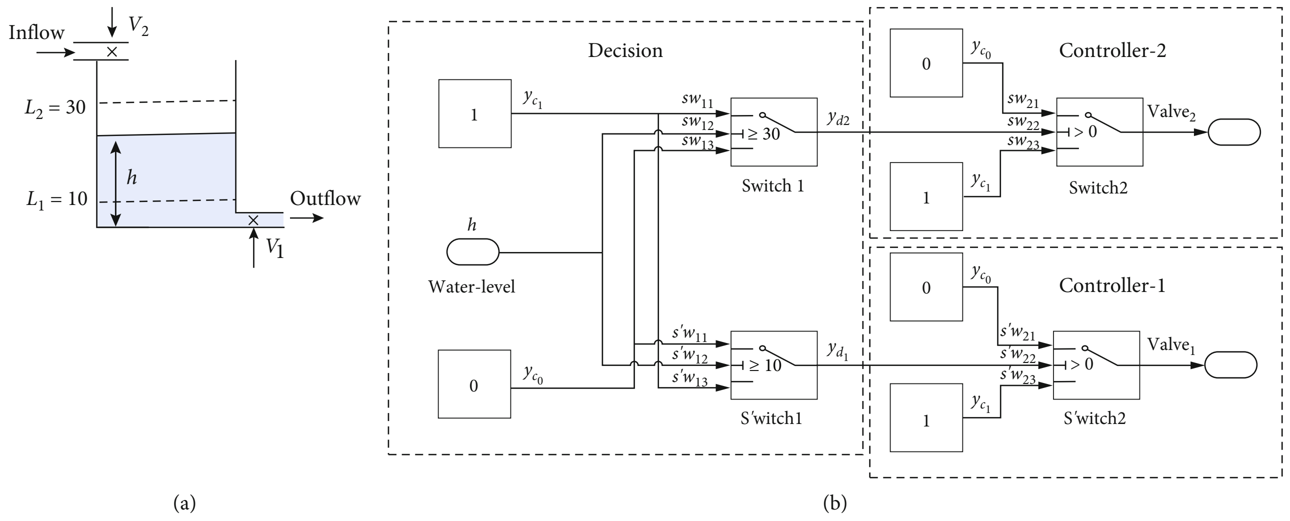


FIGURE 7: (a) An overview of reservoir system. (b) The model of controller.

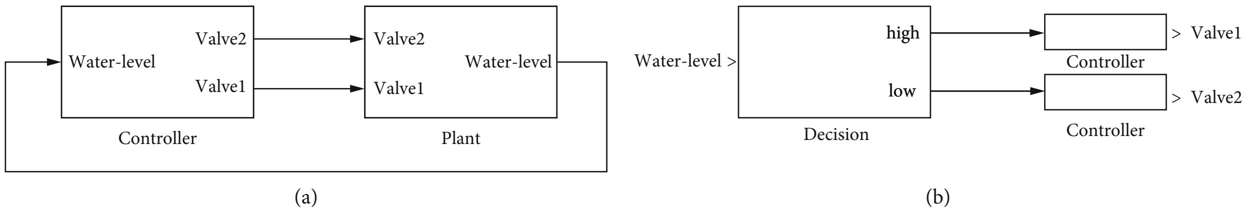


FIGURE 8: (a) A Simulink block diagram of reservoir system. (b) The block diagram of controller.

which causes the dam to be overloaded, nor can the aquaculture industry and farmland irrigation be affected by the water level being lower than the lowest water level.

6.2. Architectural Overview of Control System. The control system of water level consists of the following two major components: Sensor and Plant.

(i) *Sensor.* The *Sensor* is a water level sensor. Inputs about the water level of the reservoir from its corresponding level sensor, and we suppose that the *Sensor* works properly

(ii) *Plant.* The *Plant* is made up of four principal components, as shown in Figure 7(a): Reservoir, Valves, Controller, and Water inlet and outlet pipe

   (a) *Reservoir.* The *Reservoir* has parameters giving the maximum water lever L_2, the minimum water lever L_1, and the water level of the reservoir $h \geq 0$

   (b) *Water valves.* The *Reservoir* has two water valves. The inlet valve V_2 is at the top of the *Reservoir*, and the outlet valve V_1 is at the bottom. We suppose that the valves are either fully opened or fully closed immediately

   (c) *Controller.* The *Controller* is able to read the current water level provided by the sensor, and the output is the command signal *open* or *closed*. The controller's target is to keep the water level between the minimum water lever L_1 and the maximum water lever L_2

   (d) *Water inlet and outlet.* The inlet pipe $F_{\text{in}} \geq 0$ and $F_{\text{out}} \geq 0$, and $F_{\text{in}} \neq F_{\text{out}}$.

As the system is a closed-loop system, the controller must work with the reservoir. In Figure 8(a), a system including both controller and plant is given. We only focus on the controller.

6.3. The Safety Requirements of Water Level. The safety requirements for water level h in the controller are given below, and the purpose is to avoid the water empty and filling of reservoir.

Requirement 1 When the h of reservoir is above L_1, the V_1 is opened, and when the h of reservoir is below L_1, the V_1 is closed.

Requirement 2 When the h of reservoir is above L_2, the V_2 is closed, and when the h of reservoir is below L_2, the V_2 is opened.

Driven by requirement, we now refine system-level requirements into block-level implementations. When the system is refined, subsystem details are added. We first decompose the controller into three different controllers: *Decision* controller, *Controller*$_1$ controller, and *Controller*$_2$

controller. The block diagrams of these controllers can be depicted graphically as in Figure 8(b). In the block diagram, the *Decision* decides that the water level is high or low. The $Controller_1$ computes whether the $Valve_1$ should be open, while the $Controller_2$ computes if the $Valve_2$ should be open.

For the *Decision*, if the water level is above L_2, it is high. If the water level is below L_1, it is low. For the $Valve_2$, if the water level is high, then $Valve_2$ should be close(0). Otherwise, the $Valve_2$ should be open(1). For the $Valve_1$, if the water level is low, then $Valve_1$ should be close(0). Otherwise, then $Valve_1$ should be open(1).

We use the contract as a formal specification language. We suppose $L_1 = 10\text{m}$, $L_2 = 30\text{m}$, and $0\text{m} \le h(t) \le 40\text{m}$. The system-level safety requirement specification 1 is denoted as $C_{\text{req1}} = (\mathbf{h}, \mathbf{y}_{\mathbf{v}_1}, \phi_a^{\text{req1}}, \phi_g^{\text{req1}})$.

$$\phi_a^{\text{req1}} := \{\mathbf{h} \in \mathscr{R}^T | \forall t \in T, 0 \le h(t) \le 40\},$$
$$\phi_g^{\text{req1}} := \left\{ \left(\mathbf{h}, \mathbf{y}_{\mathbf{v}_1}\right) \in (\mathscr{R} \times \mathscr{R})^T | \forall t \in T, \boldsymbol{y}_{\boldsymbol{v}_1} = \begin{cases} \text{open}(1), & h(t) \ge 10, \\ \text{close}(0), & 0 \le h(t) < 10. \end{cases} \right\}. \tag{2}$$

The system-level safety requirement specification 2 is denoted as $C_{\text{req2}} = (\mathbf{h}, \mathbf{y}_{\mathbf{v}_2}, \phi_a^{\text{req2}}, \phi_g^{\text{req2}})$, where

$$\phi_a^{\text{req2}} := \{\mathbf{h} \in \mathscr{R}^T | \forall t \in T, 0 \le h(t) \le 40\},$$
$$\phi_g^{\text{req2}} := \left\{ \left(\mathbf{h}, \mathbf{y}_{\mathbf{v}_2}\right) \in (\mathscr{R} \times \mathscr{R})^T | \forall t \in T, \mathbf{y}_{\mathbf{v}_2} = \begin{cases} \text{close}(0), & h(t) \ge 30, \\ \text{open}(1), & 0 \le h(t) < 30. \end{cases} \right\}. \tag{3}$$

6.4. Verification of Correctness for the Controller System. We construct a Simulink block diagram for the control system according to requirements shown in Figure 7(b). Following the previous technical route, the verification method applied in this case consists of two stages.

Stage 1: verifying the Simulink block diagram satisfies the corresponding composition of contract.

The *Decision* controller has two controllers: $Decision_1$ and $Decision_2$. The $Decision_2$ determines whether the water level is higher than the highest water level. The composition of Simulink block diagrams of $Decision_2$ is denoted as $\otimes B_{d_2} = (B_{c_1} \| B_{c_0}) \,;\, \rho_1 \,;\, B_{sw_1}$, where $\rho_1 = \{(y_{c_1}, sw_{11}), (y_{c_0}, sw_{13})\}$. We denote the h by sw_{12}, and then $\otimes B_{d_2} := (sw_{12}, \mathbf{0}, y_{d_2}, -, f_{d_2}(sw_{12}(t), \mathbf{0}))$, where,

$$f_{d_2}(sw_{12}, 0) = \begin{cases} 1(\text{high}) & sw_{12}(t) \ge 30, \\ 0(\neg\text{high}) & sw_{12}(t) < 30. \end{cases} \tag{4}$$

Therefore, the observable trajectories of $\otimes B_{d_2}$ are denoted as $Tr_2 = \{(sw_{12}, y_{d_2}) \colon T \mapsto \mathscr{R} \times \mathscr{R} | \forall t \in T, y_{d_2}(t) = f_{d_2}(sw_{12}(t), \mathbf{0})\}$, where

$$y_{d_2}(t) = \begin{cases} 1(\text{high}) & sw_{12}(t) \ge 30, \\ 0(\neg\text{high}) & sw_{12}(t) < 30. \end{cases} \tag{5}$$

The main task in the next step is to apply the composition rules to calculate the composition contract and prove that the composition of blocks satisfies the composition of the corresponding contracts. We first specify a contract for each elementary block following Table 1. Then we apply the composition rules to calculate the composition contract and prove that the composition of blocks satisfies the composition contracts.

The composition contract of the $Decision_2$ is stated as $\otimes C_{d_2} = (sw_{12}, y_{d_2}, \phi_a^{d_2}, \phi_g^{d_2})$, where

$$\phi_a^{d_2} := \left\{ sw_{12} \in \mathscr{R}^T | \forall t \in T, \left(\left(\exists \mathbf{y}_{B_1} \right) \colon \left(\mathbf{0}, \mathbf{y}_{B_1} \right) \in \phi_g^{B_1} \wedge \phi_{\rho_1} \longrightarrow \mathbf{y}_{B_1} \in \phi_a^{B_{sw_1}} \right) \right\},$$
$$\phi_g^{d_2} := \left\{ \left(\mathbf{0}, y_{d_2} \right) \in (\mathscr{R} \times \mathscr{R})^T \middle| \forall t \in T, \left(\mathbf{0}, \mathbf{y}_{B_1} \right) \in \phi_g^{B_1} \wedge \phi_{\rho_1} \wedge \left(sw_{12}, y_{d_2} \right) \in \phi_g^{sw_1} \right\}, \tag{6}$$
$$\phi_{\rho_1} := \left(y_{c_1}(t) = sw_{11}(t) \right) \wedge \left(y_{c_0}(t) = sw_{13}(t) \right).$$

Since

$$\phi_g^{B_1} := \left\{ \left(\mathbf{0}, \mathbf{y}_{\mathbf{B}_1} \right) \in \left(\mathscr{R}^2 \times \mathscr{R}^2 \right)^T \middle| \forall t \in T, y_{c_1}(t) = 1 \wedge y_{c_0}(t) = 0 \right\},$$
$$\phi_g^{B_{sw_1}} := \left\{ \left(sw_{12}, y_{d_2} \right) \in (\mathscr{R} \times \mathscr{R})^T \middle| \forall t \in T, y_{d_2}(t) = f_{sw_1}(sw_{12}(t), \mathbf{0}) \right\}, \tag{7}$$

where

$$y_{d_2}(t) = \begin{cases} sw_{11} & sw_{12} \ge 30, \\ sw_{13} & sw_{12} < 30. \end{cases} \tag{8}$$

We use quantifier elimination law to simplify the expression. Therefore, the composition contract $(\otimes C_{d_2})$ of the *D* *ecision*$_2$ controller was written in the following form.

$$\phi_a^{d_2} := \{sw_{12} \in \mathscr{R}^T | \forall t \in T, sw_{12}(t) \in \mathscr{R}\},$$
$$\phi_g^{d_2} := \left\{ \left(sw_{12}, y_{d_2}\right) \in (\mathscr{R} \times \mathscr{R})^T \middle| \forall t \in T, y_{d_2}(t) = \begin{cases} \text{high}(1) & sw_{12}(t) \geq 30, \\ \neg\text{high}(0) & sw_{12}(t) < 30. \end{cases} \right\}. \tag{9}$$

Similarly, the *Decision*$_1$ determines whether the water level is below the lowest water level. The composition of Simulink block diagrams of *Decision*$_1$ is denoted as $\otimes B_{d_1} = (B_{c_1} \| B_{c_0})\,;\rho_2\,;B_{s'w_1}$, where $\rho_2 = \{(y_{c_0}, s'w_{11}), (y_{c_1}, s'w_{13})\}$. Let $B_1 = B_{c_1} \| B_{c_0}$ and $\mathbf{y}_{B1} = (y_{c_0}, y_{c_1})$. We denote the Simulink block diagrams of *Decision*$_1$ by $\otimes B_{d_1} := (s'w_{12}, \mathbf{0}, y_{d_1}, –, f_{d_1}(s'w_{12}(t), \mathbf{0}))$, where

$$f_{d_1}\left(s'w_{12}(t), \mathbf{0}\right) = \begin{cases} 0(\neg\text{low}) & s'w_{12}(t) \geq 10, \\ 1(\text{low}) & s'w_{12}(t) < 10. \end{cases} \tag{10}$$

The observable trajectories of $\otimes B_{d_1}$ is denoted as $Tr_1 = \{(s'w_{12}, y_{d_1}) : T \mapsto \mathscr{R} \times \mathscr{R} | \forall t \in T, y_{d_1}(t) = f_{d_1} s'w_{12}(t), \mathbf{0})\}$, where

$$y_{d_1}(t) = \begin{cases} 0(\neg\text{low}) & s'w_{12}(t) \geq 10, \\ 1(\text{low}) & s'w_{12}(t) < 10. \end{cases} \tag{11}$$

The composition contract of the *Decision*$_1$ controller for $\otimes B_{d_1}$ is stated as $\otimes C_{d_1} = (s'w_{12}, y_{d_1}, \phi_a^{d_1}, \phi_g^{d_1})$, where

$$\phi_a^{d_1} := \left\{s'w_{12} \in \mathscr{R}^T \middle| \forall t \in T, s'w_{12}(t) \in \mathscr{R}\right\},$$
$$\phi_g^{d_1} := \left\{ \left(s'w_{12}, y_{d_1}\right) \in (\mathscr{R} \times \mathscr{R})^T \middle| \forall t \in T, [l]@l@y_{d_1}(t) \right. \\ = \left. \begin{pmatrix} \neg\text{low}(0) & s'w_{12}(t) \geq 10, \\ \text{low}(1) & s'w_{12}(t) < 10. \end{pmatrix} \right\}. \tag{12}$$

Since variables sw_{12} and $s'w_{12}$ are used to read the values of water level, so $sw_{12}(t) = h(t)$ and $s'w_{12}(t) = h(t)$. According to Definition 11, $\forall t \in T$, we have $\{sw_{12} \in \mathscr{R}^T | sw_{12}(t) \geq 0\} \subseteq \phi_a^{d_2}$, $Tr_2 \subseteq \phi_g^{d_2}$, and $\{s'w_{12} \in \mathscr{R}^T | s'w_{12}(t) \geq 0\} \subseteq \phi_a^{d_1}$, $Tr_1 \subseteq \phi_g^{d_1}$. Hence, $\otimes B_{d_2} \vDash \otimes C_{d_2}$ and $\otimes B_{d_1} \vDash \otimes C_{d_1}$.

We next turn to research the controller *Controller*$_1$. It can be represented as a composition block $\otimes B_{v_1} = ((B_{c_1} \| B_{c_0})\,;\rho_3\,;B_{s'w_2})$, where $\rho_3 = \{(y_{c_1}, s'w_{23}), (y_{c_0}, s'w_{21})\}$. Let $\otimes B_{v_1} = (s'w_{22}, \mathbf{0}, y_{v_1}, –, y_{v_1}(t) = f_{v_1}(s'w_{22}(t), \mathbf{0}))$, where

$$f_{v_1}\left(s'w_{22}(t), \mathbf{0}\right) = \begin{cases} 0(\text{close}) & s'w_{22}(t) > 0, \\ 1(\text{open}) & s'w_{22}(t) \leq 0. \end{cases} \tag{13}$$

Therefore, the observable trajectories of $\otimes B_{v_1}$ is denoted as $Tr_3 = \{(s'w_{22}, y_{v_1}) : T \mapsto \mathscr{R} \times \mathscr{R} | \forall t \in T, y_{v_1}(t) = f_{v_1}(s'w_{22}(t), \mathbf{0})\}$, where

$$y_{v_1}(t) = \begin{cases} 0(\text{close}) & s'w_{22}(t) > 0, \\ 1(\text{open}) & s'w_{22}(t) \leq 0. \end{cases} \tag{14}$$

The composition contract of *Controller*$_1$ is stated as $\otimes C_{v_1} = (s'w_{22}, y_{v_1}, \phi_a^{v_1}, \phi_g^{v_1})$, where

$$\phi_a^{v_1} := \left\{s'w_{22} \in \mathscr{R}^T \middle| \forall t \in T, s'w_{22}(t) \in \mathscr{R}\right\},$$
$$\phi_g^{v_1} := \left\{ \left(s'w_{22}, y_{v_1}\right) \in (\mathscr{R} \times \mathscr{R})^T \middle| \forall t \in T', y_{v_1}\left(s'w_{22}(t), \mathbf{0}\right) \right. \\ = \left. \begin{cases} 0(\text{close}) & s'w_{22}(t) > 0, \\ 1(\text{open}) & s'w_{22}(t) \leq 0. \end{cases} \right\}. \tag{15}$$

According to Definition 11, $\forall t \in T$, we have $\{s'w_{22} : T \mapsto \mathscr{R} | s'w_{22}(t) \in \mathscr{R}\} \subseteq \phi_a^{v_1}$, and $Tr_3 \subseteq \phi_g^{v_1}$. Hence, $\otimes B_{v_1} \vDash \otimes C_{v_1}$.

For requirement 1 Next, we will compose the *Decision*$_1$ controller and *Controller*$_1$ controller for verifying requirement 1. We denote the composition of *Decision*$_1$ and *Cont roller*$_1$ by $\otimes B_{\text{Im }1}$. We denote the h by $s'w_{12}$. We write $\otimes B_{\text{Im }1} = \otimes B_{d_1}\,; \otimes B_{v_1} = (s'w_{12}, \mathbf{0}, y_{v_1}, –, y_{v_1} = f_{\text{Im }1}(s'w_{12}(t), \mathbf{0}))$, where

$$f_{\text{Im }1}\left(\mathbf{s'w_{12}}(t), \mathbf{0}\right) = \begin{cases} 1(\text{open}) & s'w_{12}(t) \geq 10, \\ 0(\text{close}) & s'w_{12}(t) < 10. \end{cases} \tag{16}$$

The observable trajectories of $\otimes B_{\text{Im }1}$ is denoted as $Tr_{\text{Im }1} = \{(s'w_{12}, y_{v_2}) : T \mapsto \mathscr{R} \times \mathscr{R} | \forall t \in T, y_{\text{Im }1}(t) = f_{\text{Im }1}(s'w_{12}(t), \mathbf{0}))\}$, where

$$y_{\text{Im }1}(t) = \begin{cases} 1(\text{open}) & s'w_{12}(t) \geq 10, \\ 0(\text{close}) & s'w_{12}(t) < 10. \end{cases} \tag{17}$$

The composition contract of $\otimes B_{\text{Im }1}$ is defined as $\otimes C_{\text{Im }1} = (s'w_{12}, y_{v_1}, \phi_a^{\text{Im }1}, \phi_g^{\text{Im }1})$, and

$$\phi_a^{\text{Im }1} := \left\{s'w_{12} \in \mathscr{R}^T \middle| \forall t \in T, s'w_{12}(t) \in \mathscr{R}\right\},$$
$$\phi_g^{\text{Im }1} := \left\{ \left(s'w_{12}, y_{v_1}\right) \in (\mathscr{R} \times \mathscr{R})^T | \forall t \in T, y_{v_1}(t) \right. \\ = \left. \begin{cases} 1(\text{open}) & s'w_{12}(t) \geq 10, \\ 0(\text{close}) & s'w_{12}(t) < 10. \end{cases} \right\}. \tag{18}$$

$\forall t \in T$, we have $\{s'w_{12}(t) : T \mapsto \mathscr{R} | s'w_{12}(t) \in \mathscr{R}\} \subseteq \phi_a^{\text{Im }1}$, and $Tr_{\text{Im }1} \subseteq \phi_g^{\text{Im }1}$. Hence, $B_{\text{Im }1} \vDash C_{\text{Im }1}$.

Stage 2: correctness verification. Our primary goal in this step is to verify that the Simulink block diagram $(\otimes B_{v_1})$ is correct for system contract C_{req1}. In terms of Definition 19, we get $\phi_a^{\mathrm{req1}} \subseteq \phi_a^{\mathrm{Im\ 1}}$ and $\phi_g^{\mathrm{Im\ 1}} \subseteq \phi_g^{req1}$, then, $C_{\mathrm{Im\ 1}} \preccurlyeq C_{\mathrm{req1}}$. According to Theorem 21, we have $B_{\mathrm{Im\ 1}} \vDash C_{\mathrm{Im\ 1}} \wedge C_{\mathrm{Im\ 1}} \preccurlyeq C_{\mathrm{req1}} \Rightarrow B_{\mathrm{Im\ 1}} \vDash C_{\mathrm{req1}}$.

For requirement 2

We next turn to verify the correctness of the Simulink block diagrams for requirement 2. Similar to the method above, the *Controller*$_2$ controller can be represented as a composition block $\otimes B_{v_2} = (B_{c_1} \| B_{c_0})\, ; \rho_4\, ; B_{sw_2}$, where $\rho_4 = \{(y_{c_1}, sw_{23}), (y_{c_0}, sw_{21})\}$. Then $\otimes B_{v_2} = (sw_{22}, \mathbf{0}, y_{v_2}, -, y_{v_2}(t) = f_{v_2}(sw_{22}(t), \mathbf{0}))$, where

$$y_{v_2}(t) = \begin{cases} 0(\text{close}), & sw_{22}(t) > 0, \\ 1(\text{open}), & sw_{22}(t) \leq 0. \end{cases} \tag{19}$$

Therefore, the observable trajectories of the subsystem $\otimes B_{v_2}$ is denoted as $Tr_4 = \{(sw_{22}, y_{v_2})\colon T \mapsto \mathscr{R} \times \mathscr{R} | \forall t \in T, y_{v_2}(t) = f_{v_2}(sw_{22}(t), \mathbf{0})\}$, where

$$y_{v_2}(t) = \begin{cases} 0(\text{close}), & sw_{22}(t) > 0, \\ 1(\text{open}), & sw_{22}(t) \leq 0. \end{cases} \tag{20}$$

Stage 1: verifying the Simulink block diagram satisfies the corresponding composition of contract.

The composition contract of *Controller*$_2$ is defined as $\otimes C_{v_2} = (sw_{22}, y_{v_2}, \phi_a^{v_2}, \phi_g^{v_2})$, where

$$\begin{aligned} \phi_a^{v_2} &:= \left\{ sw_{22} \in \mathscr{R}^T | \forall t \in T, sw_{22}(t) \in \mathscr{R} \right\}, \\ \phi_g^{v_2} &:= \left\{ \left(sw_{22}, \mathbf{y}_{\mathbf{v_2}} \right) \in (\mathscr{R} \times \mathscr{R})^T | \forall t \in T, y_{v_2}(t) = \begin{cases} 0(\text{close}), & sw_{22}(t) > 0, \\ 1(\text{open}), & sw_{22}(t) \leq 0. \end{cases} \right\}. \end{aligned} \tag{21}$$

According to Definition 11, $\forall t \in T$, we have $\{sw_{22}(t) \in \mathscr{R}^T | sw_{22}(t) \in \mathscr{R}\} \subseteq \phi_a^{v_2}$, and $Tr_4 \subseteq \phi_g^{v_2}$. Hence, $\otimes B_{v_2} \vDash \otimes C_{v_2}$.

Next, we will compose the *Decision*$_2$ and *Controller*$_2$ to verify requirement 2. Let $\otimes B_{\mathrm{Im\ 2}} = \otimes B_{d_2}\, ; \otimes B_{v_2} = (sw_{12}, \mathbf{0}, y_{v_2}, -, y_{v_2} = f_{v_2}(sw_{12}(t), \mathbf{0}))$, where

$$f_{\mathrm{Im\ 2}}(\mathbf{sw_{12}}(t), \mathbf{0}) = \begin{cases} 0(\text{close}), & sw_{12}(t) \geq 30, \\ 1(\text{open}), & sw_{12}(t) > 30. \end{cases} \tag{22}$$

The observable trajectories of $\otimes B_{\mathrm{Im\ 2}}$ are denoted as $Tr_{\mathrm{Im\ 2}} = \{(sw_{12}, y_{v_2})\colon T \mapsto \mathscr{R} \times \mathscr{R} | \forall t \in T, y_{\mathrm{Im\ 2}}(t) = f_{\mathrm{Im\ 2}} sw_{12}(t), \mathbf{0})\}$, where

$$y_{\mathrm{Im\ 2}}(t) = \begin{cases} 0(\text{close}), & sw_{12}(t) \geq 30, \\ 1(\text{open}), & sw_{12}(t) < 30. \end{cases} \tag{23}$$

The composition contract is defined as $\otimes C_{\mathrm{Im\ 2}} = (sw_{22}, y_{v_2}, \phi_a^{\mathrm{Im\ 2}}, \phi_g^{\mathrm{Im\ 2}})$, and

$$\begin{aligned} \phi_a^{\mathrm{Im\ 2}} &:= \left\{ sw_{12} \in \mathscr{R}^T | \forall t \in T, sw_{12}(t) \in \mathscr{R} \right\}, \\ \phi_g^{\mathrm{Im\ 2}} &:= \left\{ \left(sw_{12}, \mathbf{y}_{\mathbf{v_2}} \right) \in (\mathscr{R} \times \mathscr{R})^T | \forall t \in T', y_{v_2}(sw_{12}(t), \mathbf{0}) \right. \\ &= \left. \begin{cases} 0(\text{close}), & sw_{12}(t) \geq 30, \\ 1(\text{open}), & sw_{12}(t) < 30. \end{cases} \right\}. \end{aligned} \tag{24}$$

Stage 2: correctness verification. Our main goal in this step is to verify that the Simulink block diagram $(\otimes B_{v_2})$ is a correct implementation of system contract C_{req2}. $\forall t \in T$, we have $\{sw_{12}(t)\colon T \mapsto \mathscr{R} | sw_{12}(t) \in \mathscr{R}\} \subseteq \phi_a^{\mathrm{Im\ 2}}$, and $Tr_{\mathrm{Im\ 2}} \subseteq \phi_g^{\mathrm{Im\ 2}}$. Hence, $B_{\mathrm{Im\ 2}} \vDash C_{\mathrm{Im\ 2}}$. In terms of Definition 19, we get $\phi_a^{req2} \subseteq \phi_a^{\mathrm{Im\ 2}}, \phi_g^{\mathrm{Im\ 2}} \subseteq \phi_g^{req2}$. Then, $C_{\mathrm{Im\ 2}} \preccurlyeq C_{req2}$. According to Theorem 21, we have $B_{\mathrm{Im\ 2}} \vDash C_{\mathrm{Im\ 2}} \wedge C_{\mathrm{Im\ 2}} \preccurlyeq C_{req2} \Rightarrow B_{\mathrm{Im\ 2}} \vDash C_{req2}$.

In the example above, when an attacker adds a malicious logic bomb to Simulink, it can maliciously manipulate the input and output behaviors of any block in the block diagrams and the water level values read from the Sensor. This may lead to the block diagram's behavior not satisfying the system's formal specification. Considering from this perspective, our approach can identify and verify whether the designed CPS is planted with the logic bomb.

7. Conclusion and Future Work

In this paper, we presented a method to prove the correctness of Simulink block diagrams with *discrete-time* or *continuous-time* blocks using contract. This approach addressed the problem of proving that the system formal specifications are satisfied by composition and refinement of trajectories. We showed the usability of our proposals via a use case (as an example) which models the control system of a reservoir. Our method can improve the reliability of Simulink and reduce the development costs by performing early safety verification on verification of the target system.

In this work, we made the first step towards the contract-based verification of cyber-physical models in Simulink. Future work includes extending the work along several dimensions. First, this work considers single rate Simulink block diagrams. We will extend this work to consider the multiple sample times (multirate systems) diagrams. Another nontrivial extension involves applying our idea to prove the correctness of Simulink block diagrams mixture of *discrete-time* and *continuous-time* blocks. More challenging would be to develop a contract-based refinement approach to handle Stateflow. Second, this work only considers the blocks whose inputs-outputs can be explicitly represented through the mathematical relation. It is interesting to prove the correctness of the Simulink block diagrams, whose input-output behavior be expressed implicitly by a mathematical relation. Third, we plan to automatically verify the correctness of the Simulink block diagrams based on our method.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

Acknowledgments

This work was funded in part by the Chongqing Graduate Research and Innovation Project (grant No. CYB20098), the National Natural Science Foundation of China (Nos. 62032019, 61732019, 61672435, and 62002298), and the Capacity Development Grant of Southwest University (SWU116007). We are grateful for the valuable comments from Prof. Xinxin Liu and Prof. Zheng Yang on the earlier versions of this paper. We thank the anonymous referees whose suggestions improved the presentation.

References

[1] E. A. Lee, "Cyber physical systems: design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pp. 363–369, Orlando, FL, USA, 2008.

[2] G. Nicolescu and P. J. Mosterman, *Model-Based Design for Embedded Systems*, Crc Press, 2010.

[3] Mathworks, "Design cyber-physical systems with MATLAB and Simulink," *Mathworks*, 2022, September 2021, https://www.mathworks.com/discovery/cyber-physical-systems.html.

[4] N. Govil, A. Agrawal, and N. O. Tippenhauer, "On ladder logic bombs in industrial control systems," in *Computer Security*, pp. 110–126, Springer International Publishing, 2018.

[5] S. Tripakis, C. Sofronis, P. Caspi, and A. Curic, "Translating discrete-time Simulink to Lustre," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 4, no. 4, pp. 779–818, 2005.

[6] A. Cavalcanti, P. Clayton, and C. O'Halloran, "Control law diagrams in Circus," in *FM 2005: Formal Methods, International Symposium of Formal Methods Europe, Newcastle, UK, July 18-22, 2005*, vol. 3582 of Proceedings, ser. Lecture notes in computer science, pp. 253–268, Springer, Berlin, Heidelberg, 2005.

[7] M. Bostro, L. Morel, and M. Wald'en, "Stepwise development of Simulink models using the refinement calculus framework," in *International Colloquium on Theoretical Aspects of Computing*, C. B. Jones, Z. M. Liu, and J. Woodcock, Eds., vol. 4711 of Lecture Notes in Computer Science, pp. 79–93, Springer, Berlin, Heidelberg, 2007.

[8] S. K. Muduli, "Contract based development and refinement in Simulink [MS Thesis]," International Institute of Information Technology, Bangalore, 2017.

[9] M. Bostro, "Contract-based verification of Simulink models," in *Formal Methods and Software Engineering -13th International Conference on Formal Engineering Methods, ICFEM 2011*, S. C. Qin and Z. Y. Qiu, Eds., vol. 6991 of Lecture Notes in Computer Science, pp. 291–306, Springer, Berlin, Heidelberg, 2011.

[10] K. Ye, S. Foster, and J. Woodcock, "Compositional assume-guarantee reasoning of control law diagrams using UTP," in *From Astrophysics to Unconventional Computation*, pp. 215–254, Springer, Cham, 2020.

[11] J. W. K. Ye and S. Foster, "Compositional assume-guarantee reasoning of control law diagrams using UTP," 2018, April 2022, https://eprints.whiterose.ac.uk/129640/15/Compositional.

[12] C. Zhou and R. Kumar, "Semantic translation of Simulink diagrams to input/output extended finite automata," *Discrete Event Dynamic Systems*, vol. 22, no. 2, pp. 223–247, 2012.

[13] I. Dragomir, V. Preoteasa, and S. Tripakis, "The refinement calculus of reactive systems toolset," *International Journal on Software Tools for Technology Transfer*, vol. 22, no. 6, pp. 689–708, 2020.

[14] V. Preoteasa, I. Dragomir, and S. Tripakis, "Compositional semantics and analysis of hierarchical block diagrams," in *International Symposium on Model Checking Software*, vol. 9641 of Lecture notes in computer science, pp. 38–56, Springer, Cham, 2016.

[15] C. Chen and J. S. Dong, "Applying timed interval calculus to Simulink diagrams," in *Formal Methods and Software Engineering, 8th International Conference on Formal Engineering Methods, ICFEM*, Z. Liu and J. He, Eds., vol. 4260 of Lecture Notes in Computer Science, pp. 74–93, Springer, Berlin, Heidelberg, 2006.

[16] C. Q. Chen, J. S. Dong, and J. Sun, "A formal framework for modelling and validating Simulink diagrams," *Formal Aspects of Computing*, vol. 21, no. 5, pp. 451–483, 2009.

[17] O. Bouissou and A. Chapoutot, "An operational semantics for Simulink's simulation engine," in *SIGPLAN/SIGBED Conference on Languages, Compilers and Tools for Embedded Systems 2012, LCTES '12*, pp. 129–138, Beijing, China, 2012.

[18] L. Zou, N. J. Zhan, S. L. Wang, F. Martin, and S. C. Qin, "Verifying Simulink diagrams via a hybrid hoare logic prover," in *2013 Proceedings of the International Conference on Embedded Software (EMSOFT)*, pp. 1–10, Montreal, QC, Canada, 2013.

[19] T. Bourke, F. Carcenac, B. Pagano, C. Pasteur, and M. Pouzet, "A synchronous look at the Simulink standard library," vol. 16, no. 5, pp. 176:1–176:24, 2017.

[20] A. Benveniste, B. Caillaud, D. Nickovic et al., "Contracts for system design," *Foundations and Trends® in Electronic Design Automation*, vol. 12, no. 2-3, pp. 124–400, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2448010, 11 pages
https://doi.org/10.1155/2022/2448010

Research Article

Intrusion Detection Model for Wireless Sensor Networks Based on MC-GRU

Zhou Jingjing,[1] **Yang Tongyu**,[1] **Zhang Jilin**,[2] **Zhang Guohao**,[1] **Li Xuefeng**,[1] **and Pan Xiang**[1]

[1] *School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China*
[2] *School of Computer Science, Hangzhou Dianzi University, Hangzhou 310018, China*

Correspondence should be addressed to Zhou Jingjing; zhoujingjing@zjgsu.edu.cn

Received 15 June 2022; Accepted 18 August 2022; Published 5 September 2022

Academic Editor: Chenglu Jin

A crucial line of defense for the security of wireless sensor network (WSN) is intrusion detection. This research offers a new MC-GRU WSN intrusion detection model based on convolutional neural networks (CNN) and gated recurrent unit (GRU) to solve the issues of low detection accuracy and poor real-time detection in existing WSN intrusion detection algorithms. MC-GRU uses multiple convolutions to extract network data traffic features and uses the high-level features output after convolution operations as input parameters of the GRU network, which strengthens the learning of spatial and time series features of traffic data and improves the detection performance of the model. The experiment results based on the WSN-DS dataset show that the overall detection accuracy of the four types of attack of black hole, gray hole, flooding, and scheduling and normal behaviors reaches 99.57%, and it is also better than the existing WSN intrusion detection algorithms in real-time performance and classification ability.

1. Introduction

The use of WSN has grown in popularity across many industries, including aviation, industry, and the environment, as a result of the quick advancement of wireless communication and sensor technologies, breaking the limitations of traditional methods to monitor and collect data in harsh environments [1]. However, the security problem of WSN brings new threats. Intrusion attacks against WSN can cause great damage to the safety of individual or collective life and property. Accurate detection of various types of attacks in WSN can provide a reliable security guarantee for the network. As a result, the WSN intrusion detection method has emerged as a major area of study in the field of network security today.

A distributed wireless communication network made up of management, sink, and sensor nodes is known as a WSN. Because wireless signals are divergent and sensor nodes are also limited by their own computing power, storage capacity, and wireless communication capabilities, WSN is facing security threats such as data leakage and data forgery [2]. Existing research generally adopts a two-layer defense mechanism to ensure the security of WSN. The first layer of the defense mechanism includes data encryption, data authentication, and security protocols, but with the continuous breakthrough of network attack technology, the effect of the first layer of the defense mechanism gradually becomes less than ideal. As the second layer defense mechanism to protect WSN security, intrusion detection technology can effectively compensate for the deficiency of the first layer defense mechanism and reduce losses caused by network attacks [3].

Because WSN has low computing and communication capabilities, traditional network intrusion detection algorithms are not suitable for directly using in WSN. At present, most research on WSN intrusion detection uses traditional machine learning methods to analyze network traffic data. Due to the growth in both the network's size and its user base, the WSN network will generate high-dimensional traffic data, and the traditional machine learning approach would encounter issues like poor feature extraction and detection accuracy, which cannot meet such an application environment [4].

Compared to machine learning methods for detecting intrusions, deep learning-based intrusion detection can reduce computational complexity and increase the ability to learn the characteristics of data traffic, which can improve the precision of the detection model [5].

The MC-GRU intrusion detection algorithm, based on CNN and GRU, is suggested in this paper. It takes into account the detection accuracy and feature selection of the intrusion detection model in considerable detail. MC-GRU extracts the basic characteristics of network data traffic through CNN, uses the advanced features output after the convolution operation as the input parameters of the GRU network for time series feature learning, and then uses the dropout mechanism to suppress the occurrence of overfitting of the detection model and improve the generalization ability of the WSN intrusion detection model, and finally, the softmax function is used for multiclassification. Compared to the existing WSN intrusion detection model, it can increase the detection accuracy, real-time detection, and various multiclassification capabilities of various types of WSN attack. The experimental results show that MC-GRU is superior to existing WSN intrusion detection algorithms in terms of detection accuracy, real-time performance, and classification ability and is an effective solution to the second-layer defense mechanism of WSN.

The following describes the organizational structure of this paper. Section 1 describes the background of WSN intrusion detection research and shows the MC-GRU implementation process. Section 2 summarizes related domestic and international research on WSN intrusion detection. Section 3 describes the structure and related principles of MC-GRU. The results of the MC-GRU experimental data are displayed and analyzed in the fourth section, and the fifth section summarizes this paper.

2. Related Work

In recent years, an abundance of papers have carried out research on multiclassification of traffic types in WSN intrusion detection. Paper [6] designs the corresponding intrusion detection algorithm based on the difference in node resources and uses the lightweight random forest algorithm for WSN intrusion detection in the cluster head node with relatively scarce resources. The deep random forest algorithm further detects attack behavior that cannot be detected by the cluster head node, improving the real-time detection and accuracy of the WSN intrusion detection model, but the multiclassification effect still needs to be improved. In paper [7], the data density and the feature distance are added to the fuzzy clustering algorithm, and the fuzzy membership obtained is used as the fuzzy factor of the fuzzy support vector machine, which improves the detection efficiency of the model and improves the multiclassification effect. Paper [8] combines the self-encoding network and support vector machine (SVM) to realize the detection of WSN intrusion, which is beneficial for the extraction of high-dimensional spatial information, but the precision of multiclassification needs to be improved. Paper [9] proposes an adaptive AP clustering algorithm based on the adaptive AP algorithm and the clustering algorithm, which reduces the consumption of sensor node storage space and improves clustering efficiency, but the detection accuracy needs to be improved. Paper [10] uses SVM kernel functions such as RBF, PLOY, and sigmoid for data classification, and the best detection effect is 91%, and still room for improvement with the performance. Paper [11] uses a deep neural network (DNN) with different layers to verify the detection performance on the WSN-DS dataset. Overall, multiclassification performance is better, but the false positive rate needs to be improved. For DNN traffic with layers 1 to 5, the average false positive rates for the types were 2.34%, 4.20%, 3.4%, 4.98%, and 2.7%, respectively. Paper [12] extracts the output of each level from the trained deep CNN and implements a linear SVM and a classifier with a nearest neighbor (1-NN), which improves the detection accuracy of attack types with a small sample size in the dataset. However, the detection rate of the model needs to be improved. The K-nearest neighbor node (KNN) classification algorithm used in the paper [13] uses the compressed proximity algorithm to reduce and cluster the original data to locate the sample's center of gravity, which can result in a higher detection rate and relatively less error. However, the detection effect of some multiclassification needs to be improved.

Other studies detect a certain type of attack or do not classify the attack types. Paper [14] uses the K-means algorithm based on enhanced particle swarm optimization to detect spoofing attacks according to the strength of the signal received from the physical layer. Paper [15] uses the mini batch K-means algorithm and SVM to achieve WSN intrusion detection and uses randomly generated small batch data samples for clustering, which improves the convergence speed of the model and greatly reduces the calculation time; it is suitable for WSN environment with large sample size, but the algorithm cannot detect specific attack types. In order to detect flooding attacks, the paper [16] proposes a KNN-based WSN intrusion detection system that compares each node's cutoff value and distance function to identify aberrant nodes. Paper [17] fakes a destination node in WSN to induce a black hole node attack, finds the black hole node through identity verification and location information of the attacking node, and removes it from WSN.

As mentioned above, WSN intrusion detection has seen significant advancements in research, but there are still certain issues, such as poor detection accuracy, low real-time detection performance, and poor multiclassification effect. Meanwhile, with the development of networks and big data, the data to be detected will be more complex, and the traditional WSN intrusion detection method cannot anymore meet the current network environment.

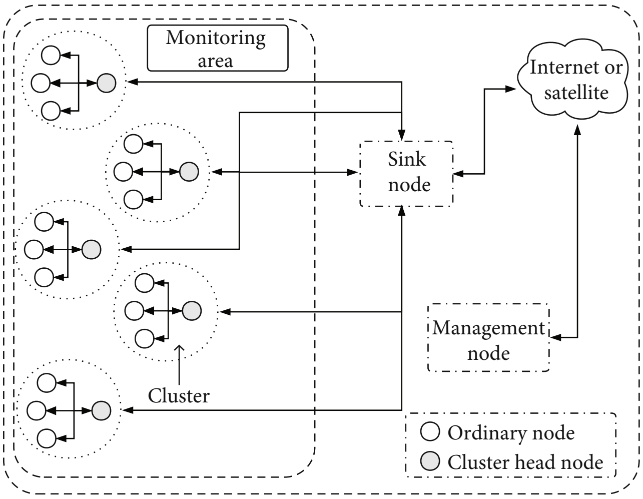


Figure 1: WSN structure [18].

3. WSN Intrusion Detection Model

3.1. Background of Problem

3.1.1. WSN Structure. The WSN MC-GRU intrusion detection model proposed in this paper is based on the layered structure in the topology of the WSN. As depicted in Figure 1, in the layered structure, the wireless sensor nodes are separated into ordinary sensor nodes, cluster head nodes, sink nodes, and management nodes [18], and the node performance improves in turn. The ordinary sensor nodes in the monitored area are responsible for completing the data monitoring and collection tasks issued by the management node and sending the collected data to the cluster head node. The data gathered by the ordinary sensor nodes in the cluster is first preprocessed by the cluster head node before being sent to the sink node. The WSN MC–GRU intrusion detection model in the sink node performs intrusion detection on the data from the cluster head node and then sends the detection results and data to the management node.

3.1.2. Type of Attack. The MC-GRU model suggested in this paper is primarily intended for the detection of the following four types of attacks.

(1) Black Hole Attack. The black hole node discards all data packets from the source node and blocks the communication service with the destination node. The specific description is shown in Figure 2 [17]. The essence of a black hole attack is a routing attack. The source node *S* needs to communicate with the destination node *D* through one or more nodes and will initiate a routing request. At this time, the black hole node will indicate that it is the most suitable relay node to the destination node, but in the data transmission process, it discards all data packets from the source node *S*, resulting in transmission holes.

(2) Gray Hole Attack. The gray hole node also discards the data packets from the source node, but not all of them, only discard a certain type of data packet or discards the data packets immediately and prevent the data packets from being forwarded to the base station.

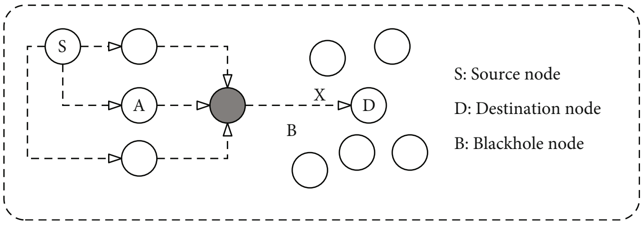


Figure 2: Black hole attack [17].

(3) Flooding Attack. Flood nodes send or broadcast a high quantity of worthless routing request packets or data packets, consuming the limited resources of WSN nodes, occupying bandwidth meaninglessly, and making WSN communication unable to keep smooth. In addition, abnormal nodes in flood attack route request (RREQ) messages are sent more frequently than normal nodes [16].

(4) Scheduling Attack. A scheduling attack occurs in the initialization phase of the low energy adaptive clustering hierarchy (LEACH) protocol when WSN starts to randomly select the cluster head node; then, the scheduling node pretends to be the cluster head node, and the scheduling node gives all ordinary sensor nodes the same time stamp for sending data. Finally, data conflict between sensor nodes is lost [7].

3.2. MC-GRU Model. Assuming that the WSN dataset that needs anomaly detection is $P = \{x_1, x_2, \cdots, x_N\}$, the preprocessed eigenvalues of the *i*-th traffic data in the dataset are expressed as $x_i = \{x_i^1, x_i^2, \cdots, x_i^M\}$. The eigenvalue of each piece of traffic data obtains the corresponding probability value through the operation of the MC-GRU model, thus judging the type of the piece of traffic. Among them, M is the number of characteristics that each traffic data sample possesses, and N denotes the total number of traffic data samples that are included in the dataset.

The structure of the WSN MC-GRU intrusion detection model established in this paper is shown in Figure 3. The model contains multiple convolutional layers (MC) and a GRU layer. There are also a pooling layer and a batch normalization layer (BN) in between. The MC uses a convolutional neural network with three layers and multiple convolution kernels to extract features from the data to obtain the deep features of the data stream. The pooling layer compresses the data obtained from the convolutional layer through pooling calculation to improve the processing efficiency of the lower network and accelerate the model convergence. The BN improves the nonlinear expression ability of the network model. To further improve the feature learning ability and processing efficiency of the model, an improved GRU layer based on long short-term memory (LSTM) is added after the batch normalization layer to learn the context and time series features in the data. The dropout layer removes some neurons in a certain proportion to reduce the complex coadaptive relationship between

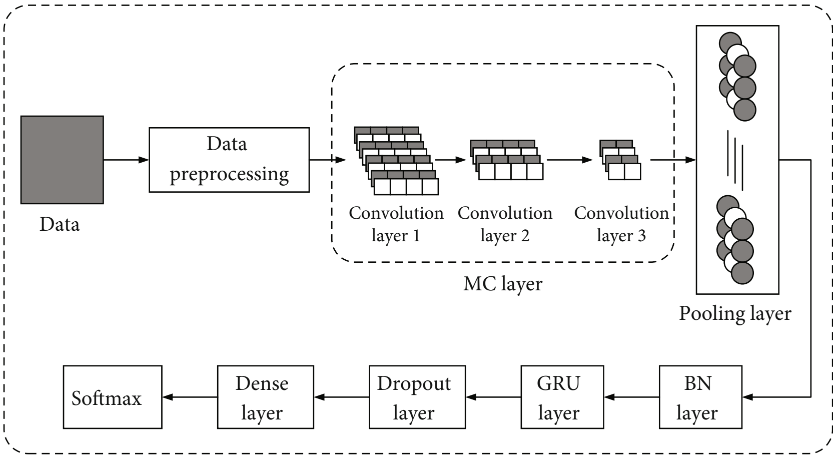


Figure 3: MC-GRU model structure.

neurons. Finally, the processed vector features are inputted into the dense layer for feature fusion, and the softmax logistic regression layer is used for final classification and output classification results.

3.2.1. Data Preprocessing. The nonnumerical properties of the dataset used in the experiment are numericalized and normalized to obtain an input format that can conform to the MC-GRU model.

3.2.2. MC Layer. With the development of wireless sensor networks, the detected traffic data will become large in sample size and complex in features. The sample data is used as the input of the CNN model, and each filter can be used to perform convolution operations on it. Performing feature extraction to obtain advanced features will greatly improve the feature extraction capability of the WSN intrusion detection model for complex traffic data. MC-GRU contains consecutive 3 layers of convolution. The number of convolution kernels is 128, 64, and 64. Iteratively extracts traffic features using multiple convolutional networks to obtain deeper and more complex features of the global data flow. All three-layer convolutional networks use the linear rectification function ReLU as the activation function. In the process of network-based intrusion detection, the sample data of WSN intrusion detection is not necessarily completely linear, and the output obtained by the signature function is a linear combination of the input. Therefore, the rectified linear unit ReLU activation function is selected instead of the sigmoid function. The rule activation function can also reduce the calculation of the model. It can improve training and detection efficiency [19], as shown in the expression:

$$\mathrm{ReLu}(x) = \max\,(x, 0). \tag{1}$$

The convolutional network's first layer's output is then as follows:

$$x_1 = \begin{cases} b_1 + w_1 * \mathrm{x} & b_1 + w_1 * x > 0, \\ 0 & b_1 + w_1 * x \leq 0. \end{cases} \tag{2}$$

The output of the second- and third-layer convolutional networks is as follows:

$$x_m = \begin{cases} b_m + w_m * \mathrm{x}_{m-1} & b_m + w_m * x_{m-1} > 0, \\ 0 & b_m + w_m * x_{m-1} \leq 0. \end{cases} \tag{3}$$

Among them, x in formula (2) is the eigenvalue of each piece of detected and preprocessed traffic data, which will be input into the MC-GRU model; b_1 and w_1 are the bias and weight matrices of the first-layer convolutional network; in formula (3), x_{m-1} is the output of the previous layer of the convolutional network; b_m and w_m are the bias and weight matrices of each layer of the convolutional network, $m = 2$, 3; "$*$" represents the convolution operation.

3.2.3. Pooling Layer. The pooling layer is used to extract the output features from the previous layer. It can reasonably reduce the dimension of the current detected data traffic feature vector, which can reduce the complexity of the entire WSN intrusion detection model and reduce the calculation after the pooling layer. The maximum value in the pooling filter is taken for the input feature vector, that is, the strongest feature part is retained. The calculation process is as follows:

$$\begin{cases} H_{\mathrm{out}} = \dfrac{h_{\mathrm{in}} - h_{\mathrm{filter}}}{T+1}, \\ W_{\mathrm{out}} = \dfrac{w_{\mathrm{in}} - w_{\mathrm{filter}}}{T+1}. \end{cases} \tag{4}$$

Among them, H_{out} and W_{out} are the height and width of the feature vector output after calculation by the pooling layer, T is the step size of the pooling filter scan, h_{in} and h_{filter} are the feature vector output from the previous layer and the height of the pooling filter, respectively, w_{in}, and w_{filter} are the feature vector output from the previous layer and the width of the pooling filter, respectively.

3.2.4. BN. When training the MC-GRU model, the parameters are updated, except that the data from the first input

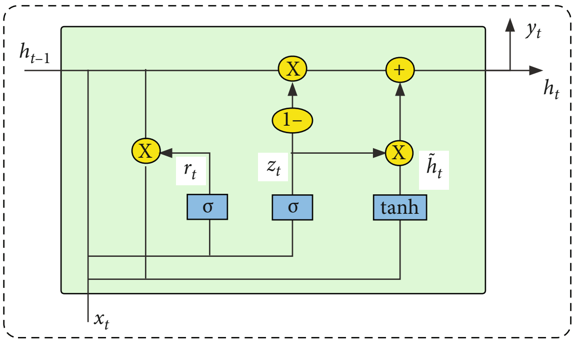


Figure 4: The structure of the GRU memory cell [21].

layer of the model are normalized, the input data distribution of each layer of the model will continue to change, and the network will learn new data distributions, which will reduce the convergence speed of the model. Therefore, batch normalization is added between the MC layer and the GRU layer to normalize the WSN detection model, so that the input samples are not correlated, and the data distribution of the output value of the MC layer and the input value of the GRU layer is closer to the data distribution of the original sample, which improves the convergence speed of the model and prevents the appearance of gradient explosion [20].

3.2.5. GRU Layer. Taking into account the computing power, detection accuracy, real-time requirements of wireless sensor nodes, and the further improvement of the classification ability of the WSN intrusion detection model for various attack categories, an improved GRU network based on LSTM is introduced to learn the contextual features of data flow and timing information [21]. There are only two gates in the GRU model: update gate z_t and reset gate r_t. The specific structure is shown in Figure 4.

The function of the reset gate is to forget the information h_{t-1} of the hidden layer unit at the previous moment:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t], b_r). \tag{5}$$

After forgetting, h_{t-1} remaining information: $r_t \cdot h_{t-1.}$

The function of the update gate is to control the balance between the hidden layer state h_{t-1} at the previous moment and the current input information:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t], b_z). \tag{6}$$

Enter information, here is the $r_t \cdot h_{t-1}$ after forgetting:

$$\tilde{h}_t = \tanh\left(W_{\tilde{h}_t} \cdot [r_t * h_{t-1}, \mathrm{x}_t], b_h\right). \tag{7}$$

h_t after balance:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t, \tag{8}$$

where σ is the sigmoid activation function, and tanh is the hyperbolic tangent function:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}, \tag{9}$$

$$\text{Tanh}(x) = \frac{1 - e^{-2x}}{1 + e^{2x}}. \tag{10}$$

3.2.6. Dropout Layer. Remove some neurons according to a certain proportion to reduce the complex coadaptation relationship between neurons [19]. In a neural network model, if the model has too many parameters and too few training samples, the resulting model will overfit. Overfitting significantly affects how well the model performs, so using dropout in the WSN intrusion detection model can improve the overall performance of the model to some extent.

In the last layer of the proposed MC-GRU network model, softmax function is used as the classifier, and the type of traffic is judged according to the probability value obtained. The mathematical expression is expressed as follows:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}. \tag{11}$$

The configuration of the specific parameter of the MC-GRU model is shown in Figure 5.

3.2.7. Adam. The Adam optimizer proposed by Kingma and Lei Ba can iteratively update the weights of the network model based on training data. It is implemented simply and computed efficiently, it uses less memory, and the scaling change of the gradient has no impact on the updating of its parameters. It is appropriate for cases involving a lot of data and parameters, such WSN intrusion detection.

3.2.8. Categorical_Crossentropy. The difference between the probability distribution obtained by the present training and the genuine distribution is assessed using the cross-entropy loss function. Typically, it works in conjunction with the softmax function to achieve multiclassification.

3.3. WSN Intrusion Detection Framework. The WSN intrusion detection framework based on MC-GRU is shown in Figure 6. According to the resources of each node, the corresponding data operations are shared. This hierarchical structure can disperse the energy overhead, reduce the communication burden, and achieve energy savings. The WSN intrusion detection model based on MC-GRU can be divided into three steps:

Step 1. In the data collection stage, common sensors are distributed in the monitoring area to perceive the environment and collect data. Because common sensor resources are limited, common sensor nodes can only perform some simple processing of the collected data before sending them to the corresponding cluster head node.

Step 2. In the data preprocessing stage, the cluster head node has richer resources than ordinary sensor nodes and is used

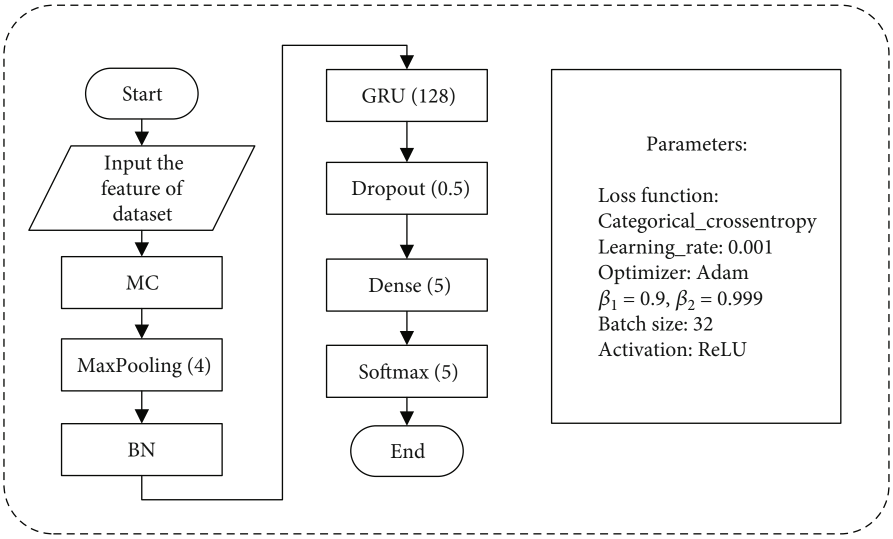


Figure 5: MC-GRU model flow chart and related configuration.

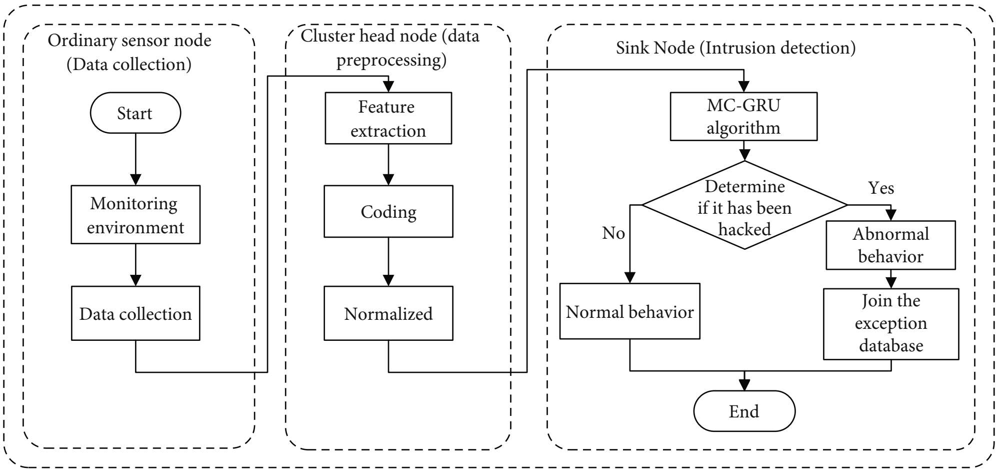

Ordinary sensor node (Data collection)
Start
Monitoring environment
Data collection
Cluster head node (data preprocessing)
Feature extraction
Coding
Normalized
Sink Node (Intrusion detection)
MC-GRU algorithm
Determine if it has been hacked
Yes
No
Abnormal behavior
Join the exception database
Normal behavior
End

Figure 6: WSN intrusion detection process.

to perform data preprocessing operations such as feature extraction, character encoding, and data normalization on the data samples sent by common sensor nodes in the cluster and send the processed data to the sink node for intrusion detection.

Step 3. In the intrusion detection stage, deploy the trained MC-GRU model at the sink node. The sink node has more abundant resources and stronger computing power than ordinary sensor nodes and cluster head nodes and is suitable for processing cluster head nodes. The completed data are subjected to intrusion detection.

4. Experimental Results and Analysis

4.1. Experimental Data. The operating system used in this experiment is Windows 10, the processor is Intel(R) Core(TM) i5-8500 CPU@3.00GHz, the memory is 16 GB, python 3.7 is used to run through the whole scheme, numpy and pandas are used for data processing, and tensorflow is used. Build the model architecture with the keras framework and finally use matplotlib for visualization.

WSN-DS is an intrusion detection dataset for WSN constructed by Almomani et al. to describe normal behavior and four types of DoS attacks in WSN. The dataset is obtained by simulating the wireless sensor network environment using the NS-2 simulator. Each attribute in the dataset is based on the features analyzed by the LEACH hierarchical routing protocol. By tracking and analyzing the hierarchical routing protocol, it can well reflect the work of the current network environment condition. Each data record in this dataset consists of 18 inherent attributes and 1 class identifier [22]. In addition to normal behavior, the class identifier has 4 possible values: black hole, gray hole, flooding, and scheduling attacks, as shown in Table 1. The WSN-DS dataset has 374661 traffic data points in total. Take 80% and 20% of the WSN-DS dataset and divide it into training and test sets, with 20% of the

Table 1: WSN-DS dataset class distribution.

Type of attack	Number
Normal	340066
Black hole	10049
Flooding	3312
Gray hole	14596
Scheduling	6638

Table 2: Experimental division of WSN-DS dataset.

Type of attack	Training set (80%, of which 20% is a validation set)	Test set (20%)
Normal	272101	67965
Black hole	8006	2043
Flooding	2681	631
Gray hole	11611	2985
Scheduling	5329	1309
All	299728	74933

training set serving as the validation set. Table 2 displays how many of each type there are.

Since the eigenvalues of the WSN-DS wireless sensor dataset used in this paper are all numerical, the feature encoding process is omitted, and the data normalization operation is performed directly. Normalization of traffic data can eliminate the difference between data of different dimensions. To ensure the reliability of the training results, these characteristics are assigned to $[0, 1]$. In this article, the min-max normalize method [23] of the following formula is used to process the data, which only compresses the data and does not change the initial information of the data samples.

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{12}$$

Among them, x is the sample value, the sample data's greatest value is $x_{\max}$, and its smallest value is $x_{\min}$.

4.2. Evaluation Indicators. The results of intrusion detection include the following four types: true positive (TP) indicates that it is actually normal behavior, the prediction is also the number of normal behavior, and false positive (FP) indicates that it is actually abnormal behavior and is predicted to be normal behavior. The number of true negatives (TN) is actually abnormal behavior, correctly predicted as abnormal behavior, and false negative (FN) represents the number of normal behaviors that are misidentified as abnormal behavior [24]. Details are shown in Table 3. According to the above four detection results, the accuracy rate, false positive rate, and recall rate are further evolved, which are used as evaluation indicators for intrusion detection technology in this paper.

Table 3: Confusion matrix.

Predicted	True	
	Positive	Negative
Positive	TP	FP
Negative	FN	TN

(1) The accuracy rate indicates the ratio of the number of samples that correctly identify abnormal samples and normal samples to the total number of samples. The calculation formula is as follows:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \tag{13}$$

(2) The false positive rate (FPR) indicates that among the samples whose true values are abnormal, the probability of being predicted to be a normal sample is calculated as:

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}. \tag{14}$$

(3) The recall rate (true positive rate, TPR) indicates that the true value is in the normal sample, and the probability of being predicted to be a normal sample is calculated as:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{15}$$

4.3. Analysis of Results. In the deep learning model, the learning rate is a hyperparameter which controls the degree to which we adjust the network weights according to the loss gradient. To choose an optimal learning rate, this article first selects different learning rate values for comparison experiments, as shown in Figure 7, these reflect the detection accuracy of the MC-GRU model under different learning rates. According to Figure 7's analysis of the experimental findings, relatively speaking, when the learning rate is 0.001, the detection accuracy of the model is relatively high, reaching 0.9957. Therefore, in this article, the model's learning rate is set to 0.001.

Figure 8 depicts the correlation between the accuracy and the number of iterations, with training acc denoting the accuracy of the training set and validation acc denoting the accuracy of the validation set.

According to Figure 9, there is a correlation between the number of iterations and the loss value, where the training loss corresponds to the loss value of the training set and the validation loss to the loss value of the validation set.

Observing the curves of the two images in Figures 8 and 9, it is clear that the overall trend of the accuracy of the

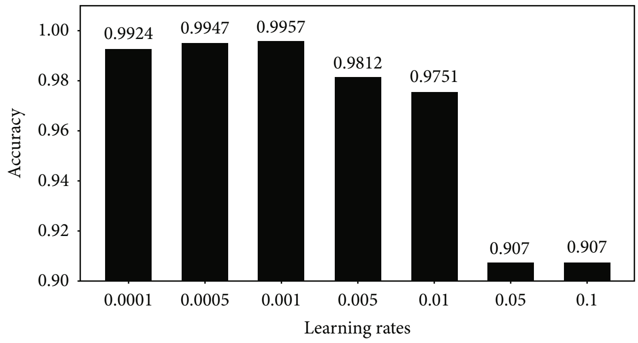


Figure 7: MC-GRU detection accuracy with different learning rates.

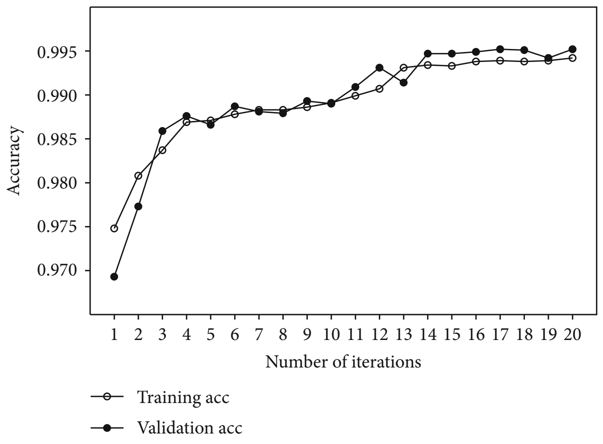


Figure 8: Accuracy curve.

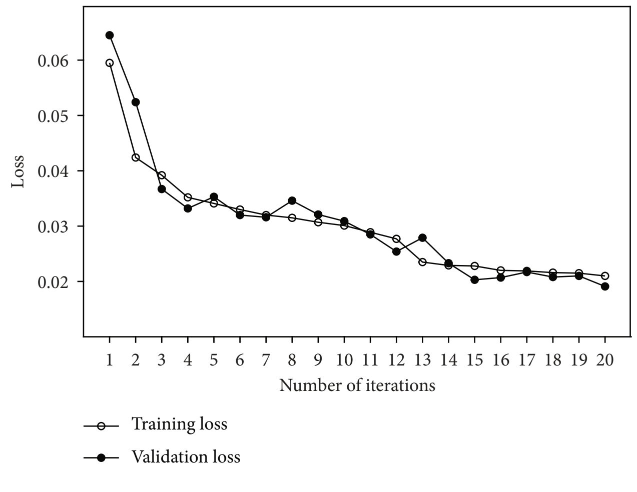


Figure 9: Loss curve.

training set and the validation set rises with each increment in the number of iterations. In cases when the epoch is more than 13, the accuracy of testing and verification tends to be stable, reaching a maximum of 0.9957, while the loss curve gradually decreases, and it can be concluded that the model has converged to the best state.

Table 4: Multiclassification results.

Performance	Normal	Black hole	Gray hole	Flooding	Scheduling
TPR	0.998	0.953	0.989	0.904	0.996
FPR	0.017	0.0014	0.0004	0.0009	0.00006
Acc	0.999	0.992	0.957	0.995	0.930

This article is mainly for the classification and identification of black hole, gray hole, flooding, scheduling attacks, and normal traffic data in the WSN-DS dataset, as shown in Table 4. Accuracy, false positive, and recall are for several types of data traffic. It is obvious from Table 4 that the detection accuracy of black hole, gray hole, flooding, scheduling, and normal types is all above 0.93; and the overall detection accuracy has reached 99.57%. It shows that the scheme proposed in this paper has a good multiclass detection effect for these types of data traffic in wireless sensor networks.

The real-time detection performance of the overall WSN intrusion detection system can be enhanced by increasing the intrusion detection rate of the traffic data model. As can be observed, the MC-GRU model took 2064.10 s to train, which is less than the training time of other models. It is clear that the model suggested in this paper has a low level of temporal complexity. The test time is 8.89 s, and the average detection time of a piece of traffic data is $8.89/74933 = 1.186 \times 10^{-4}$s, so the real-time detection performance of the MC-GRU model is high. It can also be proved that although the MC-GRU-based WSN intrusion detection model has a more complex model structure and more parameters, the training speed does not decrease with the deepening of the model but improves the real-time detection.

To be able to prove the effect of the MC-GRU algorithm on WSN intrusion detection, as shown in Figures 10–12, it is given that in the case of all using the WSN-DS dataset, some algorithms are selected for experimental comparison, including Naive Bayes (NB), SVM, KNN, and CNN, in addition to CNN-LSTM based on CNN and LSTM, to compare the accuracy, false positive, and recall of these intrusion detection algorithms.

Figure 10 shows the accuracy comparison between MC-GRU and the comparison algorithm. In contrast to the comparison algorithm, the accuracy rates of the MC-GRU algorithm proposed in this paper for detecting the five behaviors are 0.999, 0.992, 0.957, 0.995, and 0.930, respectively. The accuracy of black hole attack, gray hole attack, flood attack, and normal behavior detection is the best. Overall, the accuracy of the MC-GRU beats that of other algorithms.

Figure 11 shows the recall comparison results between MC-GRU and the comparison algorithm. In contrast to the comparison algorithm, the recall rates of the MC-GRU algorithm proposed in this paper for detecting five behaviors are 0.998, 0.953, 0.989, 0.904, and 0.996, respectively, and the accuracy in the detection of gray hole attacks and scheduling attacks is the best. SVM-RBF has a somewhat greater recall rate than MC-GRU when detecting normal behavior

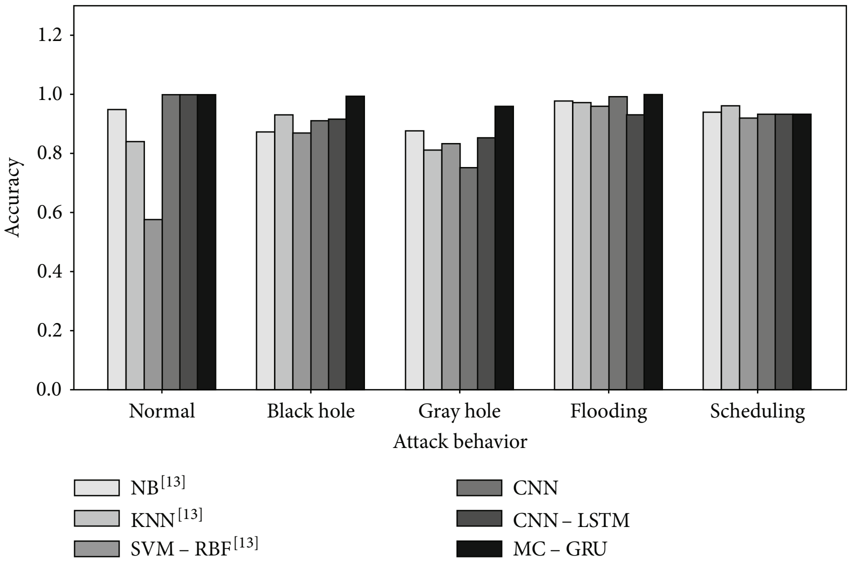


Figure 10: Comparison of the accuracy of six classification algorithms.

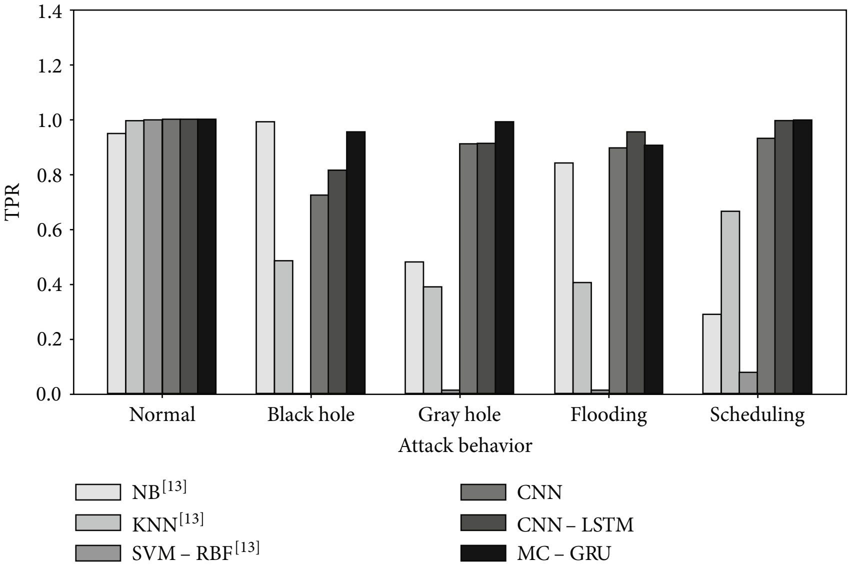


Figure 11: Comparison of recall rates of six classification algorithms.

and black hole attacks. For the detection of flooding attacks, the recall rate of CNN-LSTM is slightly higher than that of MC-GRU, but when combined with Table 5, it is found that the detection rate of MC-GRU is much higher than that of CNN-LSTM. As can be seen in the picture, MC-GRU outperforms other algorithms in terms of recall.

Figure 12 compares the false positive rate results between MC-GRU and the comparison algorithm. In contrast to the comparison algorithm, the recall rates of the MC-GRU algorithm proposed in this article to detect the five behaviors are 0.017, 0.0014, 0.0004, 0.0009, and 0.00006, respectively. The false positive rate is optimal in the detection of different behaviors. Although the false positive rate of SVM-RBF for detecting black hole attacks, gray hole attacks, flooding attacks, and scheduling attacks is similar to that of MC-GRU, the false positive rate for normal behavior detection reaches 0.922, and the overall false positive rate is too high.

In summary, the detection performance of MC-GRU for various traffic types in the WSN-DS dataset is significantly better than that of other models. When the MC-GRU model detects complex traffic attack types, it uses multiple convolutions to extract features from the original data and then introduces GRU to learn the context and time series features of the data, which makes the model more capable of extracting data flow features to speed up the convergence of the model. Therefore, compared to other models, MC-GRU has higher detection accuracy, faster detection rate, and better multiclassification effect.

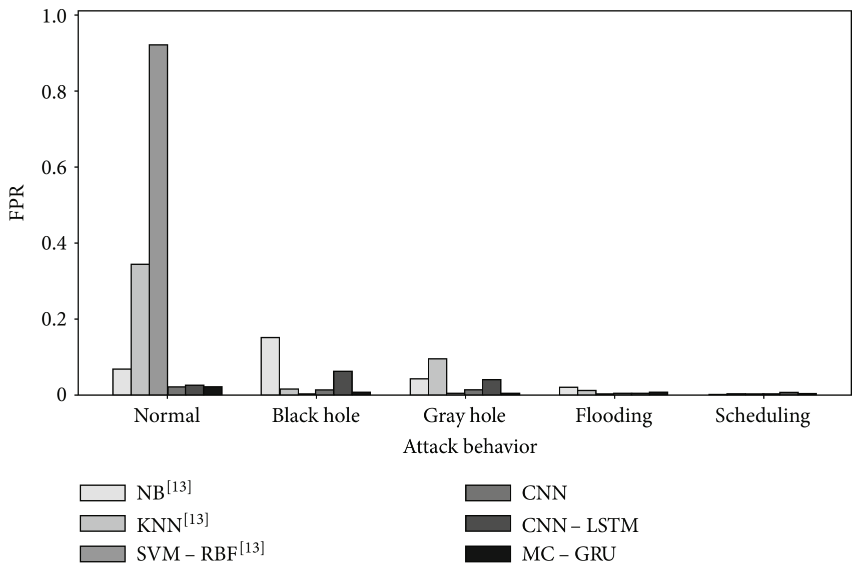


Figure 12: Comparison of false alarm rates of six classification algorithms.

Table 5: Model detection real-time comparison.

Model	Training time/s	Detecting time of each data/s
SVM	1837.18	5.404×10^{-4}
Simple RNN	2101.21	6.927×10^{-4}
CNN-LSTM	4283.63	2.461×10^{-4}
MC-GRU	2064.10	1.186×10^{-4}

5. Summary

Targeting the variety of current WSN traffic attack types, a WSN MC-GRU intrusion detection model is proposed. The experiment's findings demonstrate that the MC-GRU model's test set detection accuracy is 99.57%, and it can identify black hole attacks, gray hole attacks, flooding attacks, scheduling attacks, and normal behavior traffic types with high accuracy. Compared with other detection models, it significantly improves the ability of multiclassification of WSN attack types. At the same time, the detection rate is not slowed down due to the deepening of the model, which ensures the real-time detection of the model.

Data Availability

The data set WSN-DS used in this article can be obtained at https://gitee.com/he-feifan/matlab_workspace/blob/master/WSN-DS.csv.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is funded by the project of Natural Science Foundation of Zhejiang Province (no. LY19F020006)

References

[1] J. S. Pan, F. Fan, S. C. Chu, H. Q. Zhao, and G. Y. Liu, "A lightweight intelligent intrusion detection model for wireless sensor networks," *Security and communication Networks*, vol. 2021, Article ID 5540895, 15 pages, 2021.

[2] G. Kalnoor and S. Gowrishankar, "Minimizing energy consumption for intrusion detection model in wireless sensor network," in *Applications of Artificial Intelligence and Machine Learning*, pp. 527–537, Springer, Singapore, 2021.

[3] T. Zhang, D. Han, M. D. Marino, L. Wang, and K. C. Li, "An evolutionary-based approach for low-complexity intrusion detection in wireless sensor networks," *Wireless Personal Communications*, vol. 8, pp. 1–24, 2021.

[4] Y. Yan, L. Qi, J. Wang, Y. Lin, and L. Chen, "A network intrusion detection method based on stacked autoencoder and LSTM," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, Dublin, Ireland, 2020.

[5] M. Mittal, C. Iwendi, S. Khan, and A. Rehman Javed, "Analysis of security and energy efficiency for shortest route discovery in low-energy adaptive clustering hierarchy protocol using Levenberg-Marquardt neural network and gated recurrent unit for intrusion detection system," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 6, article e3997, 2021.

[6] D. O. Rui-hong, Y. A. Hou-hua, Z. H. Qiu-yu, and L. I. Xue-yong, "Distributed WSN intrusion detection model based on deep forest algorithm," *Journal of Lanzhou University of Technology*, vol. 46, no. 4, p. 103, 2020.

[7] L. Fu-cai, W. Fei, C. Qian, H. Jindong, and K. Liang, "Machine learning-based intrusion detection technology for wireless sensor networks," *Journal of Harbin Engineering University*, vol. 41, no. 3, pp. 433–440, 2020.

[8] N. Gao, L. Gao, Y. Y. He, and H. Wang, "A lightweight intrusion detection model based on autoencoder network with feature reduction," *ACTA Electonica Sinica*, vol. 45, no. 3, p. 730, 2017.

[9] J. Jiang, Z. F. Wang, T. M. Chen, C. C. Zhu, and B. Chen, "Adaptive AP clustering algorithm and its application on

intrusion detection," *Journal on Communications*, vol. 36, no. 11, pp. 118–126, 2015.

[10] M. A. Hamzah and S. H. Othman, "A review of support vector machine-based intrusion detection system for wireless sensor network with different kernel functions," *International Journal of Innovative Computing*, vol. 11, no. 1, pp. 59–67, 2021.

[11] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[12] M. M. Chowdhury, F. Hammond, G. Konowicz, C. Xin, H. Wu, and J. Li, "A few-shot deep learning approach for improved intrusion detection," in *2017 IEEE 8th annual ubiquitous computing, electronics and Mobile communication conference (UEMCON)*, pp. 456–462, New York, NY, USA, 2017.

[13] L. Wang, J. Li, U. A. Bhatti, and Y. Liu, "Anomaly detection in wireless sensor networks based on KNN," in *International Conference on Artificial Intelligence and Security*, pp. 632–643, Cham, 2019.

[14] L. Tao and Z. Sun, "KIPSO spoofing attack detection model in wireless sensor networks," *Journal of Transduction Technology*, vol. 29, no. 7, pp. 1049–1055, 2016.

[15] O. Xiao-qin and W. Qiu-hua, "An intrusion detection scheme based on mini batch K-means and SVM in wireless sensor networks," *Software Guide*, vol. 19, no. 3, pp. 204–209, 2020.

[16] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, "A new intrusion detection system based on KNN classification algorithm in wireless sensor network," *Journal of Electrical and Computer Engineering*, vol. 2014, Article ID 240217, 8 pages, 2014.

[17] W. Jun, Z. Zhi-wei, and L. Jun-jie, "Detection and defense method for blackhole attacks in wireless sensor networks," *Computer Science*, vol. 46, no. 2, pp. 102–108, 2019.

[18] O. U. Maheswari and S. Jayasankari, "Secure communication in wireless sensor network using intrusion detection system for agriculture," *International Journal of Modern Agriculture*, vol. 10, no. 2, pp. 1829–1845, 2021.

[19] R. Lohiya and A. Thakkar, "Intrusion detection using deep neural network with antirectifier layer," in *Applied Soft Computing and Communication Networks*, pp. 89–105, Springer, Singapore, 2021.

[20] X. Gong and Y. Xiao, "A skin cancer detection interactive application based on CNN and NLP," *Journal of Physics: Conference Series*, vol. 2078, no. 1, article 012036, 2021.

[21] A. B. Abhale and S. S. Manivannan, *Deep Learning Algorithmic Approach for Operational Anomaly Based Intrusion Detection System in Wireless Sensor Networks*, 2021.

[22] N. Singh, D. Virmani, and X. Z. Gao, "A fuzzy logic-based method to avert intrusions in wireless sensor networks using WSN-DS dataset," *International Journal of Computational Intelligence and Applications*, vol. 19, no. 3, article 2050018, 2020.

[23] R. N. Asha, "Data mining based intrusion detection system for securing wireless sensor network," *Harbin Gongye Daxue Xuebao/Journal of Harbin Institute of Technology*, vol. 53, no. 10, pp. 22–32, 2021.

[24] W. Yang, S. Wang, and M. Johnstone, "A comparative study of ML-ELM and DNN for intrusion detection," in *2021 Australasian Computer Science Week Multiconference*, Dunedin New Zealand, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7176194, 11 pages
https://doi.org/10.1155/2022/7176194

Research Article

A CAN Bus Security Testbed Framework for Automotive Cyber-Physical Systems

Dongxian Shi,[1,2] **Liang Kou**,[1] **Chaobin Huo**,[3] **and Ting Wu**[1]

[1] *School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China*
[2] *College of Information Technology, Zhejiang Institute of Economics and Trade, Hangzhou 310018, China*
[3] *National Computer System Engineering Research of China, Beijing 100000, China*

Correspondence should be addressed to Liang Kou; kouliang@hdu.edu.cn

Received 25 May 2022; Revised 9 July 2022; Accepted 19 July 2022; Published 12 August 2022

Academic Editor: Chenglu Jin

The popularization of electronic devices and the enrichment of external interfaces have increased the attack surface of the automotive cyber-physical system (CPS). As a vital part of the CPS, the controller area network (CAN) is more vulnerable to security threats due to the lack of corresponding security protection mechanisms. This kind of security problem has also attracted extensive attention from academia and industry. Researchers have proposed numerous intrusion detection models for the in-vehicle CAN bus, solving some security problems to a certain extent. However, vehicle manufacturers seldom disclose the internal details of vehicle ECUs due to safety concerns. Thus, it is difficult for researchers to investigate the operation mechanism of the bus. Meanwhile, there is a risk of personal safety in completing attack experiments on real vehicles, which can also lead to the lack and diversification of in-vehicle network data, especially the data of attack behavior. Based on real vehicle data, an open, adaptable, and low-risk CAN bus security testbed framework in the automotive CPS is proposed in this study, aiming to enrich the operation data of the CAN bus and enhance the personal safety of researchers. Besides, the delay of the testbed sending and receiving periodic and aperiodic CAN messages is theoretically explored. The results demonstrate that the generated timestamp in the dataset is mainly associated with the timestamp of the real vehicle data and that the transmission and collection of time series data are completed by Algorithm 1 and Algorithm 2. In the evaluation of the security testbed, the stability of the security testbed is studied from the two indicators of delay and packet loss rate. The experiment reveals that the testbed has a small relative delay difference and a low packet loss rate. Moreover, the DTW algorithm is employed to calculate the distance between the real vehicle and the testbed, and the experimental results demonstrate that the testbed is featured with high similarity and simulation.

1. Introduction

With the rapid development of technologies including the Internet of Vehicles, new energy, and artificial intelligence, vehicle manufacturers have configured more external interfaces and electronic devices for consumers, providing a more comfortable driving and entertainment experience. Meanwhile, the security of the CPS has been more threatened [1]. For example, hackers obtain root privileges through Wi-Fi and tamper with the electronic control unit (ECU) firmware on the bus [2, 3]. Therefore, the issue of vehicle security has attracted extensive attention from academia and industry. As one of the core components of modern automotive electronics, the ECU exchanges information with other ECUs through the bus protocol. Most bus protocols are public, such as CAN which is currently the most widely used bus protocol [4, 5]. However, the vehicle manufacturers or parts manufacturers have not disclosed the details of the internal parameters of the ECU in order to protect intellectual property rights [3], which means that the specific operating conditions on the bus are invisible to vehicle users and that security researchers need to have strong reverse analysis capabilities, thus bringing great difficulties to analyze the internal working mechanism of the ECU. At the same time, the implementation of bus attack experiments in the running state of the vehicle will result in unpredictable consequences, and it will also bring personal safety risks to researchers. Some researchers have published datasets of real vehicles [6, 7] that contain some common attack behaviors, which is conducive to the study on CAN bus security. However, in the face of complex vehicle environment, there is still a lack of rich attack types and attack behavior data. As a result, it is of practical significance to establish an open, adaptable, high-accuracy, and low-risk in-vehicle bus security testbed [8], providing important support for the theoretical research and real-time detection of vehicle CPS security [9, 10].

At present, the existing vehicle security testbeds are mainly based on the CAN bus, and there are mainly the following problems: (1) The versatility is not very strong [9, 11], and the same testbed cannot be shared between different models. (2) The security testbed is mainly constructed by simulating the vehicle through software, which cannot accurately evaluate the physical characteristics of the CAN bus and the ECU including power consumption and current, lacking certain authenticity. (3) The developed hybrid testbed improves the original vehicle bus structure and environment, but it will influence the security experiment effect of real vehicles [12]. (4) It is difficult to restore the data correlation between ECUs [13]. For example, in the homemade prototype system, the operating rod is used as the throttle to simulate the driver's acceleration and deceleration, which cannot truly represent the data correlation between the throttle and other ECUs. These problems will affect the performance evaluation of CAN bus intrusion detection methods [10], especially the research on content awareness and aperiodic characteristics of the CAN bus.

In this paper, a CAN bus security testbed based on real vehicle data in the automotive CPS is proposed, and the delay between CAN messages sent from the ECU Operation Center and received by the collector is theoretically explored. In the specific implementation process of the platform, by running Algorithm 1, the time series data of the vehicle is sent from the ECU Operation Center to the ECU, and the six designed attacks are employed to simulate the attack on the CAN bus. By running Algorithm 2, the data is sent to the CAN bus. Finally, the entire bus is generated to run time series data. The stability of the security testbed is examined through the two indicators of delay and packet loss rate. The experimental results indicate that the platform has a small relative delay difference and a low packet loss rate. Furthermore, the similarity of the time series between the real vehicle and the testbed is evaluated by the similarity index. Meanwhile, the distance between the two is calculated by the DTW algorithm, revealing that the testbed possesses high simulation.

Contributions of this paper are as follows:

(1) A CAN bus security testbed framework derived from real vehicle data is proposed. The testbed designs and implements six common attack models, and completes the sending and collection of CAN messages by running two algorithms, thereby ensuring the high real-time performance of the platform and the accuracy of the generated data

(2) In this study, the delay of sending and receiving periodic and aperiodic messages on the platform is theoretically explored. The results show that the generated time series data is mainly related to the timestamp sent. At the same time, the relative delay difference and packet loss rate are employed to explore the stability of the overall and single-type CAN messages of the platform. In addition, the experimental results also verify the ability of the platform in this regard

(3) The DTW algorithm is used to compare the similarity of the time series between the real vehicle and the security testbed. The experimental results demonstrate that the two sequences have a short distance, confirming the high simulation performance of the platform

The remainder of this paper is organized as follows. The background material about the CAN bus and related work are presented in Section 2. The framework, attack scenario and time series data generation method of the CAN bus security testbed in the automotive CPS are illustrated in Section 3. In Section 4, our experiment environment, evaluation metrics and results are elaborated. Besides, conclusions are drawn in Section 5.

2. Background

2.1. Controller Area Network. The CAN is a field bus with high reliability, high performance, and low cost [10, 14]. It was originally used in the vehicle electronic control network to realize the exchange of information between vehicle ECUs and was later extended and widely used in the field of industrial control. As an important part of the entire in-vehicle network and the automotive CPS, the CAN bus is a peer-to-peer network [15]. Each node on the CAN bus can either receive messages or actively send messages. When multiple nodes send messages to the bus at the same time, the bus adopts an arbitration mechanism to avoid conflicts. The nodes will read the messages on the bus and compare the bits of the arbitration field with the messages sent by themselves one by one. If the dominant bit is 0, it will continue to obtain control on the bus, while if the invisible bit is 1, the arbitration will be lost, and it will change to the receiving state from the next bit until the bus is free to continue sending messages [16].

2.2. Related Work. In the research of automotive CPS security, the known vehicle security testbeds mainly adopt the CAN bus protocol. In 2013, HRL and GM developed a set of security testbeds that highly imitates real vehicles [11]. As the platform employs the same ECU as real products, the simulation accuracy is high. Nevertheless, each vehicle and driving environment is complex, and this platform is not suitable for other models. In 2014, Miller and Valasek customized a portable off-road vehicle to build a security

```
INPUT: listR real time series dataset, listA attack time series data to CAN Shields
OUTPUT: listT messages received and precessed by CAN Shields from ECU Operating Center
/*ECU Operating Center sends real messages to CAN Shields according to a specified dataset*/
function $S_{END}M_{SG}(list)$
    $list\ P \longleftarrow list$
    $i \longleftarrow 0$
while $i < len(listP)$ do
    $t \longleftarrow$ current time
        if $t == listp[i].ts$ then
            /*ECU Operating Center sends real messages to CAN Shields */
            $u_i \longleftarrow$ delay for CAN Shield to process the CAN message
            $listP[i].ts \longleftarrow listP[i].ts + u_i$
            $listT.$ add $(listP[I])$
            $i \longleftarrow i + 1$
        end if
    end while
    return $listT$
end function
/* If real public dataset is not null,ECU Operating Center starts a new thread to send CAN messages*/
if $len\ (listR) > 0$ then
        Thread $t1 \longleftarrow$ $Thread(S_{END}M_{SG}(listR))$
            $t1$.start( )
end if
/* If attack dataset is not null, ECU Operating Center starts a new thread to send CAN messaged*/
if $len(list\ A)>0$ then
        Thread t2$\longleftarrow$ Thread $(S_{END}M_{SG}(listA))$
            t2.start( )
end if
return $listT$
```

Algorithm 1: ECU Operating Center sends times series data to CAN Shields.

testbed [9]. Although the price of the platform is not high, it is difficult to upgrade it. Moreover, due to the connection with real vehicles, there are certain security risks when researchers conduct attack experiments. There are also some software which can be used to simulate the vehicle ECU and CAN bus, such as CANoe. However, because of vehicle hardware with the software simulation, many physical properties, including power and current, could not be accurately simulated and evaluated. Especially in terms of the complexity of the vehicle system, software also remains stable. In 2016, Tuohy et al. proposed a hybrid testbed for the simulation of in-vehicle automotive networks. The platform incorporates multiple in-vehicle networks and is used via Ethernet in order to assist in the testing and development of automotive video systems and novel Advanced Driver Assistance System (ADAS) algorithms [12]. Although the testbed retains the in-vehicle network, this method changes the structure and environment of the original in-vehicle network, which will influence the effect of security testing.

At the BLACK HAT EUROPE 2018 conference, the Toyota Information Technology Center team proposed an adaptable portable security testbed PASTA [9] which turns the vehicle into a mini car with real vehicle functions through proportional scaling, using 4 ECUs and 1 console module to conduct the attack test of the operating system, without the need to complete the attack test in the real complex environment, lowering the risk of completing the experiment on the real vehicle. In addition, researchers can flexibly customize their own security technology through this platform, and the ECU has programmable capabilities. Meanwhile, some internal operations of PASTA are realized in the following three ways: (1) Displaying the ECU status on the monitor; (2) moving the model car through physical operation; and (3) the software simulator reads CAN messages and visualizes the behavior of the vehicle to the computer. However, there still exists a certain gap between the platform and the complexity of the real vehicle, and the correlation of CAN data generated between ECUs also lacks authenticity and rationality. Moreover, the device is still immature, causing some inconvenience to users.

3. Methodology

In this section, firstly, the framework of the CAN bus security testbed in the automotive CPS is presented. Then, the attack scenarios and the method of time series data generation for the security testbed are introduced.

3.1. Framework Overview. Figure 1 shows the proposed CAN bus security testbed framework based on real vehicle data. The platform inputs the time series data of each ECU of the real vehicle into the ECU Operation Center, which simulates the start of the vehicle and sends the corresponding data to the relevant ECUs based on the time series

```
INPUT:   list messages processed by CAN Shield, type attack type, $t_1$attack start timestamp, $t_2$ attack end timestamp
OUTPUT:   list' messages collected from the CAN bus
/∗Get the delay caused by CAN bus conflict when a CAN Shield sends messages to the CAN bus∗/
function GetConlictDelay(msg)
      $c \longleftarrow 0$
      flag ⟵has any conlict on CAN bus when sending msg?
      if flag== TRUE then
         /∗Self-delayed because other messages with smalle CANIDs are sent to the CAN bus at the same tim ∗/
         $c \longleftarrow delay$
      end if
      return c
 end function
 listP ⟵ list
 list' ⟵ null / ∗ Store in-vehicle messages collected from CAN Bus ∗/
 $i, j \longleftarrow 0$
   flag ⟵ FALSE
   while $i < len(listP)$ do
      $t \longleftarrow listP[i].ts$
      if current time== t then
            if $t < t_1$ and $t < t_2$ then
                  if listP[i].source==attack and (type==masquerade or type ≠ suspend) then
                     flag⟵ TRUE
                  else
                     flag ⟵ FALSE
                  end if
         else
                  flag ⟵ TRUE
         end if
         if flag ==TRUE then
            /∗Update timestamp for the conflict delay∗/
            listP[i].ts ⟵GetConflictDelay
            list'.add(listP[i])
            /∗Consider network delay∗/
            $d_i \longleftarrow$ delay for network transmission
            list'[j].ts ⟵$+d_i$
            $j \longleftarrow j + 1$
         end if
         $i \longleftarrow i + 1$
   end if
 end while
 return list'
```

Algorithm 2: Generate in-vehicle dataset collected from the CAN bus.

relationship. However, the ECU sends the CAN messages to the CAN bus in line with the CAN protocol. Besides, there are many interfaces for data interaction with the outside world on the entire security testbed including OBD-II, Bluetooth, and Wi-Fi, which not only facilitates the data exchange between the vehicle and the outside world, but also increases the risk of the vehicle being attacked. Hackers can connect to telematics devices through wireless communication channel and subsequently invade the CAN bus. Therefore, the platform is designed with multiple attack models to study in-vehicle bus security. In addition, the collector module on the security testbed extracts the running data on the CAN bus to form the testbed dataset of the platform that is adopted for subsequent data analysis, attack detection and attack prediction.

The platform simulates an ECU through an Arduino board loaded with CAN Shield. The ECU has programmable capabilities and can recognize commands sent by the ECU Operation Center and perform related operations, for instance, sending CAN messages to the CAN bus and performing spoofing attacks.

3.2. Attack Scenarios. According to the current common vehicle CAN bus attacks [13, 17, 18], the platform implements 6 types of attacks, respectively, fuzzy attack, replay attack, spoofing attack, suspend attack, DoS attack, and masquerade attack. The specific description is presented below.

3.2.1. Fuzzy Attack. In order to learn the correspondence between ECU and CAN ID or the meaning of CAN message fields, the attacker randomly injects CAN ID and payload

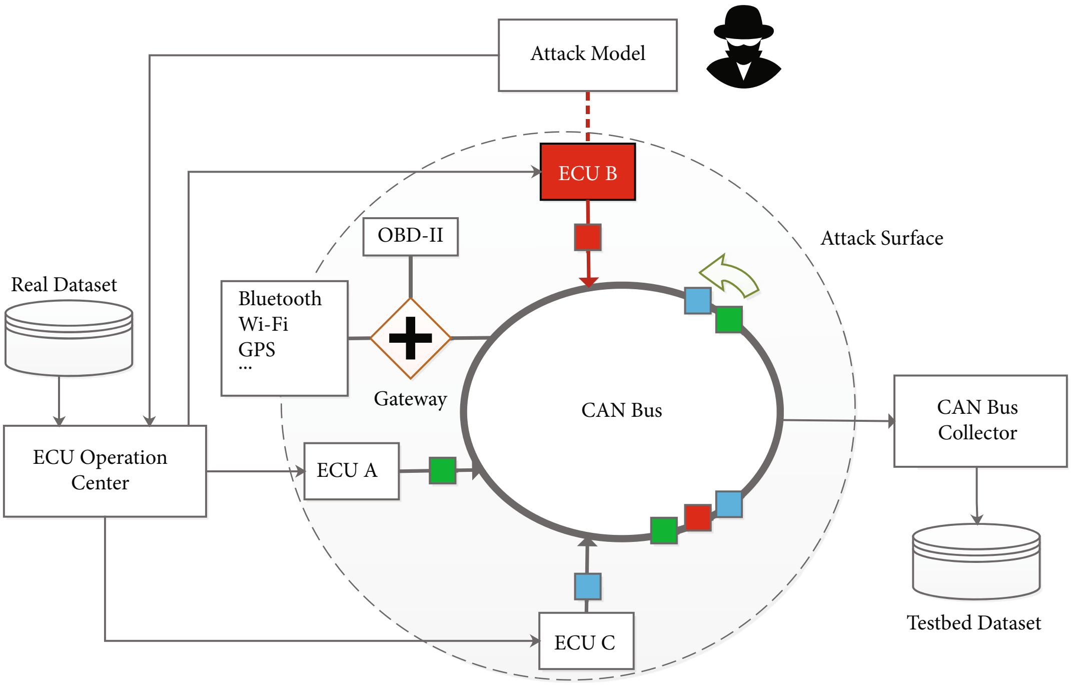


Figure 1: The framework of the CAN bus security testbed based on real vehicle datasets in the automotive CPS.

into the CAN bus and understands the CAN bus structure and ECU behavior patterns by observing the changes of the vehicle and ECU. According to the different injection content, fuzzy attack is categorized into fuzzy CAN ID attack and fuzzy payload attack. The fuzzy CAN ID attack is to observe the changes of the vehicle ECU by injecting an unknown CAN ID (generally below 0 × 700) externally when the correspondence between the CAN ID and the ECU is uncertain. However, fuzzy payload attack means that the attacker has determined the corresponding relationship between CAN ID and ECU, but does not know the specific field meaning of the payload. Therefore, he or she will familiarize with the function of an ECU by injecting the content of the modified payload of the ECU. In this testbed framework, the hacker simulates a fuzzy attack on the CAN bus by sending control commands and specified data to the ECU Operation Center.

3.2.2. Relay Attack. If the attacker is not sure about the function of the CAN ID and the semantics of the payload, they will listen to the CAN bus, capture the data fragment, and then directly inject the CAN data frame into the current time point. In this scenario, both the real ECU and the ECU impersonated by the attacker are sending data frames, while the attacker does not know the internal semantics of the specific data frame. Therefore, the flooding method is often adopted in order to force the CAN bus between the injected data frame and the original data frame. Furthermore, there are ECU data frames, while the former is selected for transmission. For example, the injected data frames are sent to the CAN bus 10-100 times faster than the original CAN data frames [13]. The hacker simulates a replay attack on the CAN bus by sending control commands and specified data to the ECU Operation Center of the testbed.

3.2.3. Spoof Attack. If the attacker determines the semantics of the payload, the modified data frame will be injected into the current CAN bus for attacking, which is called fabrication [19]. The attack will show that there will be two ECUs with the same CAN ID on the bus sending normal data frames. For example, the attacker injects 100 km/h vehicle speed data, causing the vehicle speed on the dashboard to change from 40 km/h to 100 km/h. As attackers often align the injection speed with the fake data frame, it is challenging for security detection to identify which is a normal data frame. In this scenario, hackers implement spoofing attacks by sending control commands and specified data to the Operation Center to simulate a spoofed ECU.

3.2.4. Suspension Attack. The attacker intrudes into the ECU in some way and suspends its work for a period of time, such as causing the steering wheel ECU to interrupt for 5 seconds to send data frames through malicious code injection. As ECU pauses cause changes in bus traffic and the interruption of such messages, this kind of attack is often easier to implement in intrusion detection, whether by analyzing the time interval of CAN messages [20], information entropy, machine learning [21], or other methods [22]. In this scenario, the attacker temporarily suspends the target ECU by sending control commands and specified data to the Operation Center, thus completing the ECU suspending attack.

3.2.5. DoS Attack. The attacker causes the CAN bus crash by flooding the bus with a large number of random data frames in a short amount of time [7]. Since the CAN bus determines the sending of frames through the CAN ID priority arbitration mechanism, the attacker can choose to send the CAN ID with high priority for attacking, aiming to initially check the attack effect. Sometimes, the highest priority data frame

0×000 is used to interfere with the sending and receiving of all ECUs on the CAN bus, thereby threatening the driving safety of vehicles. As DoS attacks easily lead to changes in traffic and CAN ID sequences, this kind of attack is often easier to implement in intrusion detection. In this framework, the attacker sends a large number of messages to the CAN bus by sending control commands and specified data in order to achieve the effect of DoS attack.

3.2.6. Masquerade Attack. The attacker is already familiar with the function of the CAN ID and the semantics of CAN payload, and he/she can perform various operations (such as suspend) on the relevant ECU in some way [18]. Subsequently, the attacker uses the ECU to send disguised CAN data. For example, the characteristics of the frame (periodic and aperiodic) include sending a well-designed attack data frame at an interval of 500 ms and simultaneously suspending the normal ECU. In addition, masqueraded ECU could also inject data frames at an interval of 500 ms. This kind of precision attack through semantics and camouflage is relatively difficult to detect. It is still very difficult to implement such an attack in a real vehicle. However, in this testbed framework, an attacker can simultaneously suspend a specified ECU and easily start a new ECU by sending a control command to perform a camouflage attack.

In addition to the above attacks, based on the openness and adaptability of the platform, researchers can add more attack scenarios as needed.

3.3. The Time Series Data Generation Method

3.3.1. Delay Analysis. The time series dataset generated by the security testbed in non-attack and attack environments needs to have low latency. Otherwise, it cannot better simulate the real vehicle environment. As a result, the delay of generating time series data by the security testbed is theoretically explored.

Supposing that there are two ECUs on the CAN bus, represented by A and R, respectively, and the ECU Operation Center is denoted by E, M_i as the i-th CAN message sent by A, and s_i as the real timestamp of the message. Then, E sends the message to A according to the timestamp s_i, and A sends the message to the CAN bus. In this way, the sending scenario of real vehicle CAN messages is simulated.

To calculate the timestamp when R receives the message sent by A, u_i denotes the delay of A receiving and processing the message sent from E, and c_i represents the delay of A sending due to bus arbitration failure, while d_i refers to the network delay of CAN bus transmission messages. Then, the timestamp when R receives the message is $T_{rx,i}$, as expressed below:

$$T_{rx,i} = s_i + u_i + c_i + d_i + n_i, \tag{1}$$

where n_i is the noise quantized by timestamp R [18], which is different from the ECU clock used in literature [10]. Since E distributes data through the A unified network clock, the clock shift of A is 0.

As $\Delta T_{rx,i}$ denotes the timestamp interval between the i-1th and i-th messages received by R, it is expressed as follows:

$$\Delta T_{rx,i} = \Delta s_i+\Delta u_i+\Delta c_i+\Delta d_i+\Delta n_i, \tag{2}$$

where ΔX represents the interval of X between the i-1th and i messages. Since the data length of each type of CAN messages of A is fixed, $\mathrm{E}[\Delta d_i] = 0$ [10]. However, when the zero-mean Gaussian noise distribution is satisfied, $\mathrm{E}[\Delta n_i] = 0$ [18]. In addition, as the lengths of each ECU and the ECU Operation Center are basically the same, as well as the control commands and programs programmed inside, the timestamp interval can be ignored when they receive the Operation Center E, whereas there may exist differences between ECUs due to different characteristics. For different processing times, $\mathrm{E}[\Delta u_i] \neq 0$. At the same time, c_i judges the time conflict on the CAN bus based on the timestamp s_i and the delay u_i. Although $\mathrm{E}[\Delta u_i] \neq 0$, Δu_i is almost negligible compared with Δs_i, indicating that the probability of arbitration collision is very small on the platform. Therefore, $\mathrm{E}[\Delta c_i] = 0$. Based on the above analysis, the expected value δ_{rx} of the timestamp interval between the two messages before and after R reception can be expressed as follows:

$$\begin{aligned}\delta_{rx} &= \mathrm{E}[\Delta T_{rx,i}] = \mathrm{E}[\Delta s_i+\Delta u_i+\Delta c_i+\Delta d_i+\Delta n_i] \\ &= \mathrm{E}[\Delta s_i] + \mathrm{E}[\Delta u_i+\Delta c_i+\Delta d_i+\Delta n_i] \approx \mathrm{E}[\Delta s_i] + \mathrm{E}[\Delta u_i].\end{aligned} \tag{3}$$

When A sends a message with a periodicity of T, $\mathrm{E}[\Delta s_i] = T$. Thus, δ_{rx} can be denoted as follows:

$$\delta_{rx} = \mathrm{E}[\Delta T_{rx,i}] \approx T + \mathrm{E}[\Delta u_i]. \tag{4}$$

Since the delay u_i of the ECU processing the message itself is very small, basically within 1 ms, $\mathrm{E}[\Delta u_i]$ will become smaller, and the minimum periodic message interval observed from the public dataset is not less than 10 ms. Hence, $\mathrm{E}[\Delta u_i]$ is considered negligible compared to the periodic time T.

When A sends an aperiodic message, δ_{rx} is equal to Expression (4). Although $\mathrm{E}[\Delta s_i]$ is not a constant, from the public dataset, the interval of aperiodic messages is much larger than 10 ms. Thus, compared with $\mathrm{E}[\Delta s_i]$, $\mathrm{E}[\Delta u_i]$ can be neglected.

Therefore, whether the time series data generated by the security testbed proposed in this paper is a periodic or aperiodic message, it is mainly associated with the timestamp s_i sent by the real dataset. When i is equal to 1, the timestamp of the first CAN message received by R is $s_1 + u_1 + c_1 + d_1 + n_1$, and the real dataset received is s_1. This is the difference between the two, that is, the entire time series dataset of the platform is offset backward from the real time series dataset. In fact, the value is so small that it can be almost negligible.

3.3.2. Time Series Data Generation Algorithm. In order to effectively lower the delay of CAN messages, the ECU Operation Center employs the multithread parallel transmission mode to ensure that each ECU CAN sends data to the CAN bus according to the timestamp of the dataset. Algorithm 1 presents the method that the ECU Operation Center sends time series data to the ECU.

When there is only public real data in the dataset, Algorithm 1 is employed to verify the effectiveness of the security testbed and real vehicle experiments in order to ensure that the delay rate and packet loss rate can be kept low. When there is attack data, or when the bus is attacked, the dataset generated by the platform can be applied as an important basis for investigating the security of the CAN bus, especially the changing state of the CAN ID sequence on the bus at the moment of the attack.

In this study, Algorithm 2 is adopted to obtain the time series data on the CAN bus. The ECU can perform processes according to the instructions and data of the Operation Center. When there is an attack instruction, it sends the corresponding attack data. Since arbitration conflict may occur when sending a CAN message, the delay of arbitration failure should be considered in the actual sending time. Furthermore, Algorithm 2 provides the process of the security testbed, finally generating the time series dataset.

4. Experiments

In this section, firstly, the experiment setup and the component of the CAN bus security testbed in the automotive CPS are presented. Then, three evaluation criteria of delay, packet loss rate, and similarity are introduced. Finally, the experimental results are explored in detail.

4.1. Experiment Setup. Up to now, it is known that there are few public CAN bus attack datasets and most of the public data concerning security research, including normal behaviors and common attack behaviors come from the literature [6, 7]. The real dataset currently studied comes from the literature [6], and it will be used as the basic data support for the security testbed.

The security testbed of this paper is composed of the ECU Operation Center, the CAN bus, the CAN node, the collector, and various connecting lines. Figure 2 shows the prototype of the platform, while Table 1 describes its main components and corresponding specifications. The CAN bus is simulated by a breadboard, and the CAN node consists of an Arduino UNO board and a SeeedStudio CAN Shield to simulate the sending and receiving of CAN messages. The ECU Operation Center is implemented by a computer program, which transmits data and control commands to the CAN node in line with the CAN ID classification and timestamp, while the collector collects the messages on the CAN bus in real time through the Arduino program.

4.2. Evaluation Metrics. The performance of the security testbed is evaluated from two dimensions of stability and effectiveness. For the stability of the security testbed, two commonly used network performance evaluation indicators, namely, delay and packet loss rate, are adopted. For the effectiveness of the time series data generated by the platform, the time series similarity is adopted for performing comparative analysis. The specific description is as follows.

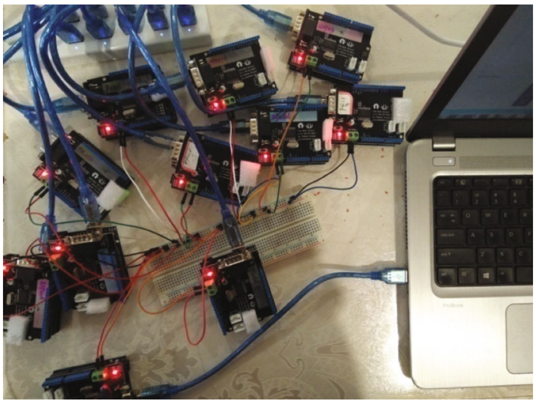


Figure 2: CAN bus security testbed prototype.

Table 1: Components of the CAN bus security testbed in the automotive CPS.

Component	Specification
	Windows Server 2012 R2
ECU operation center	Arduino Uno (ATmega328)
CAN node (ECU)	CAN-BUS shield V1.2
CAN bus	500Kbps
Collector	Arduino Uno (ATmega328)
Actual vehicle	CAN-BUS shield V1.2
	Toyota Camry 2010

4.2.1. Delay. Delay is an important indicator that must be considered in the study of platform performance [23]. It refers to the security testbed causing the message to be delayed in time during the process of CAN message transmission, due to various reasons containing sending processing, network transmission, bus arbitration, ECU processing, and receiving processing. Since the real dataset only has the time to receive the CAN message on the bus, and does not know the sending time of the message, this study takes the time interval from the first message to the last message received by the collector within a certain period of time as the delay. In the current work, the delay is divided into the overall delay and the delay of a single type of CAN message.

The overall delay is the time interval between the collector receiving the first message and the last message within a certain period of time. The specific definition is as follows.

Definition 1. During time T, if the timestamp of the collector receiving the first message is T_s and that of receiving the last message is T_e, the overall delay ΔT can be defined as

$$\Delta T = T_e - T_s. \qquad (5)$$

The delay of a single type of CAN message is the time interval from the first to the last message of the type received by the collector within a certain period of time. The specific definition is as follows.

Definition 2. During time T, if the collector receives the first message with message ID i, the timestamp is T_s^i, and the timestamp of the last message received is T_e^i. Next, the delay of ID message ΔT^i is defined as

$$\Delta T^i = T_e^i - T_s^i. \quad (6)$$

To better evaluate the performance of the platform, [24] is taken as the reference for using the relative delay difference index. The relative delay difference in this paper refers to the distance between the message delay generated by the platform and the real vehicle message delay in a unit time interval relative to the message delay of the real vehicle. The specific definition is depicted below.

Definition 3. If the delay of real vehicle messages is recorded as ΔT_1 and that of platform messages is considered ΔT_2, the relative delay difference $\Delta \tilde{T}$ is defined as

$$\Delta \tilde{T} = \frac{\Delta T_2 - \Delta T_1}{\Delta T_1}. \quad (7)$$

4.2.2. Packet Loss Rate. During the process of sending CAN messages from the ECU Operation Center to the collector obtaining messages from the CAN bus, the message loss is called packet loss [25, 26] due to various reasons including transmission processing, network transmission delay, bus arbitration waiting, ECU processing, and receiving processing. The end-to-end packet loss rate refers to the percentage of the total number of messages lost in the process of transmitting messages from the sender to the receiver within the specified time interval to the total number of sent messages. In this paper, it is called the packet loss rate, and the metric is defined as follows.

Definition 4. For the sending and receiving ends of CAN messages, within the time T, if the total number of messages sent by the sender is recorded as N_s, and that of messages not received by the receiver is denoted as N_s, then the packet loss rate R is defined as

$$PLR = \frac{N_f}{N_s}. \quad (8)$$

When the platform simulates a non-attack scenario, T in this paper represents the running time of a real dataset, and N_s denotes the total number of messages in the real dataset. When the platform simulates an attack scenario, T and N_s need to be added to the attack time and the number of attack messages, respectively.

4.2.3. Time Series Similarity. To evaluate the effectiveness of the time series data generated by the platform, the similarity between the time series data generated by the platform and the real data should be compared. Since packet loss is inevitable in a strong real-time environment, the time series lengths of the two are often different. Therefore, the similarity cannot be well reflected by calculating the Euclidean distance. Moreover, the current commonly used method aims to use dynamic time wrapping (DWT) [27, 28] to solve the existing problem. Apart from that, in this paper, the algorithm in [28] is adopted as the time series similarity calculation standard.

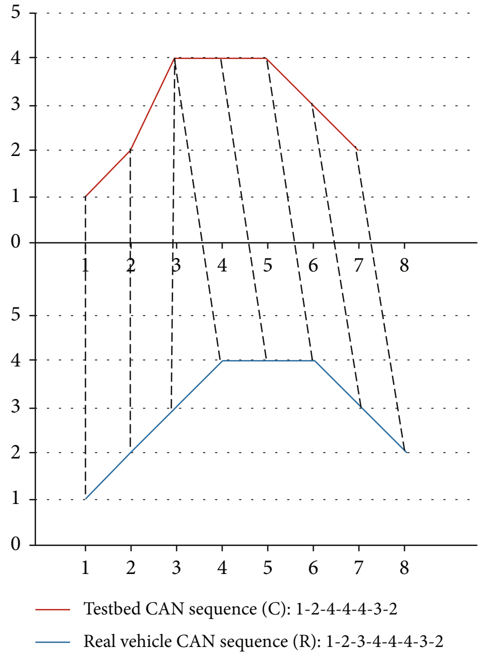


Figure 3: DTW similarity calculation example of two time series.

Supposing that there are two sequences R and C, the former is a certain type of CAN sequence in the real dataset, and the latter is the same type of CAN sequence generated by the platform, represented by $R_m = \{r_1, r_2, \cdots, r_{m-1}, r_m\}$ and $C_n = \{c_1, c_2, \cdots, c_{n-1}, c_n\}$, respectively, where m is the sequence length of R and n refers to the sequence length of C. In terms of the steps of the DWT algorithm, it is first to calculate the distance matrix of each element of the two sequences R and C and then to find a path with the shortest distance sum from the upper left corner to the lower right corner of the matrix. In addition, it indicates that the shorter the path is, the higher the similarity of the two sequences is while the lower the similarity is.

Through time warping, the point at a certain time of sequence C corresponds to the point at multiple continuous moments of sequence R in order to achieve the goal of the minimum distance sum. Figure 3 displays an example of calculating the similarity of two time series R and C. By adopting the DTW method, the sum of the shortest distance of the two series can be obtained as 1.

4.3. Result Analysis

4.3.1. Delay Analysis. In this paper, the security testbed is tested 50 times, and the results are averaged. Since there are many CAN messages on the CAN bus, Table 2 lists the delay of common CAN messages in the platform and real datasets.

As shown in the table, the total relative delay difference of the platform is approximately 0.8%, suggesting that all

Table 2: Delay results in the testbed and the real vehicle.

CAN ID	Description	Rate	Real vehicle delay (ms)	Testbed delay (ms)	Relative delay difference (%)
All	All CAN messages	—	74352	74970	0.80
0B0	Speed of wheels 1 and 2	0.01	74352	74509	0.21
0B2	Speed of wheels 3 and 4	0.01	74353	74943	0.80
610	Vehicle speed	0.5	73481	73555	0.10
025	Steering angle	0.01	74352	74929	0.78
224	Brake pedal	0.32	74322	74440	0.18
2C1	Throttle	0.99	74324	74970	0.87
2C4	Engine speed	0.02	74333	74935	0.81
398	Fuel	Aperiodic	73081	73108	0.04
3B4	PRND	Aperiodic	72921	72990	0.09

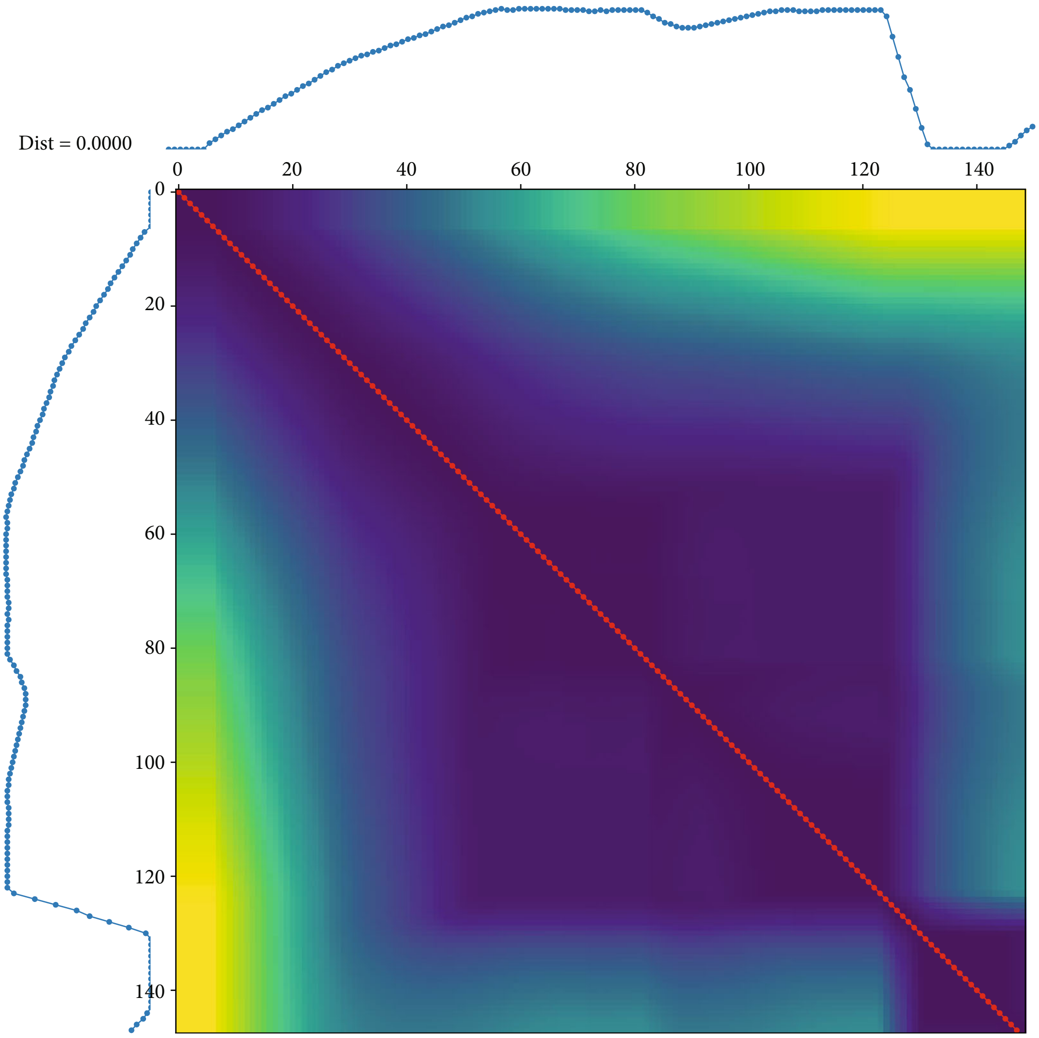


Figure 4: DTW similarity distance of CAN ID 610 between the real vehicle and the testbed.

messages are sent 0.8% later than the real vehicle. In addition, it can also be observed from a single type of CAN message that the relative delay difference of periodic and high-frequency messages is generally larger. For example, messages with CAN ID 0B2, 2C4, and 025 have a relative delay difference of periodic and low-frequency messages. In general, for smaller messages including the message with CAN ID 3B4, the relative delay difference of aperiodic messages will be smaller, such as the message with CAN ID 398. However, it is found that some messages are different, such as CAN ID 0B0 and 610, whose sending frequency is high, and relative delay difference is low. From the experimental process, it can be known that the situation is also related to the message type. For example, 0B2 and 0B0 belong to the wheels of the vehicle and are sent at the same frequency and almost simultaneously. Therefore, the 0B0 message with higher priority may block the sending of the 0B2 message, causing more long delay.

4.3.2. Similarity Analysis. In this paper, the similarity of the two time series between the real vehicle and the testbed is calculated according to the above DTW algorithm. In the following, the messages with lower frequency (CAN ID 610) and higher messages (CAN ID 0B0) as representatives are investigated.

Figure 4 shows all regular paths and the shortest paths with CAN ID 610. Obviously, the shortest distance of the two time series is 0, and the path becomes a diagonal line, indicating that the two are exactly the same. From the experimental results, it is also demonstrated that the two sequences are basically the same and the similarity is almost the same. Therefore, the packet loss rate is extremely low.

5. Conclusion

With the intelligent, networked, and electronic modern automobiles, the environment of the automotive CPS has become more complex, and the in-vehicle network, especially the CAN, has been threatened. Numerous scholars investigate the security issues of in-vehicle networks through software simulation and real vehicle experiments. In this paper, a CAN bus security testbed based on real vehicle data is proposed in order to help researchers build an open, adaptable, and low-risk infrastructure. To confirm the performance of the security testbed, firstly, the delay of CAN message sending and receiving is theoretically explored, demonstrating that the delay is mainly associated with the timestamp of real vehicle message sending from two aspects of periodicity and aperiodicity. Secondly, Algorithm 1 and Algorithm 2 are designed to complete the sending and receiving of platform messages and realize the simulation of six common attack behaviors. Finally, the security testbed is discussed in detail through relative delay difference, packet loss rate, and similarity. The experimental results demonstrate that the platform is featured by high stability and simulation.

In the next step, we plan to further study the latency of platform messaging, especially to improve the similarity between frequently sent messages and real vehicle time series data. Meanwhile, the attack model will be further enhanced, and the effectiveness of the attack model in this security testbed will be further confirmed.

Data Availability

The data used to support the findings of the study are available at the Colorado State University (https://www.engr.colostate.edu/~jdaily/tucrrc/ToyotaCAN.html).

Conflicts of Interest

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

Acknowledgments

This work was supported by the Key Research and Development Program of Zhejiang Province (No.2022C01125) and grants from "the Fundamental Research Funds for the Provincial Universities", Zhejiang Institute of Economics and Trade (No. 19SBYB06).

References

[1] W. Wu, R. Li, G. Xie et al., "A survey of intrusion detection for in-vehicle networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 919–933, 2020.

[2] D. Shi, T. Wu, M. Xu, and L. Kou, "Intrusion detecting system based on temporal convolutional network for in-vehicle CAN networks," *Mobile Information Systems*, vol. 2021, 13 pages, 2021.

[3] G. Dupont, J. D. Hartog, S. Etalle, and A. Lekidis, "Evaluation framework for network intrusion detection systems for in-vehicle CAN," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, Graz, Austria, 2019.

[4] H. Qin, M. Yan, and H. Ji, "Application of controller area network (CAN) bus anomaly detection based on time series prediction," *Vehicular Communications*, vol. 27, article 100291, 2021.

[5] L. B. Othmane, H. Weffers, M. M. Mohamad, and M. Wolf, "A survey of security and privacy in connected vehicles," in *Wireless Sensor and Mobile Ad-Hoc Networks*, pp. 217–247, Springer, 2015.

[6] R. Ruth, W. Bartlett, and Y. J. Dail, "Accuracy of event data in the 2010 and 2011 Toyota camry during steady state and braking conditions," *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, vol. 5, no. 1, pp. 358–372, 2012.

[7] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS:a novel intrusion detection system for in-vehicle network by using remote frame," in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, pp. 57–5709, Calgary, AB, Canada, 2017.

[8] Z. A. Biron, S. Dey, and P. Pisu, "Real-time detection and estimation of denial of service attack in connected vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3893–3902, 2018.

[9] T. Toyama, T. Yoshida, H. Oguma, and T. Matsumoto, *PASTA: portable automotive security testbed with adaptability*, Black Hat Europe, 2018.

[10] M. Markovitz and A. Wool, "Field classification, modeling and anomaly detection in unknown CAN bus networks," *Vehicular Communications*, vol. 9, pp. 43–52, 2017.

[11] S. Nie, L. Liu, and Y. Du, *Free-fall: hacking tesla from wireless to can bus*, Black Hat USA, 2017.

[12] S. Tuohy, M. Glavin, E. Jones, C. Hughes, and L. Kilmartin, "Hybrid testbed for simulating in-vehicle automotive networks," *Simulation Modelling Practice and Theory*, vol. 66, pp. 193–211, 2016.

[13] K. Fischer, "HACMS," *ACM SIGAda Ada Letters*, vol. 32, no. 3, pp. 51-52, 2012.

[14] C. Miller and C. Valasek, *Car hacking: for poories*, SyScan, 2014.

[15] J. Ning, J. Wang, J. Liu, and N. Karto, "Attacker identification and intrusion detection for in-vehicle networks," *IEEE Communications Letters*, vol. 23, no. 11, pp. 1927–1930, 2019.

[16] Q. Luo and J. Liu, "Wireless telematics systems in emerging intelligent and connected vehicles: threats and solutions," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 113–119, 2018.

[17] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 5–9, Baden-Baden, Germany, 2011.

[18] K. T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *Proc. of the 25th USENIX Security Symposium*, pp. 911–927, Austin, TX, 2016.

[19] K. Iehira, H. Inoue, and K. Ishida, "Spoofing attack using bus-off attacks against a specific ECU of the CAN bus," in *15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pp. 1–4, Las Vegas, NV, USA, 2018.

[20] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network," in *2016 International Conference on Information Networking (ICOIN)*, pp. 63–68, Kota Kinabalu, Malaysia, 2016.

[21] H. M. Song, J. Y. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, article 100198, 2020.

[22] S. Zander and S. J. Murdoch, "An improved clock-skew measurement technique for revealing hidden services," in *Proceedings of the 17th conference on Security symposium*, pp. 211–225, Berkeley,CA,USA, 2008.

[23] N. Wang, *Research on Stability Analysis and Control Method of Networked Control System with Time-Delay*, Shandong University of Technology, Jinan, China, 2021.

[24] K. Wu, *Research on Fast Algorithm and Link Delay Difference in Radiation Two-Step Method*, Hunan University, Changsha, China, 2020.

[25] T. Mi, *The Research on Packet Loss Rate Based on the Measured Data*, Southeast University, Nanjing, 2015.

[26] H. Lan, *Research on Key Technology for Packet Loss Estimation Based on Passive Measurement*, Southeast University, Nanjing, China, 2020.

[27] J. Serra and J. L. Arcos, "An empirical evaluation of similarity measures for time series classification," *Knowledge-Based Systems*, vol. 67, pp. 305–314, 2014.

[28] D. F. Silva and G. E. Batista, "Speeding up all-pairwise dynamic time warping matrix calculation," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, pp. 837–845, Miami, Florida, USA, 2016.