Cyber Security and Applications 1 (2023) 100004

Contents lists available at ScienceDirect

Cyber Security and Applications

journal homepage: http://www.keaipublishing.com/en/journals/cyber-security-and-applications/

Secure distributed data integrity auditing with high efficiency in 5G-enabled software-defined edge computing

Dengzhi Liu[a,b,*], Zhimin Li[a], Dongbao Jia[a]

[a] School of Computer Engineering, Jiangsu Ocean University, Lianyungang 222005, China
[b] Jiangsu Institute of Marine Resources Development, Lianyungang 222005, China

ARTICLE INFO

Keywords:
Edge computing
5G and software definition
Distributed data integrity auditing
Certificateless cryptography
Key exposure resistance
Privacy-preserving

ABSTRACT

In edge computing, the idle resources of the devices in the network can be virtualized into a platform that provides clients with storage resource and computing capability. Note that the service response of edge computing is faster than that of cloud computing. The service provision speed and the distributed resources utilization rate of edge computing will be further improved when integrated with 5 G and software definition paradigm in the design of the network system. However, the issues of data storage security and edge devices' trustworthiness seriously restrict the development of edge computing. To enhance the security of the data storage in edge computing, a secure distributed data integrity auditing is proposed. The proposed auditing scheme in this paper can be used to guarantee the correctness and the completeness of the stored data in 5G-enabled software-defined edge computing. The auditing results of the distributed data in the proposed scheme can be used as an important basis for evaluating the trustworthiness of the edge devices. Due to the utilization of certificateless cryptography in the design of the proposed scheme, the computational cost of the terminal side can be highly reduced. Security analysis of the proposed scheme demonstrates that the properties of key exposure resistance and privacy-preserving are provided in data auditing. Simulation results of the time cost of the server side and the terminal side show that the proposed scheme is highly efficient compared to previous schemes.

1. Introduction

Edge computing is a paradigm that can improve the quality of the service and the rate of resource utilization of the network system. Edge computing is developed from distributed computing that can integrate the hardware resources of the devices nearby the terminals to provide high quality services to users with low latency [1,2]. Similar to cloud computing, edge computing provides services to users via the network. Compared with the high centralization of cloud computing, edge computing is decentralized. The services provided by devices in edge computing do not need to be managed and provided through a centralized server. That is to say, the services provision of edge computing is directly client oriented that results in the service quality is more high, reliable and secure compared to that in cloud computing [3]. The obvious advantages of edge computing determines that it can be well used in many network-based scenarios, such as internet of things [4], wireless sensor networks [5], smart cities [6], etc.

Edge computing is a promising technique, especially for the developing requirements of the intellectualized and the network control in all walks of life. It is worth noting that the feature of heterogeneity of edge devices will lead to insufficient utilization of the resources. Moreover, the edge devices in edge computing are highly dispersed and only serve peripheral devices, which results in low resource utilization of edge resources. Hence, how to fully schedule the resources of edge devices has become a hot research topic in the study of edge computing. To reduce the execution latency and the energy consumption of edge devices, Miao *et al.* proposed an optimize task offloading scheduling and transmit power allocation scheme for mobile edge computing. Note that the proposed scheme in Miao *et al.*'s research is constructed based on alternating minimization [7]. To solve the stochastic task generation and dynamic network conditions in mobile edge computing, Liu *et al.* proposed a computation offloading and resource scheduling based on reinforcement learning for mobile computing [8]. Moreover, Liu *et al.*'s scheme can estimate the cumulative latency and energy rewards. However, the related schemes of scheduling in edge computing have not been studied from the mode of resource management, which limits the development of edge computing.

To improve the flexibility of edge resources management and realize the intelligent collaboration of edge devices, in 2020, Hu *et al.* proposed the paradigm of software-defined edge computing (SDEC) inspired by the concept of software-definition [9]. The proposed framework in Hu *et al.*'s scheme can well optimize the management of edge devices and

* Corresponding author.
E-mail address: liudz@jou.edu.cn (D. Liu).

https://doi.org/10.1016/j.csa.2022.100004
Received 27 June 2022; Received in revised form 15 July 2022; Accepted 20 July 2022
Available online 23 July 2022
2772-9184/© 2022 The Authors. Published by Elsevier B.V. on behalf of KeAi Communications Co., Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

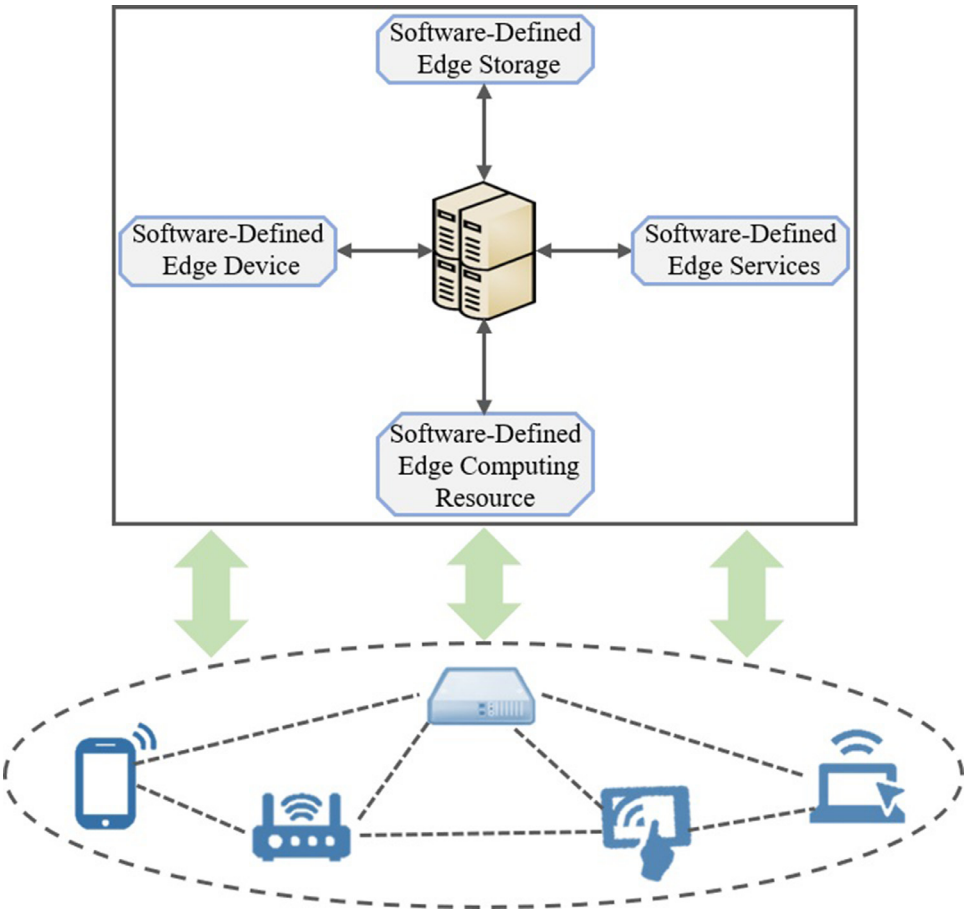

Fig. 1. The paradigm of the SDEC with IoT.

reasonably schedule the resources in edge computing. The framework of the SDEC combined with IoT (Internet of Things) is shown as Fig. 1. Note that there are four components in SDEC, including software-defined edge device, software-defined edge storage, software-defined edge computing resource and software-defined edge service. It worth noting that all the components in the system are managed and controlled by a control center. The main goal of the design of SDEC is to realize the automatic interaction and collaboration between hardware and software that can make full use of the idle resources in edge computing.

The SDEC can be viewed as a promising framework to construct a system of edge computing in the further, especially combined with the emerging network technologies, such as 5G-enabled SDEC. The technology of 5 G can provide communications with high quality and low latency [10,11] that makes the service provision in SDEC more efficient. However, with the increase of the amount of edge devices, a new problem is emerged that is how to guarantee the storage security of the terminals' stored data in edge devices. The traditional related distributed data protection methods [12–14] are not suitable for SDEC environment because the local resources of terminals are limited and the edge devices are decentralized. Hence, it is necessary to construct a data storage security protection scheme with high efficiency for 5G-enabled SDEC.

1.1. Related work

In the security assurance of edge computing, many researches of data storage have been studied. In 2018, Fu *et al.* proposed a secure data storage and searching mechanism that can execute data processing, secure data storage, efficient data retrieval and dynamic data collection [15]. Note that ID-AVL tree and hash value are used to construct the data structure to process the raw data and store the time-sensitive data in edge server in Fu *et al.*'s scheme. To improve the security of caching data in data retrieval, a cache data integrity auditing scheme is proposed in Li *et al.*'s scheme [16]. In Li *et al.*'s scheme, the variable Merkle Hash Tree is used to generate the storage proof to improve the accuracy of the cache data. To realize the stored private data integrity checking in edge computing, Wang *et al.* proposed a data integrity verification scheme based on ZSS signature to ensure the privacy protection and public auditing for the data stored in edge servers [17]. However, the consideration of data security in the previous schemes in data auditing and storage is not enough.

To improve the storage security, many auditing schemes have been proposed in recent years. The first two studies of data integrity auditing are PDP (Provable Data Possession) [18] and POR (Proof of Retrievability) [19]. Note that PDP and POR cannot support the public auditing which improves the computational cost of proof checking at the local side. In 2009, Wang *et al.* first proposed a public auditing with data dynamics to reduce the storage auditing computational cost of client side [20]. To realize the property of privacy preserving, Wang *et al.* proposed a public auditing based on random masking in 2010 [21]. The extended works [22,23] of the above public auditing are published in 2011 and 2013, respectively. In [22], the Merkle Hash Tree is firstly used to construct the data structure in storage auditing to realize secure data dynamics. In [23], the techniques of homomorphic linear authenticator and random masking are used to construct public auditing with privacy-preserving for cloud storage. In 2013, Yang *et al.* proposed a public auditing based on the property of bilinear pairing to support data dynamics and batch auditing [24]. Note that Yang *et al.*'s scheme can be executed in storage auditing with high security and efficiency. To realize efficient data dynamics, a dynamic hash table structure is designed for the public data auditing in Tian *et al.*'s scheme [25]. To deal with the big data auditing, Sookhak *et al.* proposed a structure of Divide and Conquer Table (D&CT) to support the dynamic operations for

the big data in 2018 [26]. It is worth noting that the proposed auditing in Sookhak *et al.*'s scheme can be executed with low communication and computational cost. However, the related schemes cannot be used directly in edge computing because the edge devices in the environment of edge computing cannot complete the complexity computation with limited resources. Hence, a secure and efficient distributed data integrity auditing needs to be studied for 5G-enabled software-defined edge computing.

1.2. Contributions

This paper proposed a secure distributed data integrity auditing with high efficiency in 5G-enabled software-defined edge computing. The main contributions of this paper are listed as follows.

(a) The paradigm of the certificateless cryptography is used to design the proposed scheme, which highly eliminates the computational cost of the terminal side in the generation of security keys and tags.
(b) The ECS in the system can audit the data blocks without using any keys and parameters of terminals expect the public key. In other words, the proposed scheme can provide the property of key exposure resistance in the storage auditing.
(c) In the data integrity auditing, the ECS cannot obtain the complete data file of terminals according to the storage proof elements received from edge computing. In other words, the proposed scheme supports privacy-preserving for the audited data.

1.3. Organization of this paper

The reminder of the paper is organized as follows. Section 2 describes the preliminaries that are used in the design of the proposed scheme. Section 3 introduces the system model and the design goals. Section 4 presents the details of the proposed scheme in this paper. Section 5 provides the evaluation that includes security analysis and performance analysis. The conclusion is given in Section 6.

2. Preliminaries

The preliminaries that are used in the construction of the proposed scheme are introduced briefly in this section. First, the description of the bilinear pairing is presented. Then, the main process of certificateless cryptography is provided.

2.1. Bilinear pairing

The technique of bilinear pairing was first proposed in identity-based encryption by Boneh et al. in 2001 [27]. In the bilinear pairing, three multiplicative groups of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ with prime order q are defined. The generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ are denoted as $\mathcal{G}_1$ and $\mathcal{G}_2$. The property of bilinear pairing can be denoted as $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Suppose that two elements of $\mathcal{P}$ and $\mathcal{Q}$ are randomly chosen from group $\mathbb{G}_1$. For any $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{G}_1$, we can get that $e(\mathcal{P}_1 \cdot \mathcal{P}_2, \mathcal{Q}) = e(\mathcal{P}_1, \mathcal{Q}) \cdot e(\mathcal{P}_2, \mathcal{Q})$. Here, assume that there are two integers of $a$and $b$that are randomly chosen from $\mathbb{Z}_q^*$. The properties of the bilinear pairing are shown as follows.

(a) Bilinear: $e(\mathcal{P}^a, \mathcal{Q}^b) = e(\mathcal{P}, \mathcal{Q})^{ab}$.
(b) Non-degenerate: $e(\mathcal{G}_1, \mathcal{G}_2) \neq 1$.
(c) Computable: $e(\mathcal{P}, \mathcal{Q})$ can be efficiently computed by an existing algorithm.

2.2. Certificateless cryptography

The technique of the certificateless cryptography is proposed to solve the problem of key escrow in identity-based encryption [28]. To avoid the leakage of the private key of users, the KGC should generate a partial private key according to the user's identity information. Then, the user can create his/her own private key according to the received partial key and the generated secret value. In general, certificateless cryptography consists of five phases, including $Setup$, $PKeyGen$(Partial Key Generation), $SValGen$(Secret Value Generation), $SKeyGen$(Secret Key Generation) and $PKeyGen$(Public Key Generation). The main process of the five phases is shown as follows:

(a) $Setup \rightarrow (p, M_{SK})$: The operations in this phase are executed by the KGC. The input of this phase is a security parameter θ. Then, the KGC will generate the system parameter p and the secret master key M_{SK}. Note that the system parameter p will be sent to the user.
(b) $PKeyGen \rightarrow P_{sk}$: According to the user's identity ID, the KGC will compute the partial key P_{sk} for the user using the system parameter p and the secret master key M_{SK}. Finally, the partial key is sent to the user.
(c) $SValGen \rightarrow S_v$: In this phase, the user will compute a secret value $S_v$according to his/her own identity ID and the received system parameter p from the KGC.
(d) $SKeyGen \rightarrow SK$: This phase is executed by the user. It takes as input the system parameter p, the received partial key P_{sk} and the secret value S_v. The output of this phase is the user's secret key SK.
(e) $PKeyGen \rightarrow PK$: This phase can output the user's public key PK by using the system parameter p and the secret value S_v.

3. System model and design goals

This section first presents the system model of the proposed scheme. Then, the design goals of the proposed scheme in this paper are described.

3.1. System model

The system model of the proposed scheme consists of three layers. The bottom layer contains the terminals, which are the users to enjoy the services provided by edge computing. The middle layer of the system is edge computing that the belonged edge devices are connected by the wireless and wired networks. In this paper, the communication technology between different devices and layers is 5 G that can provide high bandwidth services to the connected devices. Moreover, the technique of 5 G can ensure the high quality in communications with low latency. To reduce the impact of heterogeneous devices on service provision in edge computing, the paradigm of software-defined networks is used to manage and control the edge devices in edge computing. That is to say, the edge computing in the system is a Software-Defined Edge Computing (SDEC). In addition, an Edge Computing Server (ECS) is introduced in the system to manage the services provided by the SDEC. The system framework of the proposed 5G-enabled SDEC is shown as Fig. 2. The following description is the details of the system model and the corresponding security assumptions.

(a) **Layer 1. Terminals:** The terminals in the system are the devices in IoT (Internet of Things) networks that enjoy the services from the upper layer of edge computing via 5 G in the system. It is worth noting that the terminals in this layer are lightweight devices with limited computational capability and storage space. Hence, the terminals need to delegate the tasks of storage and computing to the edge devices. On the aspect of security, the terminals are not trusted. The terminals are easy to be attacked by adversaries for financial interest in the system. To ensure the security of the gathered data, the terminals should encrypt the data before outsourcing it to edge computing for storage and service requests. In addition, the terminals can delegate the ECS to audit the stored data in edge computing to enhance the storage security.
(b) **Layer 2. Services Provider:** The service provider in this layer are the edge devices in edge computing. The edge devices are the idle devices nearby the IoT in the network. Here, the communication network is the 5 G that can improve the service response speed and

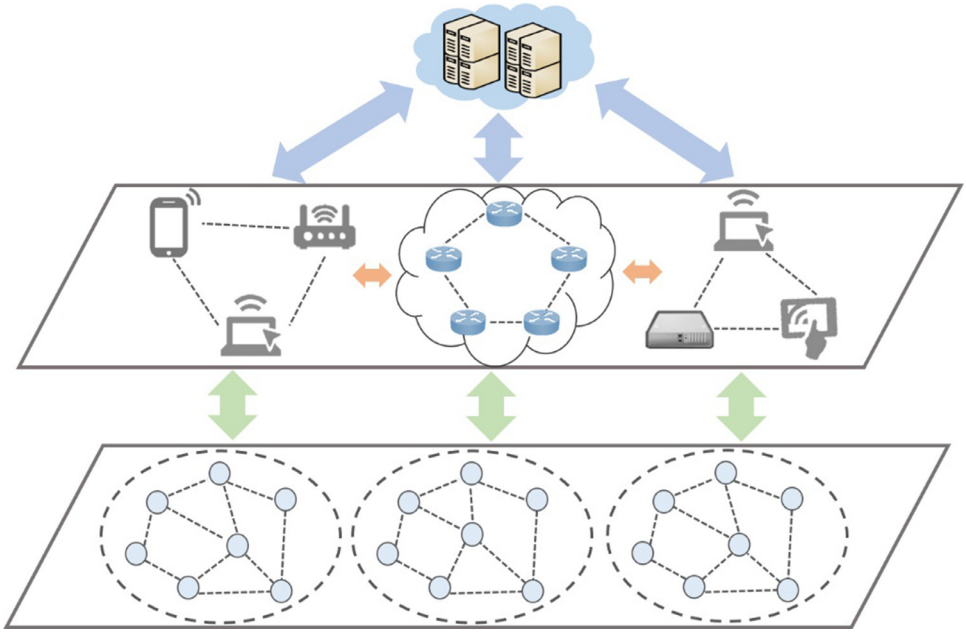

Fig. 2. The system framework of the 5G-enabled SDEC.

information transmission quality. That is to say, the devices in edge computing are the connected devices in the cellular network provided by the same base station and adjacent base stations in 5 G. Compared to cloud computing, the resource center in edge computing is closer to terminals. The distributed devices can provide high quality services to terminals with low latency. To fully schedule the resources of edge devices, the concept of software-definition is used to manage the storage resources, the computing capability and the edge services. Hence, the services and resources provision in the proposed 5G-enabled SDEC are more flexible compared to traditional edge computing. This layer is responsible for storing the gathered data and executing the computational tasks outsourced by the IoT devices. The edge devices are honest but curious to the stored data. Moreover, it is possible that the edge devices will delete the less often used data and try to forge the computational result when the corresponding data has been destroyed. Hence, it is necessary to check the trustworthiness of edge devices and audit the storage integrity of the data periodically in edge computing.

(c) **Layer 3. Server/Controller:** The server is the ESC that can be used to manage the resource and services of edge computing. Moreover, the ESC plays the role of the controller in SDEC. That is to say, the ESC is the management and control center that schedules the resources of edge computing and achieves the collaboration among devices, storage, computing capability and services in edge computing. Moreover, the ESC is the KGC of the system and can be seen as the auditor of the distributed data integrity auditing in edge computing. Different from the entities in layer 1 and layer2, the ESC is a fully-trusted entity that can successfully generate and compute the parameters and keys according to the algorithms in the proposed scheme.

3.2. Design goals

The design goals that the proposed scheme should be achieved in the design are listed as follows.

(a) **Distributed Data Integrity Auditing.** To enhance the security of the system, the proposed scheme should ensure that the distributed data stored in edge devices can be audited by the fully-trusted ESC on behalf of the terminals.

(b) **High Efficiency.** Due to the constrained computing capability of terminals, the proposed scheme should reduce the computational cost of local side in data processing for the storage and auditing.

(c) **Key Exposure Resistance.** In the data storage and auditing, the proposed scheme should avoid the key leakage. In other words, the ESC audits the data using the public keys of terminals and its own security parameters.

(d) **Privacy-Preserving.** The proposed scheme should support that the ESC cannot obtain and recover the original data file from the auditing elements. In other words, the proposed scheme should provide the property of privacy-preserving for the data in auditing.

4. The proposed scheme

The details of the proposed scheme are introduced in this section. First, a high description is resented, which describes the main process of the proposed auditing scheme. Then, the detailed description of the proposed scheme is provided, including the phases of Key Generation, Data Tag Generation, Data Auditing Challenge, Storage Proof Generation and Data Integrity Auditing.

4.1. High description

The proposed scheme mainly focuses on the distributed data integrity auditing in edge computing. Moreover, the edge computing in this paper is managed and controlled by using the paradigm of the software-definition to fully schedule the resource of storage, service and computing in edge computing. In the proposed scheme, the certificateless cryptography is used in the design of the proposed auditing protocol. The main computation of the authenticator and tag in the previous auditing [22–26] has been eliminated. That is to say, the proposed data integrity auditing has low computational cost at the terminal side. In the data tag generation step, the terminal can compute the data block tags and the corresponding data file tag using the secret value and the public key. Then, the data tags along with the data file will be outsourced to the edge devices for storage. When the ECS initiates a challenge of storage auditing, the edge devices can compute the storage proof for the data according to the stored tags and the received challenge information. Finally, the ECS checks whether the challenged data storage is correct

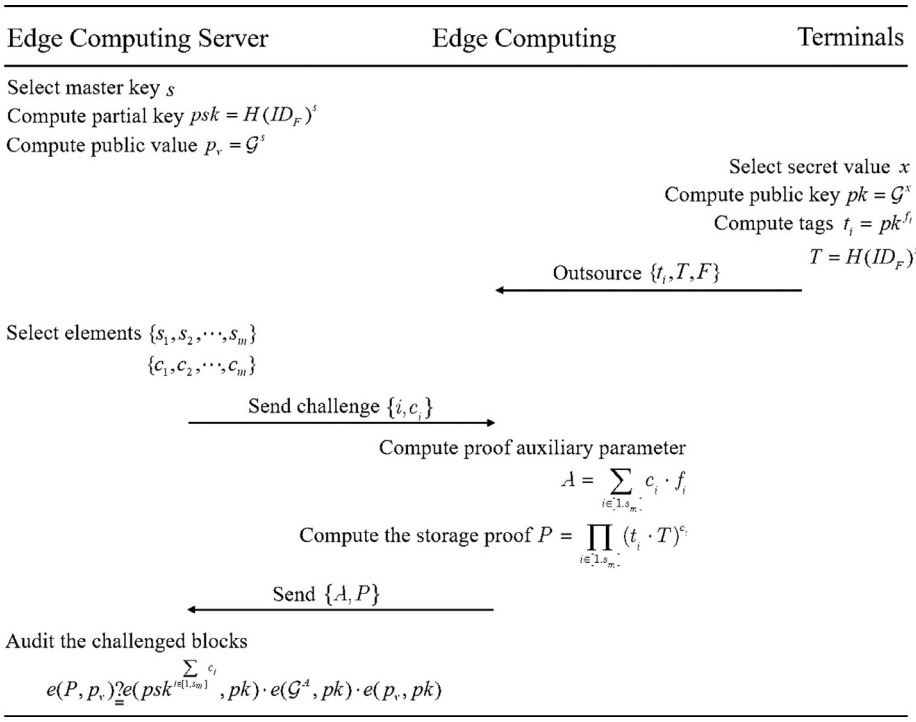

Fig. 3. The main process of the proposed scheme.

and complete by verifying the storage proof using the public keys and its own security parameters.

4.2. The detailed description

The proposed auditing scheme mainly consists of five steps, including Key Generation, Data Tag Generation, Data Auditing Challenge, Storage Proof Generation and Data Integrity Auditing. The main process of the auditing protocol is shown as Fig. 3. Note that the notations with the corresponding explanation are provided in Table 1. The detailed description of the proposed scheme is shown as follows.

(1) **Key Generation.** In the system setup, two multiplicative groups of $\mathbb{G}_1$ and $\mathbb{G}_2$ are generated. The prime order of groups $\mathbb{G}_1$ and $\mathbb{G}_2$ is q. The generator of $\mathbb{G}_1$ is $\mathcal{G}$. Here, a bilinear pairing is denoted as $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Then, a parameter of s is randomly chosen from $\mathbb{Z}_q^*$ as the master key by the ECS. To generate the partial key for the system, the ECS selects a hash function H for the system that can map a value to a point on $\mathbb{G}_1$. Here, the data file of one terminal stored in edge devices is denoted as $F = \{f_1, f_2, \cdots, f_n\}$. The partial key can be computed as $psk = H(ID_F)^s$ based on the master key and the identifier of the data file stored in edge devices. After that, the ECS computes a public value $p_v = \mathcal{G}^s$ for the further data integrity auditing.

(2) **Data Tag Generation.** In this step, the terminals can generate the secret value for the public key computation and the data encryption. Suppose that a terminal in the system selects a security parameter x from $\mathbb{Z}_q^*$. Note that the parameter of x is a secret value of the terminal. Then, the public key of the terminal can be computed as $pk = \mathcal{G}^x$ by the terminal using the secret value. To improve the storage security of the data stored in the edge devices, the terminal can compute a block tag for f_i as $t_i = pk^{f_i} = \mathcal{G}^{x \cdot f_i}$ using its public key. For the data block integrity auditing, the data file tag can be computed as $T = H(ID_F)^x$ by the terminal utilizing its secret value x. Finally, the block tags and the data file tag are outsourced to edge devices via the network for the storage.

(3) **Data Auditing Challenge.** In the proposed scheme, the ECS is responsible for auditing the stored data blocks of terminals. The periodically sampling auditing to the data blocks stored in the edge devices can improve the security of the data storage. Moreover, the auditing result of the stored data is an important factor for evaluating the reputation of the edge devices in the data storage. Suppose that the number of data blocks of one terminal is n. Suppose that the ECS wants to audit m data blocks for the terminal in edge computing. Here, $1 \leq m \leq n$.
In the phase of auditing challenge generation, the ECS first randomly selects m elements from $\mathbb{Z}_q^*$ as $\{s_1, s_2, \cdots, s_m\}$. Then, the ECS chooses c_i from $\mathbb{Z}_q^*$ for every s_i. The challenge for the m data blocks auditing can be denoted as $\{i, c_i\}$. Finally, the ECS sends the auditing challenge to edge computing for the corresponding data blocks storage proof computation.

(4) **Storage Proof Generation.** According to the received challenge, the edge devices that store the challenged data blocks will compute the storage proof. First, the edge devices compute a proof auxiliary parameter based on the challenge element c_i and the stored blocks f_i. The proof auxiliary parameter of A can be computed as $A = \sum_{i \in [1, s_m]} c_i \cdot f_i$. Then, the edge devices compute the corresponding

Table 1
Notations.

Notation	Description
$\mathbb{G}_1, \mathbb{G}_2$	Multiplicative groups
$\mathcal{G}$	Generator of group $\mathbb{G}_1$
$\mathbb{Z}_q^*$	Integers set of 1 to $q-1$
q	Large prime order
s, psk	Master key and partial key
H	Hash function
F, f_i	Data file and data blocks
ID_F	Identifier of the data file
p_v	Public value of ECS
x	Secret value of the terminal
pk	Public key of the terminal
t_i, T	Data block tag and data file tag
n, m	Number of data blocks and challenged data blocks
s_i, c_i	Radom elements for the auditing challenge
A, P	Proof auxiliary parameter and the storage proof

storage proof of the data blocks as $P = \prod_{i\in[1,s_m]} (t_i \cdot T)^{c_i}$. Finally, the edge devices send $\{A, P\}$ to the ECS for the integrity auditing.

(5) **Data Integrity Auditing.**
Upon receiving the storage proof from the edge devices, the ECS can check the correctness of the storage proof using the public key of the terminal, the generated public value and partial key. According to the correctness of the following Eq. (1), the ECS can determine whether the audited data blocks have been corrupted or stored completely in edge devices.

$$e(P, p_v) \stackrel{?}{=} e(psk^{\sum_{i\in[1,s_m]} c_i}, pk) \cdot e(\mathcal{G}^A, pk) \cdot e(p_v, pk) \quad (1)$$

5. Evaluation

To demonstrate the advantages of the proposed auditing scheme, the security and the performance are evaluated in this section. First, the proofs of the correctness and the security of the proposed scheme are presented in security analysis. Then, the computational efficiency of the proposed scheme is analyzed in the performance analysis according to the theoretical comparison and the simulation of the computational overhead.

5.1. Security analysis

Three theorems of the proposed scheme and the corresponding proofs are given. The details are shown as the following description.

Theorem 1. *The proposed data integrity auditing can be proved to be correct if the required security parameters are generated correctly according to the protocol.*

Proof. Suppose that all the entities are honest and can successfully generate the required parameters and keys defined in the abovementioned auditing protocol. The correctness of the auditing protocol can be proved by elaborating Eq. (1). The left-hand side of Eq. (1) can be computed as follows.

$$\begin{aligned} &e(P, p_v) \\ &= e(\prod_{i\in[1,s_m]} (t_i \cdot T)^{c_i}, \mathcal{G}^s) \\ &= e(\prod_{i\in[1,s_m]} (\mathcal{G}^{x\cdot f_i} \cdot H(ID_F)^x)^{c_i}, \mathcal{G}) \\ &= e(\prod_{i\in[1,s_m]} (\mathcal{G}^{f_i} \cdot H(ID_F))^{s\cdot x\cdot c_i}, \mathcal{G}) \end{aligned}$$

Similarly, the right-hand side of Eq. (1) can also be elaborated using the keys and parameters mentioned in the details of the proposed scheme. The elaboration of the right-hand side of Eq. (1) is provided as follows.

$$\begin{aligned} &e(psk^{\sum_{i\in[1,s_m]} c_i}, pk) \cdot e(\mathcal{G}^A, pk) \cdot e(p_v, pk) \\ &= e(H(ID_F)^{s\cdot \sum_{i\in[1,s_m]} c_i}, \mathcal{G}^x) \cdot e(\mathcal{G}^{\sum_{i\in[1,s_m]} c_i\cdot f_i}, \mathcal{G}^x) \cdot e(\mathcal{G}^s, \mathcal{G}^x) \\ &= e(\prod_{i\in[1,s_m]} (H(ID_F))^{s\cdot c_i}, \mathcal{G}^x) \cdot e(\prod_{i\in[1,s_m]} (\mathcal{G}^{c_i\cdot f_i})^s, \mathcal{G}^x) \\ &= e(\prod_{i\in[1,s_m]} (H(ID_F) \cdot \mathcal{G}^{f_i})^{s\cdot x\cdot c_i}, \mathcal{G}) \end{aligned}$$

According to the above elaboration of Eq. (1), it can be seen that the final results of the two side are the same. That is to say, the correctness of the proposed data integrity auditing scheme can be proved. ∗

Theorem 2. *The proposed scheme can resist the exposure of keys in the data auditing.*

Proof. In the phase of the data integrity auditing, the retrieved data storage proof from the edge devices can be checked by the ECS using the public value p_v, the partial secret key psk and the public key of the terminal. It can be seen that there is no need to use the terminal's secret keys and parameters in the auditing. Moreover, the public value and the partial secret key are generated by the ESC in Key Generation. That is to say, the ESC only uses the terminal's public key in the data integrity auditing. Hence, the property of the key exposure resistance can be proved. ∗

Table 2
Computational cost of different phase.

Phase	Computational Cost
Key Generation	$2T_{E.} + 1T_{H.}$
Data Tag Generation	$3T_{E.} + 1T_{H.}$
Storage Proof Generation	$mT_{E.} + 3mT_{Mul.} + T_{Add.}$
Data Auditing	$2T_{E.} + 4T_{P.} + mT_{Add.}$

*$T_{E.}, T_{H.}, T_{Mul.}, T_{Add.}, T_{P.}$: Time required to execute the corresponding operation.
*m: The number of the challenged data blocks.

Theorem 3. *The proposed auditing scheme can provide the property of privacy-preserving for the original data in the auditing.*

Proof. The ECS can audit the data storage in the edge devices according to the received proof auxiliary parameter and the storage proof. From the description in the step of Storage Proof Generation in Section 4.2, we can get that the proof auxiliary parameter $A = \sum_{i\in[1,s_m]} c_i \cdot f_i$. That is to say, when the number of the challenged blocks stored in the edge devices is large, the ECS cannot distinguish which element is the determined data block even if the ECS has random element c_i. That is to say, the true data blocks can be concealed in the proof auxiliary parameter by using the random elements. In the storage proof, the true data blocks are blinded in the data block tag t_i. Here, $t_i = pk^{f_i} = \mathcal{G}^{x\cdot f_i}$. The adversary can get $\mathcal{G}^{f_i}$ by computing t_i/pk. Then, the adversary tries to get f_i from $\mathcal{G}^{f_i}$. It other words, the DL (Discrete Logarithm) problem must be solved by the adversary if the adversary wants to obtain the original data block, which contradicts the DL assumption [29]. Hence, the proposed scheme supports the privacy-preserving for the terminal's data in the auditing.

5.2. Performance analysis

In order to show the high efficiency of the proposed auditing scheme in this paper, we list the computational cost of each phase and compare the computational operation amounts of them. For ease of understanding, the symbols of $T_{E.}, T_{H.}, T_{Mul.}, T_{Add.}, T_{P.}$ are used to denote the time that executes the corresponding operations of exponentiation, hash to point, multiplication, addition and bilinear pairing in the construction of the auditing protocol in this paper. Table 1 is the comparasion result of the main computational time of the four phases in the proposed auding protocol. According to the comparasion result listed in Table 1, it can be found that the computational cost in phases of Key Generation and Data Tag Generation is a constant. Note that the computational cost in the two phases of Storage Proof Generation and Data Auditing is determined by the number of the challenged data blocks. That is to say, with the growth of the challenged data blocks in the auditing request, the computational time of phases in Storage Proof Generation and Data Auditing will be higher. In addition, from Table 2, we can find that the computational cost of the phase in Data Tag Generation is larger than that of Key Generation.

The proposed auditing scheme is also simulated to present the real computational time cost. In the simulation analysis, we compare the time cost of the proposed auditing scheme with that of the previous schemes [22,23,25,26]. The comparative schemes are PPA [22], EPA [23], DPA [25] and ABD [26]. The experimental platform used to conduct the simulation is constructed based on MIRCAL cryptographic library. MTRCAL was developed by Shamus software Ltd. that can be used to perform the program of cryptography related to large number operation, including RSA public cryptography, Diffie Hellman key exchange,

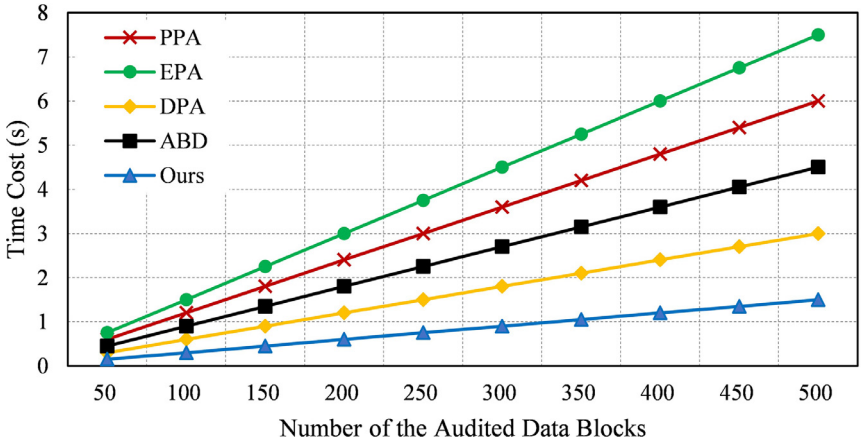

Fig. 4. The time cost of the server side under different audited data blocks.

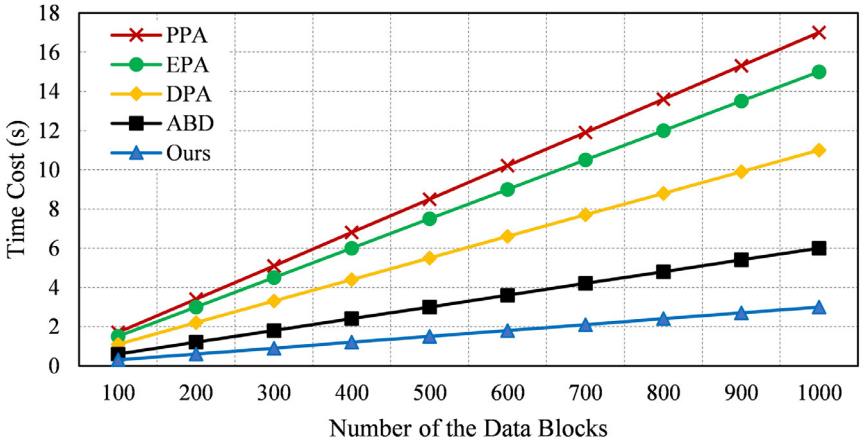

Fig. 5. The time cost of the terminal side under different data blocks.

AES, DSA digital signature and elliptic curve cryptography [30]. The programming languages used in the simulation are C++ and Python. The conducted computer is configured with 8GB RAM and 2.60 GHZ CPU.

First, we simulate the time cost of the server side that executes the auditing operations. In the simulation, the number of the audited data blocks ranges from 50 to 100. The simulation result under the different number of the audited data blocks is shown as Fig. 4. From Fig. 4, we can find that the time cost of the proposed auditing scheme and the comparative schemes will be higher with the increase of the number of the audited data blocks. Moreover, the time cost of the proposed scheme is always lower than that of the previous schemes under the condition of different audited data blocks. In particular, when the number of the audited data blocks is 500, the computational time cost of the server side in the proposed auditing scheme is only about 1/3 of the previous schemes' average time cost. In addition, the computational time of the terminal side is also simulated on the experimental platform. Fig. 5 is the simulation result of the time cost at the terminal side under different data blocks. Similar to that in Fig. 4, we can find that the time cost of the proposed auditing scheme and the previous schemes increases linearly with the growth of the data block number from Fig. 5. It is vital to show that the time cost of the terminal side in the proposed auditing scheme is always less compared to that in the previous schemes under the same number of data blocks. As the simulation results show, it can be concluded that the proposed scheme can be performed with high efficiency compared to the previous schemes.

6. Conclusion

To improve the security of the services provision in 5G-enabled software-defined edge computing, a secure distributed data integrity auditing with high efficiency is proposed in this paper. The technique of certificateless cryptography is used to construct the process of the security keys and parameters generation that can highly improves the computational efficiency of the terminal side in the system. Moreover, the proposed auditing scheme in this paper can realize the data integrity auditing by using the terminals' public keys and the ECS's own security parameters. In other words, the proposed can provide the property of the key exposure resistance for terminals. In addition, the privacy-preserving of the terminals' stored data blocks can be guaranteed in the data auditing. Security analysis proves the correctness of the proposed auditing scheme as well as the security properties of key exposure resistance and privacy-preserving. Performance analysis demonstrates that the proposed auditing scheme can be executed with low computational cost. According to the obvious advantages in security and efficiency, the proposed scheme can be well used to improve the security of edge computing in the aspect of the services provision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Science Foundation of China under Grant No. 62102169, No. 62072249, No. 12105120, the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 21KJB520033, the Key Research and Development Program (Social Development) of Lianyungang under Grant SF2102, the Research Start-up Fund of JOU under Grant KQ20039.

References

[1] W. Shi, C. Jie, Z. Quan, Y. Li, L. Xu, Edge computing: vision and challenges, Internet Things J. 3 (5) (2016) 637–646, doi:10.1109/JIOT.2016.2579198.
[2] D. Liu, Y. Zhang, D. Jia, Q. Zhang, X. Zhao, H. Rong, Toward secure distributed data storage with error locating in blockchain enabled edge computing, Comput. Standards Interfaces (2021), doi:10.1016/j.csi.2021.103560.
[3] W. Shi, S. Dustdar, The promise of edge computing, Computer (Long Beach Calif.) 49 (5) (2016) 78–81, doi:10.1109/MC.2016.145.
[4] J. Pan, J. McElhannon, Future edge cloud and edge computing for internet of things applications, IEEE Internet Things J. 5 (1) (2018) 439–449, doi:10.1109/JIOT.2017.2767608.
[5] Z. Sheng, S. Pfersich, A. Eldridge, J. Zhou, D. Tian, V.C.M. Leung, Wireless acoustic sensor networks and edge computing for rapid acoustic monitoring, IEEE/CAA J. Automatica Sinica 6 (1) (2019) 64–74, doi:10.1109/JAS.2019.1911324.
[6] W. Hou, Z. Ning, L. Guo, Green survivable collaborative edge computing in smart cities, IEEE Trans. Ind. Inf. 14 (4) (2018) 1594–1605, doi:10.1109/TII.2018.2797922.
[7] Y. Mao, J. Zhang, K.B. Letaief, Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC); 2017 March 19-22, IEEE, San Francisco, CA, USA, 2017, pp. 1–6, doi:10.1109/WCNC.2017.7925615.
[8] T. Liu, Y. Zhang, Y. Zhu, W. Tong, Y. Yang, Online computation offloading and resource scheduling in mobile-edge computing, IEEE Internet Things J. 8 (8) (2021) 6649–6664, doi:10.1109/JIOT.2021.3051427.
[9] P. Hu, W. Chen, C. He, Y. Li, H. Ning, Software-defined edge computing (sdec): principle, open iot system architecture, applications, and challenges, IEEE Internet Things J. 7 (7) (2020) 5934–5945, doi:10.1109/JIOT.2019.2954528.
[10] J. Sachs, L.A.A. Andersson, J. Araújo, C. Curescu, J. Lundsjö, G. Rune, E. Steinbach, G. Wikström, Adaptive 5g low-latency communication for tactile internet services, Proc. IEEE 107 (2) (2019) 325–349, doi:10.1109/JPROC.2018.2864587.
[11] J. Varga, A. Hilt, C. Rotter, G. Járó, Providing ultra-reliable low latency services for 5g with unattended datacenters, in: Proceedings of the International Symposium on Communication Systems, Networks & Digital Signal Processing; 2018 July 18-20, IEEE, Budapest, Hungary, 2018, pp. 1–4, doi:10.1109/CSNDSP.2018.8471756.
[12] C. Wang, J. Shen, P. Vijayakumar, B.B. Gupta, Attribute based secure data aggregation for isolated iot-enabled maritime transportation systems, IEEE Trans. Intell. Transp. Syst. (2021), doi:10.1109/TITS.2021.3127436.
[13] R. Rayarikar, A. Bokil, An encryption algorithm for end-to-end secure data transmission in manet, Int. J. Comput. Appl. 56 (16) (2012) 29–33, doi:10.5120/8977-3187.
[14] D. Liu, Y. Zhang, W. Wang, K. Dev, S.A. Khowaja, Flexible data integrity checking with original data recovery in iot-enabled maritime transportation systems, IEEE Trans. Intell. Transp. Syst. (2021), doi:10.1109/TITS.2021.3125070.
[15] J. Fu, Y. Liu, H. Chao, B.K. Bhargava, Z. Zhang, Secure data storage and searching for industrial iot by integrating fog computing and cloud computing, IEEE Trans. Ind. Inf. 14 (10) (2018) 4519–4528, doi:10.1109/TII.2018.2793350.
[16] B. Li, Q. He, F. Chen, H. Jin, Y. Xiang, Y. Yang, Auditing cache data integrity in the edge computing environment, IEEE Trans. Parallel Distrib. Syst. 32 (5) (2021) 1210–1223, doi:10.1109/TPDS.2020.3043755.
[17] H. Wang, J. Zhang, Y. Lin, H. Huang, ZSS signature based data integrity verification for mobile edge computing, in: Proceedings of the IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid); 2021 May 10-13, IEEE, Melbourne, Australia, 2021, pp. 356–365, doi:10.1109/CCGrid51090.2021.00045.
[18] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song, Provable data possession at untrusted stores, in: Proceedings of the 14th ACM Conference on Computer and Communications Security; 2007 October 28-31, ACM, Alexandria, Virginia, USA, 2007, pp. 598–609, doi:10.1145/1315245.1315318.
[19] A. Juels, B.S. Kaliski, Pors: proofs of retrievability for large files, in: Proceedings of the 14th ACM Conference on Computer and Communications Security; 2007 October 28-31, ACM, Alexandria, Virginia, USA, 2007, pp. 584–597, doi:10.1145/1315245.1315317.
[20] Q. Wang, C. Wang, J. Li, K. Ren, W. Lou, Enabling public verifiability and data dynamics for storage security in cloud computing, in: Proceedings of the 14th European Symp. Research in Computer Security (ESORICS 2009); 2009 September 21-23, Springer, Saint-Malo, France, 2009, pp. 355–370, doi:10.1007/978-3-642-04444-1_22.
[21] C. Wang, Q. Wang, K. Ren, W. Lou, Privacy-preserving public auditing for data storage security in cloud computing, in: Proceedings of the International Conference on Computer Communication (IEEE INFOCOM 2010); 2010 March 14-19, IEEE, San Diego, CA, USA, 2010, pp. 1–9, doi:10.1109/INFCOM.2010.5462173.
[22] Q. Wang, C. Wang, K. Ren, W. Lou, J. Li, Enabling public auditability and data dynamics for storage security in cloud computing, IEEE Trans. Parallel Distrib. Syst. 22 (5) (2011) 847–859, doi:10.1109/TPDS.2010.183.
[23] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, W. Lou, Privacy preserving public auditing for secure cloud storage, IEEE Trans. Comput. 62 (2) (2013) 362–375, doi:10.1109/TC.2011.245.
[24] K. Yang, X. Jia, An efficient and secure dynamic auditing protocol for data storage in cloud computing, IEEE Trans. Parallel Distrib. Syst. 24 (9) (2013) 1717–1726, doi:10.1109/TPDS.2012.278.
[25] H. Tian, Y. Chen, C.C. Chang, H. Jiang, Y. Huang, Y. Chen, J. Liu, Dynamic-hash-table based public auditing for secure cloud storage, IEEE Trans. Serv. Comput. 10 (5) (2017) 701–714, doi:10.1109/TSC.2015.2512589.
[26] M. Sookhak, F.R. Yu, A.Y. Zomaya, Auditing big data storage in cloud computing using divide and conquer tables, IEEE Trans. Parallel Distrib. Syst. 29 (5) (2018) 999–1012, doi:10.1109/TPDS.2017.2784423.
[27] D. Boneh, M.K. Franklin, Identity-based encryption from the weil pairing, in: Proceedings of the 21st Annual International Cryptology Conference (CRYPTO 2001); 2001 August 19–23, Springer, Santa Barbara, California, USA, 2001, pp. 213–229, doi:10.1007/3-540-44647-8_13.
[28] S.S. Al-Riyami, K.G. Paterson, Certificateless public key cryptography, in: Proceedings of the in International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2003); 2003 November 30-December 4, Springer, Taipei, Taiwan, 2003, pp. 452–473, doi:10.1007/978-3-540-40061-5_29.
[29] K. Nyberg, R.A. Rueppel, Message recovery for signature schemes based on the discrete logarithm problem, in: Proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques (EUROCRYPT 1994); 1994 May 9-12, Springer, Perugia, Italy, 1995, pp. 182–193, doi:10.1007/BFb0053434.
[30] X. Hu, H. Lu, H., .H Xu, J. Wang, Y. Yang, An efficient identity-based proxy signature scheme in the standard model with tight reduction, in: Proceedings of the International Joint Conference, Advances in Intelligent Systems and Computing (CISIS 2015); 2015 June 15-17, Springer, Burgos, Spain, 2015, pp. 309–319, doi:10.1007/978-3-319-19713-5_27.

Dengzhi Liu received the M.E. degree and Ph.D. degree from the School of Computer and Software, Nanjing University of Information Science and Technology, in 2017 and 2020, respectively. He is currently an Assistant Professor with the School of Computer Engineering, Jiangsu Ocean University, China. He mainly focuses on the security and privacy issues in data storage and transmission. He has authored more than 50 research papers and published in international conferences and journals. His-current research interests include cloud computing security, cyber security, and data security.

Zhimin Li received the Ph.D. degree from the School of Information Engineering, Beijing University of Posts and Telecommunications in 2009. She is currently a Lecture with the School of Computer Engineering, Jiangsu Ocean University, China. Her current research interests include applied cryptography, electronic forensics, blockchain and so on.

Dongbao Jia is an associate professor at School of Computer Engineering, Jiangsu Ocean University, Lianyungang, China. And he is also currently a special researcher at the University of Toyama, Toyama, Japan. He received his M.E. and Ph.D. degrees from the Department of Intellectual Information Engineering, Graduate School of Science and Engineering for Education, University of Toyama, Toyama, Japan, in 2015 and 2018, respectively. His-main research interests are intelligence algorithm, neural engineering and system security.