Journal of Network and Computer Applications 127 (2019) 59–69

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Privacy-preserving public auditing for secure data storage in fog-to-cloud computing

Hui Tian [a,*], Fulin Nan [a], Chin-Chen Chang [b], Yongfeng Huang [c], Jing Lu [d], Yongqian Du [a]

[a] College of Computer Science and Technology, National Huaqiao University, Xiamen, 361021, China
[b] Department of Information and Computer Science, Feng Chia University, Taichung, 40724, Taiwan
[c] Department of Electrical Engineering, Tsinghua University, Beijing, 100084, China
[d] Network Technology Center, National Huaqiao University, Xiamen, 361021, China

ARTICLE INFO

Keywords:
Public auditing
Data storage
Privacy protection
Data integrity
Fog-to-cloud computing
Internet of things

ABSTRACT

With increasing popularity of fog-to-cloud based Internet of Things (IoT), how to ensure the integrity of IoT data outsourced in clouds has become one of the biggest security challenges. However, little effort has been put into addressing the problem. To fill this gap, this paper presents a tailor-made public auditing scheme for data storage in fog-to-cloud based IoT scenarios, which can achieve all indispensable performance and security requirements. Particularly, we design a tag-transforming strategy based on the bilinear mapping technique to convert the tags generated by mobile sinks to the ones created by the fog nodes in the phase of proof generation, which cannot only effectively protect the identity privacy, but also reduce the communication and computational costs in the verification phase; moreover, we present a zero-knowledge proof mechanism to verify the integrity of IoT data from various generators (e.g., mobile sinks and fog nodes) while achieving perfect data-privacy preserving. We formally prove the security of our scheme and evaluate its performance by theoretical analysis and comprehensive experiments. The results demonstrate that our scheme can efficiently achieve secure auditing for data storage in fog-to-cloud based IoT scenarios, and outperforms the straight-forward solution in communication and computational costs as well as energy consumption.

1. Introduction

With the explosive growth of Internet-of-Things (IoT) devices in various industrial fields, fog computing, first proposed by Cisco in 2012 (Bonomi et al., 2012), have been popularly considered as a new paradigm that can effectively support geographically distributed, latency sensitive, and QoS-aware IoT applications (Zhang, 2016). As a middleware between IoT devices and clouds, fog computing nodes have their own basic computing, storage and networking resources to fulfill the requirements for data preprocessing and transmission. Thus, fog-to-cloud computing emerges as an attractive solution for data processing and storage in both delay-constraint and resource-constraint large-scale industrial applications (Shen et al., 2018; Zhang, 2016). It is estimated that, over half of IoT-created data, the amount of which would reach 500 zettabytes by 2019 (Cisco Visual Networking, 2014), would first be preprocessed by the fog nodes, and then outsourced to the clouds for enduring storage and further analysis (Roman et al., 2018). As a new cutting-edge technique, however, fog-to-cloud computing still faces many security challenges (Granjal et al., 2015; Li et al., 2018; Lin et al., 2017; Roman et al., 2018; Zhang et al., 2015). One of the biggest concerns is how to ensure the correctness and integrity of important data outsourced in the cloud. Like in the traditional cloud storage, cloud service providers (CSPs), for their own self-interest, may attempt to conceal the fact that some data has been corrupted due to internal and external attacks or Byzantine failures (Wang et al., 2010). Moreover, CSPs might neglect to keep or even deliberately delete rarely accessed data that belong to ordinary data owners to save storage space (Wang et al., 2011). Thus, it is significant to develop efficient auditing techniques to ensure the integrity of data in the cloud. So far, many auditing schemes for cloud data have been presented (Ateniese et al., 2007; Erway et al., 2009; Jiang et al., 2016; Juels and Kaliski, 2007; Shen et al., 2017; Tian et al., 2017; Wang et al., 2011, 2013; 2014, 2015; Yang and Jia, 2013; Yu et al., 2017; Yuan and Yu, 2015; Zhu et al., 2013). However, they cannot be directly put into practice in

* Corresponding author.
E-mail addresses: htian@hqu.edu.cn (H. Tian), flnan@hqu.edu.cn (F. Nan), alan3c@gmail.com (C.-C. Chang), yfhuang@tsinghua.edu.cn (Y. Huang), jlu@hqu.edu.cn (J. Lu), yqdu@hqu.edu.cn (Y. Du).

https://doi.org/10.1016/j.jnca.2018.12.004
Received 7 June 2018; Received in revised form 8 November 2018; Accepted 1 December 2018
Available online 6 December 2018
1084-8045/

fog-to-cloud based IoT scenarios for two main reasons. First, differing from traditional cloud data outsourced by users (data owners), the IoT data are generated by various devices. In this case, it is apparently inadvisable for the users to first retrieve these data and generate corresponding authenticators prior to outsourcing. Second, the existing auditing schemes do not involve the fog nodes, which, however, are crucial parities in fog-to-cloud computing. As mentioned above, the fog nodes can aid in the efficient processing and rapid transmission for large scale of IoT data, so they could be surely employed to ensure the data integrity. Therefore, in this paper, we are motivated to present a novel public auditing scheme for secure data storage in fog-to-cloud based IoT scenarios.

There are a great many successful industrial applications of fog-to-cloud based IoT. For clarifying the purpose of our work more clearly, we take a common application as an example, as shown in Fig. 1. In a highly modernized company, there are a certain number of factories, each of which consists of many workshops. Each workshop is surveilled under large quantities of sensors, gathering the related environmental data, such as temperature, humidity, dust concentrations, and brightness, in case of work safety accidents. Besides, some mobile sinks are pre-arranged at appropriate sites around the workshops (Hu and Hu, 2010; Kaswan et al., 2018) to collect the environmental data and transmit it to fog nodes. Each fog node is a local server cluster of the company for data processing and analysis. At last, all data are outsourced to the public clouds for cost-efficient storage and the other uses in the future. To ensure the integrity of the data, each mobile sink should create authenticatable metadata for collected data before sending them to the fog nodes; for the received data, the fog nodes should first verify the correctness prior to the local analysis and processing; for the processed data, the fog nodes should create new authenticatable metadata; finally, all data and their authenticatable metadata are transmitted to the clouds for long-term storage. The data owners or any other authorized parties should be able to verify the integrity of the data whenever and wherever, namely, conducting the remote auditing for the data correctness. It is the very purpose of this work to design a secure and efficient auditing scheme for data storage in the fog-to-cloud based IoT scenarios. Note that, there are two models for data auditing, namely, private auditing in which the verification operation is performed directly between users and CSPs (Erway et al., 2009; Juels and Kaliski, 2007), and public auditing in which the verification work is customarily done by an authorized third-party auditor (TPA) (Tian et al., 2017; Wang et al., 2011, 2013; Zhu et al., 2013). In comparison, the latter is widely deemed to be more practical and reasonable (Ateniese et al., 2007; Wang et al., 2010, 2011; 2013; Zhu et al., 2013), because it can provide more convincing auditing results while markedly alleviating the computational and communication overhead of users. Therefore, in this work, we are devoted to presenting a novel public auditing for secure data storage in fog-to-cloud computing. To achieve the purpose, the following vital problems should be addressed.

- ***Blockless verification***: the TPA can effectively verify the integrity of the target data using the proof provided by the CSP, without fetching back the actual data.
- ***Data-privacy preserving***: data privacy protection is an important issue for public auditing, which means that the TPA is not allowed to know any information about the data content while conducting credible auditing. Although introducing the data encryption prior to outsourcing is an approach to mitigate the privacy concern, it cannot totally prevent data leakage during the auditing process (Wang et al., 2013). Thus, it is significant for public auditing to include a mechanism independent to data encryption for data-privacy protection.
- ***Identity-privacy preserving***: in the auditing process, the TPA, who is usually considered to be trusted but curious, might collect the identity information of data generators to obtain significant privacy information, such as the production status or even patterns of the workshops during particular period of time, and the importance of the workshops in the factory. Thus, it is indispensable to protect the identity privacy of data generators in the auditing for fog-to-cloud computing.

To achieve the public auditing in the fog-to-cloud based IoT scenario, a straightforward solution is as follows: 1) in the phase of data collection, each mobile sink generates homomorphic verifiable authenticators (called tags) for all collected data blocks and sends them to the fog nodes; for the processed data blocks, the fog nodes should also create the corresponding tags; finally, the fog nodes send all the data blocks and their tags to the cloud. 2) In the verification phase, the CSP aggregates the proofs according to the tag generators (the mobile sinks and fog nodes), respectively, and sends them to the TPA; accordingly, the TPA conducts the auditing by verifying the proofs of each tag generator one by one. This basic scheme will be described in more detail in Section 3.1. In this basic scheme, owing to adopting homomorphic verifiable authenticators, the TPA just need to authenticate the block tags instead of original data blocks in the verification process, namely, achieve the blockless auditing (Ateniese et al., 2007). In the verification phase, the TPA can only obtain the aggregated proofs of each data generator, so the identity privacy can be also protected well. However, the TPA have to verify the proofs of each tag generator one by one, which will not only induce the large costs of communication between the TPA and CSP, but also impose heavy computational overhead on the TPA. In addition, as pointed out in (Wang et al., 2013), the linear combination of the data blocks cannot well protect the content privacy of the data.

Thus, in this paper, we seek to present a more efficient and secure public auditing scheme. Specifically, we present a tag-transforming strategy based on the bilinear mapping technique to convert the tags generated by mobile sinks to the ones created by the fog nodes in the phase of proof generation, which can not only effectively protect the identity privacy, but also reduce both the communication cost and computational overhead of the TPA in the verification phase. Moreover, we present a zero-knowledge proof mechanism to verify the integrity of IoT data from various generators (e.g., mobile sinks and fog nodes) while achieving perfect data-privacy preserving.

In general, our contributions in this paper can be summarized as follows:

1. We present a novel public auditing scheme for secure data storage in the fog-to-cloud based IoT scenario, which can fully satisfy the three crucial requirements, i.e., blockless verification, data-privacy protection, and identity-privacy protection. To the best of our knowledge, this is the first tailored public auditing scheme in the fog-to-cloud based IoT scenario.
2. We design a tag-transferring strategy based on the bilinear mapping technique to protect the identity privacy, and present a zero-knowledge proof mechanism for the privacy preserving of multi-source IoT data.
3. We formally prove the security of the proposed scheme, and evaluate its performance by theoretical analysis and experimental comparisons with the basic solution. The results show that the proposed scheme can effectively achieve secure auditing in fog-to-cloud computing, and outperforms the basic one in computation complexity, communication overhead and energy consumption.

The rest of the paper is organized as follows: We introduce the background and preliminaries in Section 2 and present our detailed solution for public auditing in the fog-to-cloud based IoT scenario in Section 3. We prove and analyze the security of our suggested scheme in Section 4, which is followed by comprehensive performance evaluation through theoretical and experimental analysis in Section 5. In Section 6, we review the related work on cloud data auditing, especially the public auditing schemes. Finally, Section 7 concludes this paper.

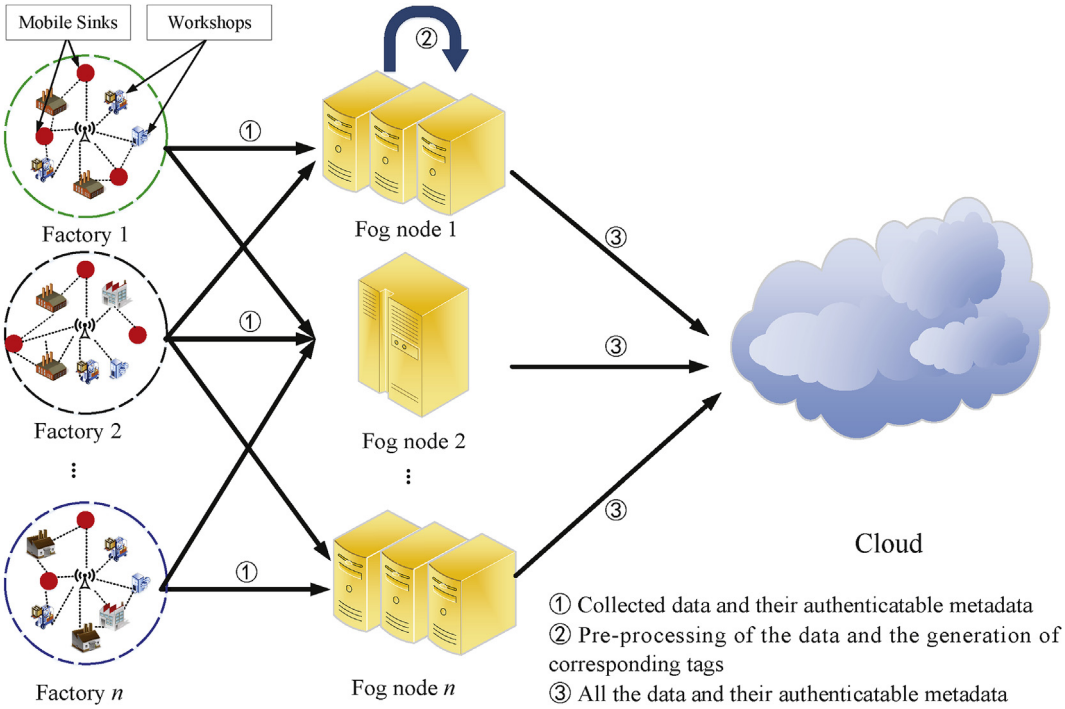

Fig. 1. An industrial application of the fog-to-cloud IoT.

2. Background and preliminaries

2.1. Problem statement

As shown in Fig. 2, the following five entities are involved in the public auditing scheme for secure data storage in the fog-to-cloud based IoT scenario.

- **IoT devices** record the raw data with the equipped sensors, and send them to mobile sinks in established formats.
- **Mobile sinks** are considered as the data sources in the fog-to-cloud computing. Each mobile sink collects data from many IoT devices, and organizes them into data blocks (Wang et al., 2017). Moreover, they create authenticatable tags for all the data blocks prior to transmitting them to the fog nodes for processing.
- **Fog nodes** are credible local computing centers, gathering the data from different mobile sinks. Obviously, it is indispensable to verify the integrity of the received data blocks prior to further operations. The fog nodes may further process some data. For the processed data, they should generate a new authenticatable tag for each block. For all the data blocks from the mobile sinks, they sign them again to enhance the security. Finally, all the data blocks and their authenticatable tags are sent to the cloud for long-term storage.
- **Cloud Service Provider (CSP)** coordinates and manages a lot of cloud servers to provide scalable and on-demand data storage services.
- **Third-party auditor (TPA)**, also called the public auditor, can verify the integrity of the outsourced data independently and dependably on behalf of data owners regularly or upon request.

As in the existing public auditing schemes for cloud storage (Ateniese et al., 2007; Jiang et al., 2016; Shen et al., 2017; Tian et al., 2017; Wang et al., 2011, 2013; 2014, 2015; Yang and Jia, 2013; Yu et al., 2017; Yuan and Yu, 2015; Zhu et al., 2013), we consider the TPA is honest-but-curious. That is to say, the TPA can provide credible results by independent auditing, but may be curious about the content of the cloud data and the identity information of data generators. In addition, the CSP is popularly considered to be semi-trusted. That is, the CSP can provide scalable and on-demand data storage services in the normal cases, but is most likely to conceal data corruption incidents intentionally for his/her own interests (Wang et al., 2010). To hide the fact of data corruption, the CSP may launch the following attacks to the TPA:

- *Forge attack.* The CSP attempts to forge the data blocks and corresponding tags to pass the verification.
- *Replacing attack.* The CSP attempts to replace a corrupted block and its tag with another block and its corresponding tag to pass the verification.
- *Replay attack.* The CSP attempts to pass the verification using the proofs generated previously.

To enable secure and efficient public auditing for data storage in the fog-to-cloud based IoT scenario, our scheme aims to achieve the following security and function requirements: *a*) Public auditing: Any authorized auditor is capable of verifying the correctness of data storage in the fog-to-cloud based IoT scenario; *b*) Blockless verification: The TPA can verify the integrity of the outsourced data without retrieving any data blocks; *c*) Auditing correctness: The TPA can offer a correct and credible auditing result for data integrity; *d*) Content-privacy preserving: The TPA cannot learn anything about the content of data during the whole auditing process; *e*) Identity-privacy preserving: The TPA cannot obtain any information about the identity of data generators; *f*) Lightweight: The public verification should be performed with the minimum communication and computational overhead.

2.2. Preliminaries

2.2.1. Bilinear map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups whose order is large prime p. A bilinear map is a map function $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ with following properties: *a*) **Computability**: there must be an efficient algorithm for computing the bilinear map e; *b*) **Bilinearity**: for all $g, h \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$, $e(g^a, h^b) = e(g, h)^{ab}$; *c*) **Non-degeneracy**: $e(g, g) \neq 1$.

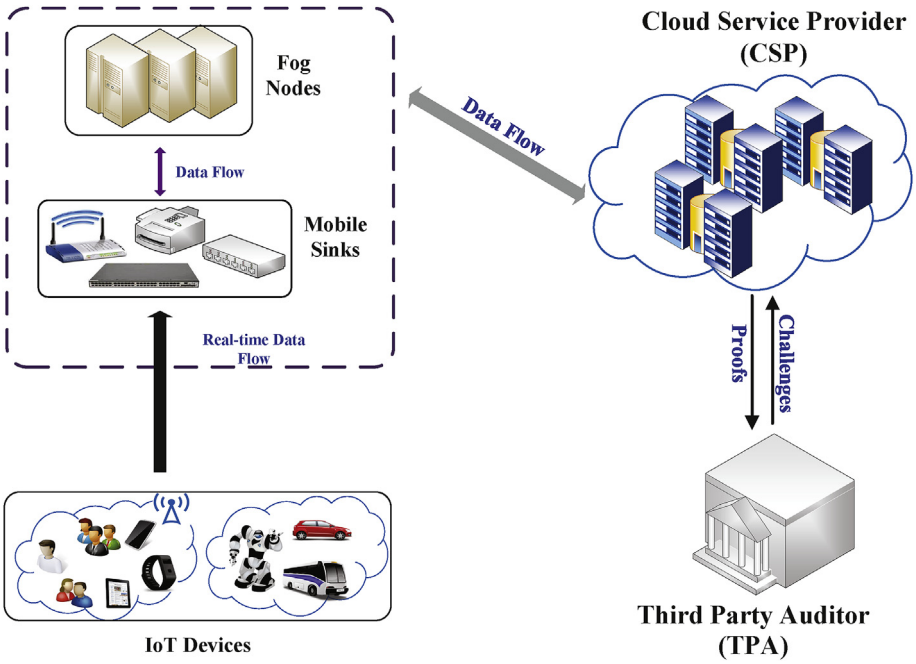

Fig. 2. Overview of public auditing in the fog-to-cloud based IoT scenario.

2.2.2. Equality of discrete logarithm

Let $\mathbb{G}$ be a finite cyclic group such that $|\mathbb{G}| = q$ for some prime q, and g_1, g_2 be generators of $\mathbb{G}$. H is a secure hash function such that $\mathbb{G} \rightarrow \{0,1\}^*$. The following protocol (Rogaway and Shrimpton, 2004; Yu et al., 2017) enables a prover P to prove to a verifier V that two elements Y_1, Y_2 have equal discrete logarithm to base g_1 and g_2 respectively in a non-interactive manner.

Commitment: P randomly chooses $\rho \in \mathbb{Z}_q$, then calculates $c = H(T_1 \parallel T_2)$ where $T_1 = g_1^{\rho}$, $T_2 = g_2^{\rho}$, and $z = \rho - cx \bmod q$, finally sends (c, z) to V.
Verify: V accepts the proof if and only if $c = H(g_1^z Y_1^c \| g_2^z Y_2^c)$ holds.

This protocol can be further extended to demonstrate the equality of discrete logarithm for multiple elements, which can be denoted as $POK\{(x) : Y_1 = g_1^x \wedge Y_2 = g_2^x \wedge Y_3 = g_3^x \wedge \ldots \wedge Y_n = g_n^x\}$.

2.2.3. Homomorphic verifiable authenticator

Homomorphic verifiable authenticator (HVA), is popularly adopted as a building block for public auditing (Ateniese et al., 2007; Jiang et al., 2016; Shen et al., 2017; Tian et al., 2017; Wang et al., 2011, 2013; 2014, 2015; Yang and Jia, 2013; Yu et al., 2017; Yuan and Yu, 2015; Zhu et al., 2013), which enables a public auditor to check the integrity of cloud data without retrieving any original data. Generally speaking, digital signatures (e.g. RSA-based signature and BLS signature) can be used to generate HVAs for public verification. Thus, HVAs for public auditing are considered as homomorphic verifiable signatures, also called homomorphic verifiable tags (HVTs). Besides unforgeability, HVTs satisfy the following properties (Ateniese et al., 2007):

- **Blockless verifiability.** Using the proof constructed by HVTs, the TPA can verify the integrity of the data blocks without knowing their actual data content.
- **Homomorphism.** Let $\mathbb{G}$ and $\mathbb{H}$ be multiplicative groups of a large prime order p, "$\oplus$" and "$\otimes$" be operations in $\mathbb{G}$ and $\mathbb{H}$. If a map function $f : \mathbb{G} \rightarrow \mathbb{H}$ satisfies homomorphism, then $\forall g_1, g_2 \in \mathbb{G}$, $f(g_1 \oplus g_2) = f(g_1) \otimes f(g_2)$.
- **Non-malleability.** Let σ_1 and σ_2 be signatures on blocks m_1 and m_2 respectively, α_1 and α_2 be two random numbers in $\mathbb{Z}_p$. For the given block, $m' = \alpha_1 m_1 + \alpha_2 m_2$, a user, who does not know the private key sk, cannot generate a legitimate signature σ' on m' by combining σ_1 and σ_2.

2.3. Security assumptions

The security of the proposed scheme is based on the following assumptions.

Definition 1 (Computational Diffe-Hellman (CDH) Assumption). *Let $\mathbb{G}$ be a cyclic group of prime order p, for a randomly chosen generator g and random numbers $a, b \in \mathbb{Z}_p$, given $(g, g^a, g^b) \in \mathbb{G}$, it is computationally intractable to compute the value g^{ab}. That is, for any probabilistic polynomial-time adversary $\mathcal{A}$, the probability of solving the CDH problem is negligible, namely,*

$$Pr(\mathcal{A}_{\mathrm{CDH}}(g, g^a, g^b \in \mathbb{G}) \rightarrow g^{ab} \in \mathbb{G} : \forall\, a, b \underset{\mathrm{R}}{\in} \mathbb{Z}_p) \leq \varepsilon \tag{1}$$

Definition 2 (Discrete Logarithm (DL) Assumption). *Let $\mathbb{G}$ be a cyclic group of prime order p with a generator g, for a given $h \in \mathbb{G}$, it is computationally intractable to compute the value $a \in \mathbb{Z}_p$, such that $h = g^a$. In other words, for any probabilistic polynomial-time adversary $\mathcal{A}$, the probability of solving the DL problem is negligible, namely,*

$$Pr(\mathcal{A}_{DL}(g, h \in \mathbb{G}) \rightarrow a \in \mathbb{Z}_p, s.t.\ h = g^a) \leq \varepsilon \tag{2}$$

3. The proposed schemes

In this section, we first start with a basic solution described in Section 3.1, which would not only induce the large costs of communication between the TPA and CSP, but also impose heavy computational overhead on the TPA. In view of the demerits of the basic scheme, we give our suggested scheme in Section 3.2.

3.1. The basic scheme

Assume that there are w mobile sinks, denoted as $S = \{s_1, s_2, \ldots, s_w\}$, and t fog nodes, denoted as $F = \{f_1, f_2, \ldots, f_t\}$. Usually, the data blocks collected by the mobile sinks are sent to the

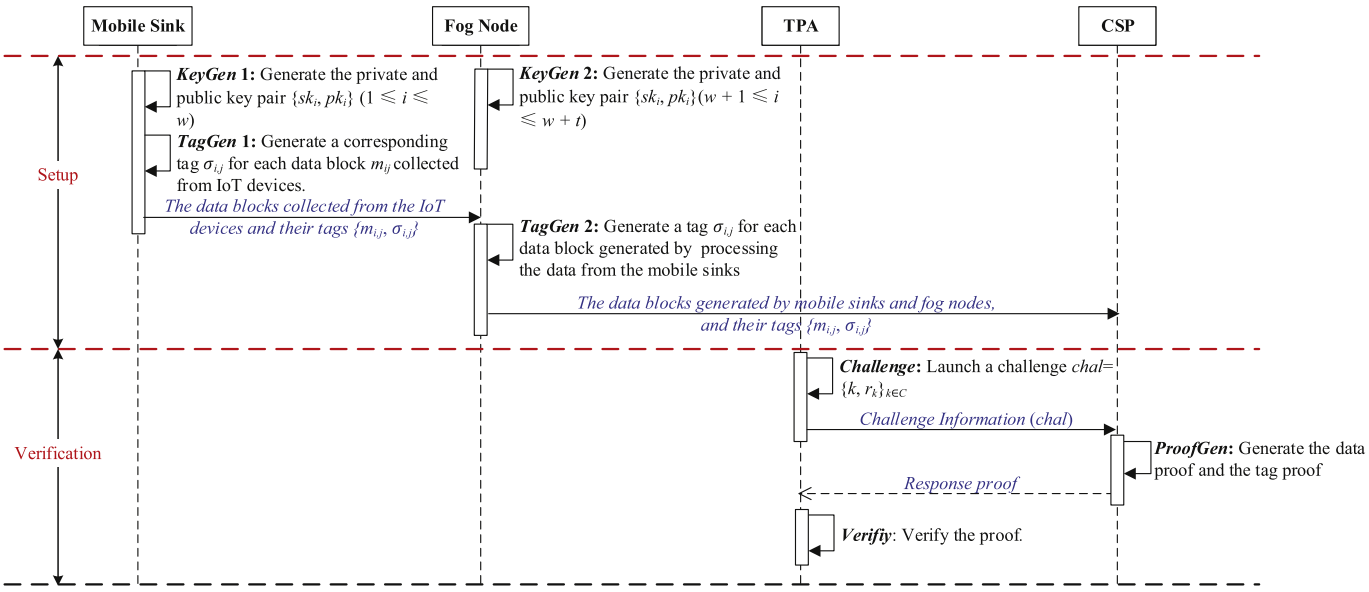

Fig. 3. Workflow of the auditing process for the basic scheme.

CSP through the fog nodes. Moreover, the fog nodes could further produce some new data by processing and analyzing the data from the mobile sinks. Let $M_i = \{m_{i,1}, m_{i,2}, \ldots, m_{i,n_i}\}$ be the set of data blocks collected or generated by the i-th entity, where n_i is the number of data blocks in M_i and $1 \leq i \leq w + t$. If $1 \leq i \leq w$, the i-th entity is the i-th mobile sink; otherwise, it is the v-th fog node, where $v = i–w$. Each entity has its own private and public key pairs, i.e., $sk_i = a_i, pk_i = g^{a_i}$, $a_i \in \mathbb{Z}_p^*$ is a random number. For each block $m_{i,j}$, the i-th entity creates a corresponding authenticatable tag $\sigma_{i,j} = (H(id_{i,j})g^{m_{i,j}})^{sk_i}$, where $H : \{0,1\}^* \to \mathbb{G}$ is a secure hash function and $id_{i,j}$ is the identifier of $m_{i,j}$. Finally, all the data blocks and their tags are sent to the CSP for long-term storage. When the TPA wants to verify the integrity of the data in the cloud, he/she generates a challenge message $\{k, r_k\}_{k \in C}$, where C is the identifier set of randomly selected data blocks for sampling checking and $r_k \in \mathbb{Z}_p^*$ is a random number. Once receiving the challenge message, the CSP first ascertains the required blocks are respectively generated by which entities. Let c_i be the identifier set of the challenged blocks handled by the i-th entity, where $1 \leq i \leq w + t$. Apparently, $C = c_1 \cup c_2 \cup \cdots \cup c_w \cup c_{w+1} \cup \cdots \cup c_{w+t}$. Further, for each entity (i.e., a fog node or mobile sink), the CSP computes the data proof $\{\Omega_i = \sum r_k m_{i,k}\}_{k \in c_i}$ and the tag proof $\{\Phi_i = \prod \sigma_{i,k}^{r_k}\}_{k \in c_i}$, and sends them to the TPA. After receiving all the proof messages, the TPA should conduct the verification of data integrity for all the fog nodes and the mobile sinks, respectively, since their public keys are different.

Fig. 3 illustrates the auditing process of the basic scheme. It is not hard to learn that, in the basic scheme, both the communication cost and the computational overhead for public verification are linear to the number of the involved entities including all the mobile sinks and fog nodes, which suggests that it is not well suitable for the industrial scenarios involving large-scale deployments of IoT devices and related computing entities. Moreover, the basic scheme cannot protect the content privacy of data well, since the linear combination of blocks may still potentially reveal data information.

3.2. The suggested scheme

Overview. In view of the demerits of the basic scheme, we present a more efficient and secure public auditing scheme in this section, as shown in Fig. 4. Specifically, to decrease the communication and computational overhead in the verification phase as well as protect the identity privacy, we present a tag-transforming strategy based on the bilinear mapping technique, which can convert the block tags generated by mobile sinks to the ones created by the fog nodes. Moreover, we design a zero-knowledge proof mechanism to verify the integrity of IoT data from the mobile sinks or/and fog nodes while achieving perfect data-privacy preserving. Note that, in the proposed scheme, we use the public key based HVAs to achieve public auditability as well as blockless verification. Particularly, we prefer to choose BLS-HVA for a shorter length of each block tag (Wang et al., 2011, 2015).

Scheme Details. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of large prime order p, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. g and u are the generators of $\mathbb{G}$. $H_1 : \{0,1\}^* \to \mathbb{Z}_p^*$, $H_2 : \{0,1\}^* \to \mathbb{G}$ and $H_3 : \mathbb{G}_T \to \mathbb{Z}_p^*$ are secure hash functions. The suggested scheme involves six algorithms, namely, ***KeyGen***, ***TagGenMS***, ***TagGenF***, ***Challenge***, ***ProofGen***, and ***Verify***.

KeyGen (Key Generation). Each mobile sink $s_i (i = 1, 2, \ldots, w)$ first selects a random number $a_i \in \mathbb{Z}_p^*$ as its private key (i.e., $ssk_i = a_i$), and calculates its public key as $spk_i = g^{1/a_i}$. Each fog node $f_v (v = 1, 2, \ldots, t)$ chooses a random number $b_v \in \mathbb{Z}_p^*$ as its private key (i.e., $fsk_v = b_v$), and calculates its public key as $fpk_v = g^{b_v}$.

TagGenMS (Tag Generation for Mobile Sinks). For each data block $m_{i,j} (j = 1, 2, \ldots, n_i)$, the responsible mobile sink s_i generates a corresponding tag $\sigma'_{i,j}$ with its private key ssk_i as

$$\sigma'_{i,j} = \left(u^{H_1(id_{i,j})} \cdot g^{m_{i,j}}\right)^{ssk_i}. \tag{3}$$

Further, each mobile sink would upload its data blocks and the corresponding tags to the fog node.

TagGenF(Tag Generation for Fog Nodes). For each received data block $m_{i,j}$, the corresponding fog node $f_v (v = 1, 2, \ldots, t)$ would first verify its integrity by checking the following equation:

$$e\left(\sigma'_{i,j}, g\right) \stackrel{?}{=} e\left(u^{H_1(id_{i,j})} \cdot g^{m_{i,j}}, spk_i\right). \tag{4}$$

If the result is FALSE, f_v would ask the responsible mobile sink s_i for retransmitting $m_{i,j}$ and $\sigma'_{i,j}$; otherwise, it would sign the corresponding tag again, i.e.,

$$\sigma_{i,j} = (\sigma'_{i,j})^{fsk_v}. \tag{5}$$

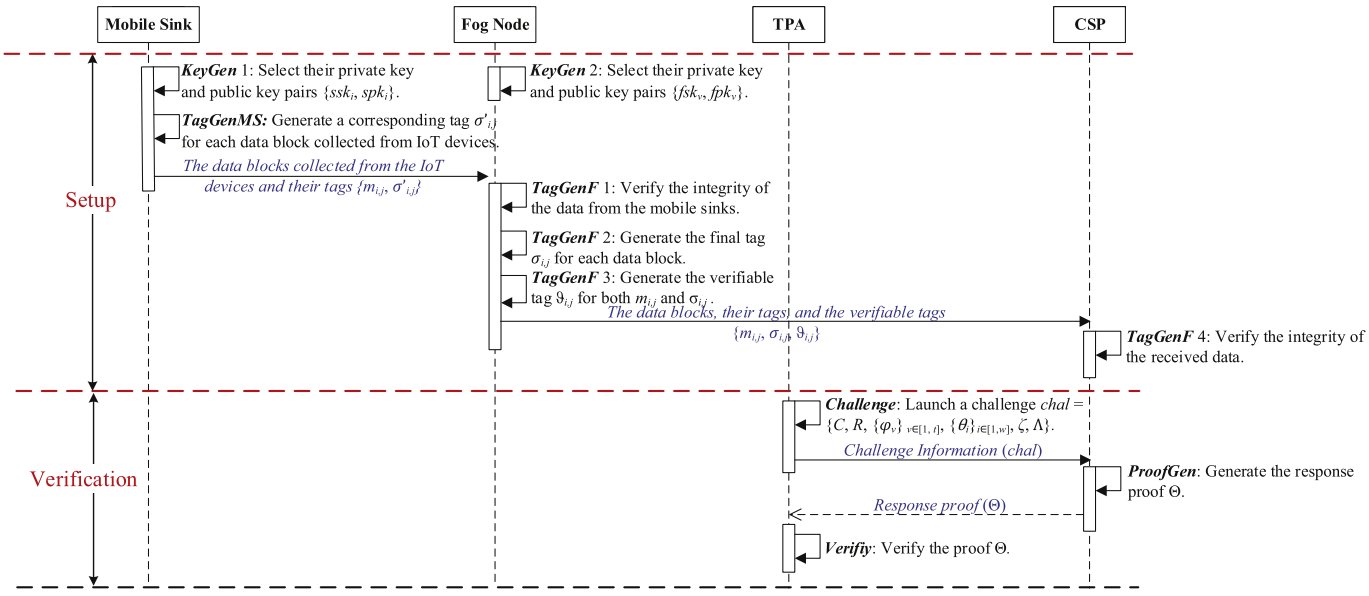

Fig. 4. Workflow of the auditing process for the suggested scheme.

This is a necessary pre-operation for the tag transformation in ***ProofGen***. Now, $\sigma_{i,j}$ is the final tag for $m_{i,j}$. Further, f_v would generate a signature $\vartheta_{i,j}$ for both $m_{i,j}$ and $\sigma_{i,j}$ as

$$\vartheta_{i,j} = \left(H_2(\sigma_{i,j} \| m_{i,j})\right)^{fsk_v}. \tag{6}$$

f_v would send all the data blocks and their final tags to the CSP. For each received data block $m_{i,j}$, the CSP would first verify its integrity by checking the following equation:

$$e\left(\vartheta_{i,j}, g\right) \stackrel{?}{=} e\left(H_2(\sigma_{i,j} \| m_{i,j}), fpk_v\right). \tag{7}$$

If the result is TRUE, the CSP would store $m_{i,j}$ and its tag $\sigma_{i,j}$; otherwise, the CSP would ask f_v for retransmitting $m_{i,j}$, $\sigma_{i,j}$ and $\vartheta_{i,j}$.

In addition, for each data block $m_{q,j}$ produced by itself where $q = w + v$, f_v will also generate a corresponding tag as

$$\sigma_{q,j} = \left(u^{H_1(id_{q,j})} \cdot g^{m_{q,j}}\right)^{fsk_v}. \tag{8}$$

These data blocks produced by f_v and their tags should be also sent to the cloud. For each data block $m_{q,j}$, the CSP would first verify its integrity by checking the following equation:

$$e\left(\sigma_{q,j}, g\right) \stackrel{?}{=} e\left(u^{H_1(id_{q,j})} \cdot g^{m_{q,j}}, fpk_v\right). \tag{9}$$

If the result is TURE, the CSP would store $m_{q,j}$ and its tag $\sigma_{q,j}$; otherwise, the CSP would ask f_v for retransmitting $m_{q,j}$ and $\sigma_{q,j}$.

Challenge. To verify the integrity of the data in the cloud, the TPA should generate challenge information as follows: 1) generate a challenge block set $C = \{ID_1, ID_2, \ldots, ID_\tau\}$ by randomly choosing τ different blocks from all the data blocks, and a set of random numbers $R = \{r_k \in \mathbb{Z}_p^* \mid k \in C\}$. 2) select a random number $\rho \in \mathbb{Z}_p^*$, and for each fog node $f_v (v \in [1, t])$, calculate

$$\varphi_v = e(g, fpk_v)^\rho; \tag{10}$$

For each mobile sink $s_i (i \in [1, w])$, calculate

$$\theta_i = spk_i^\rho = g^{\rho/a_i}; \tag{11}$$

For all the fog nodes, calculate

$$\zeta = g^\rho. \tag{12}$$

3) generate a knowledge proof (denoted by Λ) as

$$\Lambda = POK\{(\rho) : \varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_t \wedge \theta_1 \wedge \theta_2 \wedge \cdots \wedge \theta_w \wedge \zeta\}. \tag{13}$$

Finally, the TPA sends the challenge message $chal = \{C, R, \{\Phi_v\}_{v\in[1,t]}, \{\theta_i\}_{i\in[1,w]}, \zeta, \Lambda\}$ to the CSP.

ProofGen (Proof Generation). Once receiving the challenge from the TPA, the CSP would first verify the proof Λ according to the principle described in Section 2.2. If it is invalid, the CSP does not respond to the challenge; otherwise, it would generate the response proof for the correctness of data storage.

Assume that $\boldsymbol{C}_i (1 \le i \le w + t)$ is the identifier set of the challenged blocks handled by the i-th entity, which contains x_i elements. Apparently, $C = \boldsymbol{C}_1 \cup \boldsymbol{C}_2 \cup \cdots \cup \boldsymbol{C}_w \cup \boldsymbol{C}_{w+1} \cup \cdots \cup \boldsymbol{C}_{w+t}$ and $\sum_{i=1}^{w+t} x_i = \tau$. Accordingly, R can also be grouped into $w + t$ groups, denotes $R = \boldsymbol{R}_1 \cup \boldsymbol{R}_2 \cup \cdots \cup \boldsymbol{R}_{w+t}$. For each $id_{i,j} \in \boldsymbol{C}_i$ denotes the j-th identifier (element) of the i-th entity, so do the $r_{i,j}$. For each entity, the CSP computes the tag proof Φ_i as

$$\Phi_i = \prod_{j \in \boldsymbol{C}_i} \sigma_{i,j}^{r_{i,j}}, \tag{14}$$

and the data proof as

$$\Omega_i = \sum_{j \in \boldsymbol{C}_i} \left(m_{i,j} \cdot r_{i,j}\right). \tag{15}$$

Since the data blocks are resigned or generated by the fog nodes, they can be divided into t groups according to their handling fog nodes, denoted as $C = \mathfrak{C}_1 \cup \mathfrak{C}_2 \cup \cdots \cup \mathfrak{C}_t$, where $\exists i \in [1, w + t]$, $j \in [1, t]$, $\boldsymbol{C}_i \subset \mathfrak{C}_j$, and $\forall i, j \in [1, t]$, $i \ne j$, $\mathfrak{C}_i \cap \mathfrak{C}_j = \emptyset$. For each group (e.g, the v-th group), the CSP further calculates the aggregated data proof as

$$\Psi_v = \sum_{\boldsymbol{C}_i \subset \mathfrak{C}_v} \Omega_i. \tag{16}$$

In addition, the CSP obtains the response proof Θ as

$$\Theta = H_3\left(\left(\prod_{i=1}^{w} e\left(\Phi_i, \theta_i\right)\right) \cdot \prod_{v=1}^{t} e\left(\Phi_v, \zeta\right) \cdot \prod_{v=1}^{t} \varphi_v^{-\Psi_v}\right). \tag{17}$$

Finally, the CSP sends Θ back to the TPA as the response.

Verify. Upon receiving the proof Θ, the TPA calculates the hash value for each group of the challenged blocks as follows:

$$\boldsymbol{\mathcal{H}}_v = \sum_{j \in \mathfrak{C}_i} \left(r_{v,j} \cdot H_1\left(id_{v,j}\right)\right). \tag{18}$$

Further, the TPA verifies the correctness of data storage by checking the following equation:

$$\Theta \stackrel{?}{=} H_3\left(\prod_{v=1}^{t} e\left(u^{\mathcal{H}_v}, fpk_v^{\rho}\right)\right). \tag{19}$$

The correctness of the above verification equation can be demonstrated as follows.

$$\begin{aligned}
\Theta &= H_3\left(\left(\prod_{i=1}^{w} e\left(\Phi_i, \theta_i\right)\right) \cdot \prod_{v=1}^{t} e\left(\Phi_v, \zeta\right) \cdot \prod_{v=1}^{t} \varphi_v{}^{-\Psi_v}\right) \\
&= H_3\left(\left(\prod_{i=1}^{w} e\left(\prod_{j\in C_i} \sigma_{i,j}^{r_{i,j}}, g^{\rho/a_i}\right)\right) \cdot \prod_{v=1}^{t} e\left(\prod_{j\in C_{w+v}} \sigma_{v,j}^{r_{v,j}}, g^{\rho}\right)\right. \\
&\quad \left. \cdot \prod_{v=1}^{t} e(g, fpk_v)^{-\Psi_v\cdot\rho}\right) \\
&= H_3\left(\frac{\prod_{v=1}^{t} \prod_{C_i\subset\mathbb{G}_v, i\in[1,w]} e\left(\prod_{j\in C_i}\left(u^{H_1(id_{i,j})}\cdot g^{m_{i,j}}\right)^{a_i\cdot b_v\cdot r_{i,j}}, g^{\rho/a_i}\right)}{\prod_{v=1}^{t} e\left(g, g^{b_v}\right)^{\Psi_v\cdot\rho}}\right. \\
&\quad \left. \cdot \prod_{v=1}^{t} e\left(\prod_{j\in C_{w+v}}\left(u^{H_1(id_{w+v,j})}\cdot g^{m_{w+v,j}}\right)^{b_v\cdot r_{w+v,j}}, g^{\rho}\right)\right) \\
&= H_3\left(\frac{\prod_{v=1}^{t} \prod_{C_i\subset\mathbb{G}_v, i\in[1,w+t]} e\left(\prod_{j\in C_i}\left(u^{r_{i,j}\cdot H_1(id_{i,j})}\cdot g^{r_{i,j}m_{i,j}}\right), g^{\rho\cdot b_v}\right)}{\prod_{v=1}^{t} e\left(g^{\sum_{k\in\mathbb{G}_v} r_k\cdot m_k}, g^{\rho\cdot b_v}\right)}\right) \\
&= H_3\left(\frac{\prod_{v=1}^{t} e\left(\prod_{j\in\mathbb{G}_v}\left(u^{r_{v,j}\cdot H_1(id_{v,j})}\right)\cdot g^{\sum_{k\in\mathbb{G}_i} r_k\cdot m_k}, g^{\rho\cdot b_v}\right)}{\prod_{v=1}^{t} e\left(g^{\sum_{k\in\mathbb{G}_v} r_k\cdot m_k}, g^{\rho\cdot b_v}\right)}\right) \\
&= H_3\left(\prod_{v=1}^{t} e\left(\prod_{j\in\mathbb{G}_v} u^{r_{v,j}\cdot H_1(id_{v,j})}, g^{\rho\cdot b_v}\right)\right) = H_3\left(\prod_{v=1}^{t} e\left(u^{\mathcal{H}_v}, fpk_v^{\rho}\right)\right).
\end{aligned}$$

To sum up, compared with the basic scheme that employs the fog node only for data transferring, the suggested scheme takes full advantage of the fog node, and introduces a tag-transforming strategy based on the bilinear mapping technique to convert the tags generated by mobile sinks to the ones created by the fog nodes in the phase of proof generation, which can not only effectively protect the identity privacy but also reduce both the communication cost and computational overhead of the TPA in the verification phase. Moreover, the suggested scheme adopts a zero-knowledge proof mechanism to verify the integrity of IoT data efficiently while achieving perfect data-privacy preserving. Thus, the suggested scheme significantly outperforms the basic one, which would be also demonstrated by the experimental comparison described in Section 5.

4. Security analysis

Theorem 1 (Protection of Data Privacy). *From the response of the CSP, the TPA cannot obtain any knowledge of the data content.*

Proof. To respond to the challenge, the CSP generates

$$\begin{aligned}
\Theta &= H_3\left(\left(\prod_{i=1}^{w} e\left(\Phi_i, \theta_i\right)\right) \cdot \prod_{v=1}^{t} e\left(\Phi_v, \zeta\right) \cdot \prod_{v=1}^{t} \varphi_v{}^{-\Psi_v}\right) \\
&= H_3\left(\prod_{v=1}^{t} e\left(u^{\mathcal{H}_v}, fpk_v^{\rho}\right)\right)
\end{aligned}$$

and sends it back to the TPA.

Apparently, Θ is essentially a hash value. According to the property of preimage resistance for secure hash functions (Rogaway and Shrimpton, 2004; Stinson, 2006), it is computationally infeasible to find an x such that $H_3(x) = \Theta$, let alone obtain any information related to the data content. This completes the proof of Theorem 1.

Theorem 2 (Unforgeability of BLS-HVAs). *For any adversary $\mathcal{A}$, it is computationally infeasible to forge an HVA under BLS signature scheme, if the computational Diffie-Hellman (CDH) assumption holds.*

Proof. This theorem follows from the previous works (Tian et al., 2017; Wang et al., 2011), where it has been proven that the HVA scheme is existentially unforgeable, in that the BLS short signature scheme is secure with the assumption that the CDH problem is hard in bilinear groups (Boneh et al., 2001). Therefore, we omit the detailed proof here.

Theorem 3 (Immunity of forge attacks). *The proposed scheme can effectively resist forging attacks. That is, it is infeasible for the CSP to forge a valid proof to pass the verification.*

Proof. To respond to the challenge message $\{C, R, \Phi, \{\theta_i\}_{i\in[1,w]}, \zeta, \Lambda\}$, the CSP should provide a corresponding proof Θ. Therefore, we only need to prove the unforgeability of Θ. According to Eq. (17), if wants to forge Θ, the CSP should forge at least a tag proof Φ_i or/and at least a data proof Ω_i, where $i = 1, 2, \ldots, w + t$. In Theorem 2, it has been proven that the BLS-HVAs are unforgeable, so Φ_i, the aggregation value of the tags for the i-th entity (i.e., a fog node or mobile sink) cannot be forged too. Thus, we only need to prove the unforgeability of data proofs. To prove this, we design a game as follows: to respond to the challenge, the CSP provides a forged data proof Ω_i', namely,

$$\Omega_i = \sum_{j\in C_i}\left(m_{i,j}\cdot r_{i,j}\right) \neq \Omega_i' = \sum_{j\in C_i}\left(m_{i,j}'\cdot r_{i,j}\right),$$

which suggests that $\exists j \in C_i, m_{i,j} \neq m_{i,j}'$. If the CSP still passes the verification, he/she wins; otherwise, he/she fails. Suppose that the CSP wins the game, then,

$$\begin{aligned}
\Theta' &= H_3\left(\left(\prod_{i=1}^{w} e\left(\Phi_i, \theta_i\right)\right) \cdot \prod_{v=1}^{t} e\left(\Phi_v, \zeta\right) \cdot \prod_{v=1}^{t} \varphi_v{}^{-\Psi'_v}\right) \\
&= H_3\left(\frac{\prod_{v=1}^{t} \prod_{C_i\subset\mathbb{G}_v, i\in[1,w]} e\left(\prod_{j\in C_i}\left(u^{H_1(id_{i,j})}\cdot g^{m_{i,j}}\right)^{a_i\cdot b_v\cdot r_{i,j}}, g^{\rho/a_i}\right)}{\prod_{v=1}^{t} e\left(g, g^{b_v}\right)^{\Psi'_v\cdot\rho}}\right. \\
&\quad \left. \cdot \prod_{v=1}^{t} e\left(\prod_{j\in C_{w+v}}\left(u^{H_1(id_{w+v,j})}\cdot g^{m_{w+v,j}}\right)^{b_v\cdot r_{w+v,j}}, g^{\rho}\right)\right) \\
&= H_3\left(\frac{\prod_{v=1}^{t} e\left(\prod_{j\in\mathbb{G}_v}\left(u^{r_{v,j}\cdot H_1(id_{v,j})}\right)\cdot g^{\sum_{k\in\mathbb{G}_i} r_k\cdot m_k}, g^{\rho\cdot b_v}\right)}{\prod_{v=1}^{t} e\left(g^{\sum_{k\in\mathbb{G}_v} r_k\cdot m_k'}, g^{\rho\cdot b_v}\right)}\right) \\
&= H_3\left(\prod_{v=1}^{t} e\left(\prod_{j\in\mathbb{G}_v} u^{r_{v,j}\cdot H_1(id_{v,j})}\cdot g^{\sum_{k\in\mathbb{G}_i} r_k\cdot\left(m_k - m_k'\right)}, g^{\rho\cdot b_v}\right)\right) \\
&= H_3\left(\prod_{v=1}^{t} e\left(u^{\mathcal{H}_v}, fpk_v^{\rho}\right)\right).
\end{aligned}$$

According to the properties of bilinear maps, we can learn that

$$g^{\sum_{k\in\mathbb{G}_i} r_k\cdot\left(m_k - m_k'\right)} = 1 \rightarrow \sum_{k\in\mathbb{G}_i} r_k\cdot\left(m_k - m_k'\right) = 0.$$

Table 1
Comparison of communication costs.

Schemes	Challenge	Response
Scheme I	$O(t)$	$O(w)$
Scheme II	$O(t) + O(w)$	$O(1)$

Note: t is the number of challenged blocks and w is the number of the mobile sinks.

That is to say, $\forall k \in \mathfrak{C}_i, m_k = m'_k$, which apparently contradicts the assumption. Therefore, Ω_i is unforgeable. Further, we can conclude that the CSP cannot forge any valid proofs to pass the verification.

5. Performance evaluation

In this section, we will evaluate the performance of the suggested scheme (denoted as Scheme II) and compare it with the basic one (denoted as Scheme I) described in Section 3.1.

5.1. Theoretical analysis

The theoretical comparison between Scheme I and Scheme II involves three aspects, i.e., communication costs, computational costs and energy consumption.

5.1.1. Communication costs

Table 1 shows the communication costs of Scheme I and Scheme II in the challenge and response phases, from which we can learn that, for the challenge, the communication costs of Scheme I are $O(t)$, while the ones of Scheme II are $O(t) + O(w)$, a slightly larger than $O(t)$. The reason for this is that, to protect the data and identity privacy, some additional information (i.e., Λ and $\{\theta_i\}_{i\in[1,w]}$) should be sent to the CSP along with the challenge message, of which the communication costs are $O(w)$. Λ is the knowledge proof for the equality of discrete logarithm, and $\{\theta_i\}_{i\in[1,w]}$ are important parameters for hiding the actual generators (mobile sinks) of the data blocks. Thus, we prefer to trade slightly larger communication costs for privacy protection. In addition, in the response phase, the communication costs of Scheme II are $O(1)$ (actually $|\mathbb{G}_T|$ bits) while the ones of Scheme I are $O(w)$ ($2w \cdot |\mathbb{G}|$ bits in fact), because it needs to generate and transmit the data and tag proofs for each mobile sink and the fog node to the TPA. Therefore, in the response phase, Scheme II significantly outperforms Scheme I in the communication overhead.

5.1.2. Computational costs

Table 2 shows the computational costs of Scheme I and Scheme II in the phases of tag generation and verification, from which we can learn two facts as follows. First, the computational costs of the mobile sinks for tag generation in Scheme II are smaller than those in Scheme I, since $\mathcal{H}_1$ is much less than $\mathcal{H}_2$. Although a part of the work for tag processing is transferred to the fog node to protect the identity privacy, the total computational costs of Scheme II for tag processing are still significantly less than those of Scheme I, which is also demonstrated by the experimental results in Figs. 5 and 6. Second, in the verification phase, Scheme II just only needs to verify one proof according to the principle of zero knowledge proof while Scheme I has to verify the data and tag proofs for all the mobile sinks and the fog nodes one by one. Thus, the computational costs of Scheme II for verification are far less than those of Scheme I.

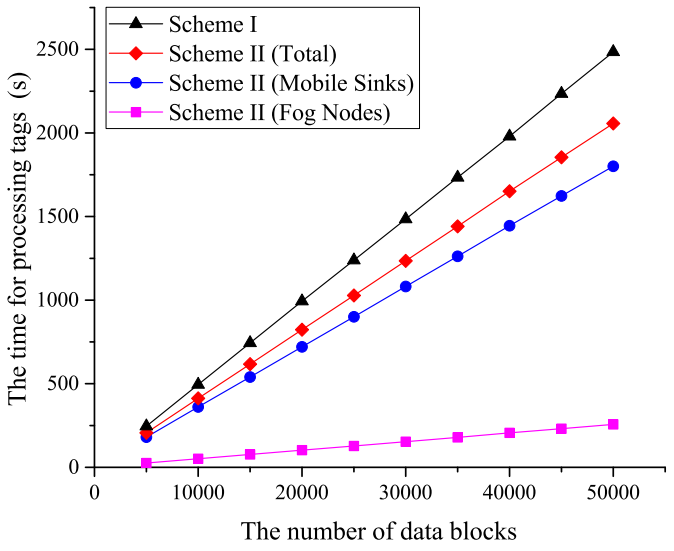

Fig. 5. Time of tag processing for different numbers of data blocks (block size = 4 KB).

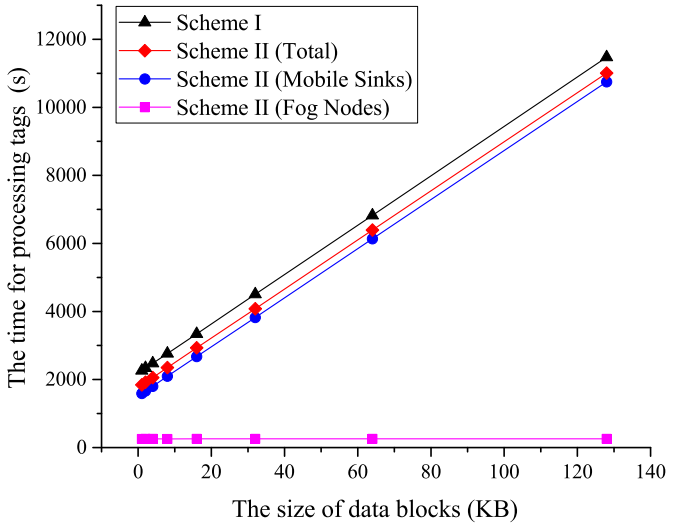

Fig. 6. Time of tag processing for different sizes of data blocks (the number of data blocks = 50000).

Table 2
Comparison of computational costs.

Schemes	TagGenMS	TagGenF	Verify
Scheme I	$n \cdot (\mathcal{H}_2 + \mathcal{M}_1 + 2\mathcal{E})$	$m \cdot (\mathcal{H}_2 + \mathcal{M}_1 + 2\mathcal{E})$	$t \cdot (\mathcal{E}+\mathcal{H}_2) + w \cdot (2\mathcal{P} + \mathcal{E} + \mathcal{M}_T)$
Scheme II	$n \cdot (\mathcal{H}_1 + \mathcal{M}_1 + 3\mathcal{E})$	$n \cdot (2\mathcal{E}+\mathcal{H}_2) + m \cdot (\mathcal{H}_1 + \mathcal{M}_1 + 2\mathcal{E})$	$2\mathcal{E} + \mathcal{P} + \mathcal{H}_3 + t \cdot \mathcal{H}_1$

Note: n is the number of all the data blocks; t is the number of all the challenged blocks; w is the number of mobile sinks; m is the number of the data blocks generated by the fog servers; $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$ are the average operation time for H_1, H_2 and H_3, respectively; $\mathcal{E}$ is the average operation time of the exponentiation on $\mathbb{G}$; $\mathcal{M}_1$ and $\mathcal{M}_T$ are the average operation time for multiplication on $\mathbb{G}$ and $\mathbb{G}_T$, respectively; $\mathcal{P}$ is the average operation time of the pairing computation.

Table 3
Comparison of energy consumption.

Schemes	Mobile Sink	Fog Node
Scheme I	$E_{1,C}(n\cdot(\mathcal{H}_2+\mathcal{M}_1+2\mathcal{E}))+E_{1,T}(n\cdot s\cdot b)$	$E_{2,C}(m\cdot(\mathcal{H}_2+\mathcal{M}_1+2\mathcal{E}))+E_{2,T}((n+m)\cdot s\cdot b)$
Scheme II	$E_{1,C}(n\cdot(\mathcal{H}_1+\mathcal{M}_1+3\mathcal{E}))+E_{1,T}(n\cdot s\cdot b)$	$E_{2,C}(n\cdot(2\mathcal{E}+\mathcal{H}_2)+m\cdot(\mathcal{H}_1+\mathcal{M}_1+2\mathcal{E}))+E_{2,T}((n+m)\cdot s\cdot b)$

Note: s is the number of segments per data block; b is the size of each segment; n is the number of all the data blocks; m is the number of the data blocks generated by the fog servers; $\mathcal{H}_1$ and $\mathcal{H}_2$ are the average operation time for H_1 and H_2, respectively; $\mathcal{M}_1$ is the average operation time for multiplication on $\mathbb{G}$; $\mathcal{E}$ is the average operation time of the exponentiation on $\mathbb{G}$; $E_{1,C}(x)$ is the energy consumption function of computing for mobile sinks; $E_{1,T}(y)$ is the energy consumption function of data transferring for mobile sinks; $E_{2,C}(x)$ is the energy consumption function of computing for fog nodes; $E_{2,T}(y)$ is the energy consumption function of data transferring for fog nodes.

5.1.3. Energy consumption

Table 3 shows the energy consumption of the mobile sinks and the fog nodes in Scheme I and Scheme II. Apparently, the energy consumption of each entity (i.e., mobile sink and fog node) is proportional to the computational and communication costs. It is easy to learn that, in both schemes, the energy consumption of communications for the mobile sinks (fog nodes) is identical. Therefore, the energy consumption for the mobile sinks (fog nodes) depends on their computational burden. As mentioned above, the computational costs of the mobile sinks (fog nodes) in Scheme II are smaller than those in Scheme I. Accordingly, we could conclude that the energy consumption for the mobile sinks (fog nodes) is less than that in Scheme I.

5.2. Experimental evaluations

In this section, we further evaluate the computational costs of Scheme II and compare it with Scheme I. We implement the prototype systems of the two schemes based on the Charm-Crypto Library v0.53 (the Python version of the Pairing-Based Cryptography (PBC) Library (0.5.14)). Each mobile sink is simulated by a Raspberry Pi 3 model B board with a quad-core 1.2 GHz Broadcom BCM2837 64bit CPU, an 1 GB LPDDR2 memory, a 16 GB Class 10 microSD card and the Raspbian Stretch operation system. The fog nodes and the TPA are simulated by a certain number of HP workstations, each of which is equipped with an Intel Core i5-6500 CPU @ 3.20 GHz, 2 × 4 GB DDR4 2133 MHz RAM, and 7200 RPM 2 TB Serial ATA drive with a 64 MB buffer. The cloud is simulated by an IBM x3850 X6 storage server with 2 Intel Xeon E7-4809 v3 @ 2.0 GHz CPUs, 4 × 8 GB TruDDR4 2133 MHz RAM, and 6 × 2 TB 7200 RPM 64 MB cache Serial ATA drive with RAID level 0. In the experiments, an MNT d159 curve, which has a 160-bit group order, is employed. All the statistical results are the averages of 20 trials.

5.2.1. Computational costs for generating tags

Figs. 5 and 6 respectively shows the experimental results of the time of tag processing for different numbers of data blocks and for different sizes of data blocks, from which we can learn that: 1) in both schemes, the time for generating tags is proportional to the block number (or block size); and 2) although the time of Scheme II involves the time of the mobile sinks for tag processing and the time of the fog node for tag processing, it is still significantly less than that of Scheme I for the same number of blocks or the same block size, which suggests Scheme II outperforms Scheme I in tag generation.

5.2.2. Computational costs in the verification phase

Fig. 7 shows the experimental results of the verification time for various numbers of mobile sinks, from which we can learn that the verification time of Scheme II is irrelevant with the number of mobile sinks, while the verification time of Scheme I increases with the number of mobile sinks. Moreover, for the same number of mobile sinks, the verification time of Scheme II is three orders of magnitude less than that of Scheme I. That to say, compared with Scheme I, Scheme II markedly reduces the computational overhead in the verification phase.

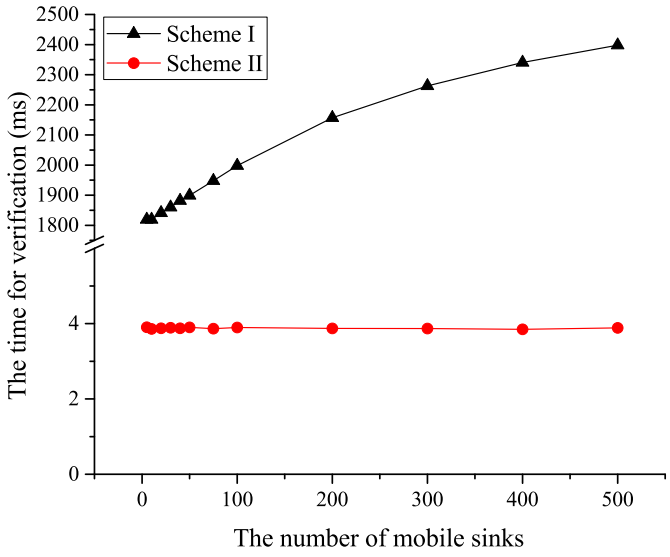

Fig. 7. Verification time for different numbers of mobile sinks.

6. Related work

With the increasing popularity and significance of IoT, its security and privacy concerns have attracted extensive attentions from industrial and academic communities (Singh et al., 2016). Many fruitful related studies have been conducted, such as, how to achieve secure communications among IoT nodes (Esposito et al., 2018b), how to preserve privacy while providing secure and cost-efficient data services for users in IoT applications (Belguith et al., 2018), and how to ensure the integrity for messages moving throughout the IoT infrastructure (Esposito et al., 2018a). However, from the existing literature, there have not been practical solutions for public auditing in the fog-to-cloud based IoT scenario (Granjal et al., 2015; Lin et al., 2017; Roman et al., 2018; Zhang et al., 2015), and our work in this paper is the first attempt. However, as the foundation of our study, we will briefly introduce some related work on public auditing in the traditional cloud storage.

One of the earliest work is proof of retrievability (PoR) proposed by Juels and Kaliski (2007), which combines the error-correcting code and spot-checking of data blocks to ensure the integrity as well as availability. However, PoR cannot support the auditing performed by a third party, and only support a limited number of verification operations. At the same time, Ateniese et al. (2007) first presented provable data possession (PDP) using RSA-based homomorphic authenticators, which can support both the public auditing and unlimited number of challenges. In addition to public verification, there are some other crucial security and function requirements for the cloud storage auditing, such as pri-

vacy preserving (Tian et al., 2017; Wang et al., 2013; Yang and Jia, 2013; Yu et al., 2017), dynamic auditing (Shen et al., 2017; Tian et al., 2017; Wang et al., 2011; Zhu et al., 2013), and shared data auditing (Jiang et al., 2016; Shen et al., 2017; Tian et al., 2019; Wang et al., 2015; Yuan and Yu, 2015).

Privacy preservation is popularly witnessed as an essential precondition of the public auditing (Tian et al., 2017; Wang et al., 2010, 2013). To address this concern, the random masking that can blind the data proof is widely adopted (Tian et al., 2017; Wang et al., 2013; Yang and Jia, 2013). Generally speaking, its implementation ways can be divided into two categories. In the first one (Tian et al., 2017; Yang and Jia, 2013), the TPA first generates a mask number R with a random number r and a global parameter y as $R = y^r$, and sends R to the CSP together with the challenge; while responding to the challenge, the CSP computes the masked data proof of M as $M' = e(u,R)^M$, where e is a bilinear map and u is a global parameter. In the other category (Wang et al., 2013), the CSP calculates a mask number $R = y^r$, and blinds the data proof of M by computing $M' = M + rH(R)$, where r is a randomly chosen number, y is a global parameter and H is a hash function. In addition, Yu et al. (2017) presented a zero-knowledge proof strategy based on identity authentication to protect data privacy. However, because it adopts the identity-based signature, the scheme cannot aggregate the proofs from different identities, except for the batch auditing strategy. That is, this strategy can only support the data from a single source, and cannot be directly applied in the IoT scenario where the data are collected from many different sources, namely, various mobile sinks and the fog nodes. Therefore, in this paper, we present a new zero-knowledge proof mechanism to achieve the auditing for the multisource IoT data while protecting their content privacy well.

To support data dynamics, authenticated data structures are widely introduced into the public auditing schemes (Shen et al., 2017; Tian et al., 2017; Wang et al., 2011, 2014; 2015; Zhu et al., 2013). For example, Wang et al. (2011) introduced Merkle hash tree to achieve public auditing for dynamic data; Zhu et al. (2013) designed an authenticated data structure, called index hash table, to achieve data dynamics; Tian et al. (2017) further presented a two-dimensional data structure, dynamic hash table, to achieve both the dynamic data updating and public auditing effectively; Shen et al. (2017) proposed another novel dynamic structure, including a doubly linked info table and a location array, to achieve public auditing for dynamic data.

With the increasing popularity of collaboration in the cloud, shared data auditing, has also attracted extensive attention from the research community. For example, Wang et al. (2014) successively proposed a shared-data auditing scheme based on ring signatures, called Oruta, and a shared-data auditing scheme based on proxy re-signatures, called Panda (Wang et al., 2015); Yuan and Yu (2015) presented an efficient public auditing scheme for shared data using the polynomial-based authentication technique; Jiang et al. (2016) designed a public auditing scheme based on a group signature algorithm with verifier-local revocation, which can achieve verification for shared data while supporting secure user revocation; Recently, Tian et al. (2019) presented a comprehensive public auditing scheme for shared data to achieve all indispensable functional and security requirements.

Although there are fruitful public auditing schemes in the traditional cloud storage, few of them can be directly extended to achieve secure and efficient verification for data storage in the fog-to-cloud based IoT scenarios, for two main reasons. First, the data are generated by various IoT devices, instead of the data owners themselves. Thus, it is necessary and advisable to enable some devices rather than the data owners to generate corresponding authenticatable metadata to ensure the data integrity. Second, in the fog-to-cloud computing, fog nodes, not involved in the traditional cloud storage, play important roles in the processing and transmission of IoT data. Accordingly, in the public auditing scheme, it is essential and feasible to employ the fog nodes to achieve secure and efficient verification for data integrity. Therefore, in this paper, we are devoted to present a tailor-made public auditing scheme for data storage in fog-to-cloud based IoT scenarios.

7. Conclusion and future work

In recent years, fog-to-cloud computing has become a promising and attractive solution for data processing and storage in large-scale industrial Internet-of-things applications. However, as a new cutting-edge technique, fog-to-cloud computing still faces many security challenges. One of the biggest concerns is how to ensure the correctness of the important data outsourced to the cloud. To address this problem, we present a novel public auditing scheme for data storage in the fog-to-cloud based IoT scenarios, which can achieve all the indispensable performance and security requirements, such as blockless verification, protection of data privacy and preserving users' identity privacy. Specifically, we employ the public key based homomorphic verifiable authenticators to achieve public auditability as well as blockless verification; we design a tag-transforming strategy based on the bilinear mapping technique to convert the tags generated by mobile sinks to the ones created by the fog nodes in the phase of proof generation, which can not only protect the identity privacy effectively, but also reduce the communication costs and computational overhead in the verification phase; we present a zero-knowledge proof mechanism to verify the integrity of the IoT data from various generators (e.g., mobile sinks and fog nodes) while well protecting data privacy. We formally prove the security of the suggested scheme, and evaluate the performance by detailed experiments and comparisons with a straight-forward scheme extended from the traditional cloud data auditing scheme. The experimental results demonstrate that the proposed scheme can efficiently achieve public auditing for secure data storage in fog-to-cloud based IoT scenarios, and outperforms the straight-forward solution in communication costs, computational overhead and energy consumption.

Although this paper has provided a tailor-made public auditing scheme for data storage in fog-to-cloud based IoT scenarios, there are still some significant problems for ensuring data integrity and usability in the fog-to-cloud scenario, which deserve further study. For example, how to achieve secure and effective integrity auditing while supporting data deduplication in fog nodes and clouds; to address the challenge for the ever-increasing scale of big data in IoT, it may be advisable to design the most appropriate auditing algorithms for various data categories; moreover, in the future, it is also significant to construct a new distributed data-auditing model to achieve efficient auditing for big data as well as load balancing of multiple auditors.

Acknowledgment

The authors sincerely thank the anonymous reviewers for their valuable time and efforts spent in reviewing this paper. This work was supported in part by National Natural Science Foundation of China under Grant Nos. U1405254 and U1536115, Natural Science Foundation of Fujian Province of China under Grant No. 2018J01093, Program for New Century Excellent Talents in Fujian Province University under Grant No. MJK2016-23, Program for Outstanding Youth Scientific and Technological Talents in Fujian Province University under Grant No. MJK2015-54, and Research Project for Young Teachers in Fujian Province (Program for High-Education Informationization) under Grant No. JAT170055.

References

Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D., 2007. Provable data possession at untrusted stores. In: Proc. 14th ACM Conference on Computer and Communications Security. ACM, Alexandria, Virginia, USA, pp. 598–609, https://doi.org/10.1145/1315245.1315318.

Belguith, S., Kaaniche, N., Laurent, M., Jemai, A., Attia, R., 2018. PHOABE: securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT. Comput. Network. 133, 141–156, https://doi.org/10.1016/j.comnet.2018.01.036.

Boneh, D., Lynn, B., Shacham, H., 2001. Short signatures from the weil pairing. In: Advances in Cryptology — ASIACRYPT 2001. Springer, Berlin, Heidelberg, Gold Coast, Australia, pp. 514–532, https://doi.org/10.1007/3-540-45682-1_30.

Bonomi, F., Milito, R., Zhu, J., Addepalli, S., 2012. Fog computing and its role in the Internet of things. In: Proc. 1st Edition of the MCC Workshop on Mobile Cloud Computing. ACM, New York, NY, USA, pp. 13–16, https://doi.org/10.1145/2342509.2342513.

Cisco Visual Networking, 2014. Cisco Global Cloud Index: Forecast and Methodology, 2014-2019. Cisco white paper.

Erway, C., Küpçü, A., Papamanthou, C., Tamassia, R., 2009. Dynamic provable data possession. In: Proc. 16th ACM Conference on Computer and Communications Security. ACM, New York, NY, USA, pp. 213–222, https://doi.org/10.1145/1653662.1653688.

Esposito, C., Castiglione, A., Palmieri, F., Santis, A.D., 2018a. Integrity for an event notification within the Industrial Internet of Things by using group signatures. IEEE Trans. Ind. Inf. 14, 3669–3678, https://doi.org/10.1109/TII.2018.2791956.

Esposito, C., Ficco, M., Castiglione, A., Palmieri, F., Santis, A.D., 2018b. Distributed group key management for event notification confidentiality among sensors. IEEE Trans. Dependable Secure Comput. 1, https://doi.org/10.1109/TDSC.2018.2799227.

Granjal, J., Monteiro, E., Silva, J.S., 2015. Security for the Internet of Things: a survey of existing protocols and open research issues. IEEE Commun. Surv. Tutor. 17, 1294–1312, https://doi.org/10.1109/COMST.2015.2388550.

Hu, J., Hu, X., 2010. Nonlinear filtering in target tracking using cooperative mobile sensors. Automatica 46, 2041–2046, https://doi.org/10.1016/j.automatica.2010.08.016.

Jiang, T., Chen, X., Ma, J., 2016. Public integrity auditing for shared dynamic cloud data with group user revocation. IEEE Trans. Comput. 65, 2363–2373, https://doi.org/10.1109/TC.2015.2389955.

Juels, A., Kaliski Jr., B.S., 2007. PoRs: proofs of retrievability for large files. In: Proc. 14th ACM Conference on Computer and Communications Security. ACM, Alexandria, Virginia, USA, pp. 584–597, https://doi.org/10.1145/1315245.1315317.

Kaswan, A., Singh, V., Jana, P.K., 2018. A multi-objective and PSO based energy efficient path design for mobile sink in wireless sensor networks. Pervasive Mob. Comput. 46, 122–136, https://doi.org/10.1016/j.pmcj.2018.02.003.

Li, J., Chen, X., Chow, S.S.M., Huang, Q., Wong, D.S., Liu, Z., 2018. Multi-authority fine-grained access control with accountability and its application in cloud. J. Netw. Comput. Appl. 112, 89–96, https://doi.org/10.1016/j.jnca.2018.03.006.

Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., Zhao, W., 2017. A survey on Internet of Things: architecture, enabling technologies, security and privacy, and applications. IEEE Internet Things J. 4, 1125–1142, https://doi.org/10.1109/JIOT.2017.2683200.

Rogaway, P., Shrimpton, T., 2004. Cryptographic hash-function basics: definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In: Fast Software Encryption. Springer, Berlin, Heidelberg, Delhi, India, pp. 371–388, https://doi.org/10.1007/978-3-540-25937-4_24.

Roman, R., Lopez, J., Mambo, M., 2018. Mobile edge computing, fog et al.: a survey and analysis of security threats and challenges. Future Generat. Comput. Syst. 78, 680–698, https://doi.org/10.1016/j.future.2016.11.009.

Shen, J., Gui, Z., Ji, S., Shen, J., Tan, H., Tang, Y., 2018. Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. J. Netw. Comput. Appl. 106, 117–123, https://doi.org/10.1016/j.jnca.2018.01.003.

Shen, J., Shen, J., Chen, X., Huang, X., Susilo, W., 2017. An efficient public auditing protocol with novel dynamic structure for cloud data. IEEE Trans. Inf. Forensics Secur. 12, 2402–2415, https://doi.org/10.1109/TIFS.2017.2705620.

Singh, J., Pasquier, T., Bacon, J., Ko, H., Eyers, D., 2016. Twenty security considerations for cloud-supported Internet of Things. IEEE Internet Things J. 3, 269–284, https://doi.org/10.1109/JIOT.2015.2460333.

Stinson, D.R., 2006. Some observations on the theory of cryptographic hash functions. Des. Codes Cryptogr. 38, 259–277, https://doi.org/10.1007/s10623-005-6344-y.

Tian, H., Chen, Y., Chang, C.C., Jiang, H., Huang, Y., Chen, Y., Liu, J., 2017. Dynamic-hash-table based public auditing for secure cloud storage. IEEE Trans. Serv. Comput. 10, 701–714, https://doi.org/10.1109/TSC.2015.2512589.

Tian, H., Nan, F., Jiang, H., Chang, C.C., Ning, J., Huang, Y., 2019. Public auditing for shared cloud data with efficient and secure group management. Inf. Sci. 472, 107–125, https://doi.org/10.1016/j.ins.2018.09.009.

Wang, B., Li, B., Li, H., 2014. Oruta: privacy-preserving public auditing for shared data in the cloud. IEEE Trans. Cloud Comput. 2, 43–56, https://doi.org/10.1109/TCC.2014.2299807.

Wang, B., Li, B., Li, H., 2015. Panda: public auditing for shared data with efficient user revocation in the cloud. IEEE Trans. Serv. Comput. 8, 92–106, https://doi.org/10.1109/TSC.2013.2295611.

Wang, C., Chow, S.S.M., Wang, Q., Ren, K., Lou, W., 2013. Privacy-preserving public auditing for secure cloud storage. IEEE Trans. Comput. 62, 362–375, https://doi.org/10.1109/TC.2011.245.

Wang, C., Ren, K., Lou, W., Li, J., 2010. Toward publicly auditable secure cloud data storage services. IEEE Netw. 24, 19–24, https://doi.org/10.1109/MNET.2010.5510914.

Wang, Q., Wang, C., Ren, K., Lou, W., Li, J., 2011. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans. Parallel Distr. Syst. 22, 847–859, https://doi.org/10.1109/TPDS.2010.183.

Wang, T., Li, Y., Wang, G., Cao, J., Bhuiyan, M.Z.A., Jia, W., 2017. Sustainable and efficient data collection from WSNs to cloud. IEEE Trans. Sustain. Comput. 1, https://doi.org/10.1109/TSUSC.2017.2690301.

Yang, K., Jia, X., 2013. An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE Trans. Parallel Distr. Syst. 24, 1717–1726, https://doi.org/10.1109/TPDS.2012.278.

Yu, Y., Au, M.H., Ateniese, G., Huang, X., Susilo, W., Dai, Y., Min, G., 2017. Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage. IEEE Trans. Inf. Forensics Secur. 12, 767–778, https://doi.org/10.1109/TIFS.2016.2615853.

Yuan, J., Yu, S., 2015. Public integrity auditing for dynamic data sharing with multiuser modification. IEEE Trans. Inf. Forensics Secur. 10, 1717–1726, https://doi.org/10.1109/TIFS.2015.2423264.

Zhang, K., Liang, X., Lu, R., Yang, K., Shen, X.S., 2015. Exploiting mobile social behaviors for sybil detection. In: 2015 IEEE Conference on Computer Communication (INFOCOM '15), pp. 271–279, https://doi.org/10.1109/INFOCOM.2015.7218391.

Zhang, T., 2016. Fog Boosts Capabilities to Add More Things Securely to the Internet. http://blogs.cisco.com/innovation/fog-boosts-capabilities-to-add-more-things-securely-to-the-internet.

Zhu, Y., Ahn, G.J., Hu, H., Yau, S.S., An, H.G., Hu, C.J., 2013. Dynamic audit services for outsourced storages in clouds. IEEE Trans. Serv. Comput. 6, 227–238, https://doi.org/10.1109/TSC.2011.51.