Hindawi
Security and Communication Networks
Volume 2018, Article ID 9081814, 12 pages
https://doi.org/10.1155/2018/9081814

*Research Article*

# Secure Deduplication Based on Rabin Fingerprinting over Wireless Sensing Data in Cloud Computing

**Yinghui Zhang**,[1,2,3] **Haonan Su**,[1] **Menglei Yang**,[1] **Dong Zheng**,[1,3] **Fang Ren**,[1] **and Qinglan Zhao**[1]

[1]*National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China*
[2]*State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China*
[3]*Westone Cryptologic Research Center, Beijing 100070, China*

Correspondence should be addressed to Yinghui Zhang; yhzhaang@163.com and Dong Zheng; zhengdong@xupt.edu.cn

Received 9 June 2018; Accepted 12 August 2018; Published 6 September 2018

Academic Editor: Lianyong Qi



The rapid advancements in the Internet of Things (IoT) and cloud computing technologies have significantly promoted the collection and sharing of various data. In order to reduce the communication cost and the storage overhead, it is necessary to exploit data deduplication mechanisms. However, existing data deduplication technologies still suffer security and efficiency drawbacks. In this paper, we propose two secure data deduplication schemes based on Rabin fingerprinting over wireless sensing data in cloud computing. The first scheme is based on deterministic tags and the other one adopts random tags. The proposed schemes realize data deduplication before the data is outsourced to the cloud storage server, and hence both the communication cost and the computation cost are reduced. In particular, variable-size block-level deduplication is enabled based on the technique of Rabin fingerprinting which generates data blocks based on the content of the data. Before outsourcing data to the cloud, users encrypt the data based on convergent encryption technologies, which protects the data from being accessed by unauthorized users. Our security analysis shows that the proposed schemes are secure against offline brute-force dictionary attacks. In addition, the random tag makes the second scheme more reliable. Extensive experimental results indicate that the proposed data deduplication schemes are efficient in terms of the deduplication rate, the system operation time, and the tag generation time.

## 1. Introduction

The wireless sensor network (WSN) is an ad hoc network composed of a large number of sensors, and the sensors communicate with each other over a wireless channel in a multihop manner [1–5]. Sensors are usually a low-cost, simple device with limited computing power and working batteries, which have the ability to collect, process, and transfer data. With the rapid development of Internet of Things (IoT) and cloud computing technologies, WSN has found many promising applications. As an extension to the cloud computing paradigm, fog computing makes it possible to execute the IoT applications in the network of edge. Xu et al. [6] proposed a dynamic resource allocation method for load balancing in fog environment. Cloud computing [7, 8] supports distributed data storage and parallel processing and its data processing framework handles huge amounts of data in a local computer rather than requiring to transmit these data remotely [9–11]. We know that cloud storage technology is the most common and most popular cloud computing service today. The extensive application of cloud storage motivates enterprises and organizations to outsource data storage to third-party cloud providers [12–16]. Zhang et al. [17] proposed a fine-grained access control system suitable for resource-constrained users in cloud computing. It is reported that the average size of backup data for a medium size enterprise is 285 TB and faces an annual growth rate of about 24-27%. According to the analysis report of IDC, personal user data has reached terabytes in 2006. From 2006 to 2010, global data volume continues to grow at a rate of 57% annually. In 2011, the global data volume has entered the era of ZB, and the total amount of data used globally exceeds 1.8

ZB. It is expected that the global data volume will reach 40 ZB by 2020 [18].

Data deduplication has been widely accepted as an effective technique to reduce workload and overhead of the cloud storage system [19–23]. Today's commercial cloud storage services, such as Dropbox, Google Drive, Bitcasa, Mozy, and Memopal, have been applied deduplication to save maintenance cost. However, the extensive application of data deduplication makes its security problems increasingly prominent [24, 25]. Compared with traditional information security, cloud storage security [26–28] mainly has two characteristics: users do not enjoy physical control over the data they upload to the cloud storage system and the same kind of physical resources is shared by multiple users. The confidentiality and integrity of data will be threatened. It is noted that cloud storage security has drawn many attentions [29, 30]. Xu et al. [31] proposed a cost and energy aware data placement method for privacy-aware applications over big data in hybrid cloud. Harnik et al. [32] pointed out that there were security vulnerabilities in the deduplication technology used by the provider. Douceur et al. [33] introduced convergent encryption (CE) that uses the hash value of the data itself as a secret key to solve the problem of contradiction between deduplication and confidentiality. Bellare et al. [34] defined a cryptographic primitive called message-locked encryption. Li et al. [35] implemented Dekey using the Ramp secret sharing scheme to manage the CE keys. Literature [21] pointed out that, in the data deduplication, simply using the hash value of the file represents the entire file, making the data deduplication process vulnerable to hacking, and the hash value is not confidential, and the attacker can obtain the entire file content by obtaining the hash value. Abadi et al. [36] proposed two schemes, including a completely random scheme and a deterministic scheme, which support the randomization of tags to ensure the security of the data deduplication system. In the schemes, CE directly uses the data fingerprint as the key derivation function and hence only achieves security for unpredictable data. In fact, offline brute-force dictionary attacks can be easily launched because of the determination of CE keys [37]. Moreover, current deduplication schemes [35, 37] directly deduplicate the encrypted data, which increases the computational overhead. In the future, it is possible to realize decentralized data deduplication schemes via blockchain technologies, which have been used to realize decentralized outsourcing computation [38, 39] and searchable encryption with two-side verifiability [40] in cloud computing.

Deduplication can be defined based on different granularities [41]: file-level deduplication and block-level deduplication (fine-grained fixed-size or variable-size data block). File-level deduplication is the easiest but inefficient method. Fixed-size block-level deduplication refers to blocking the file into fine-grained fixed-size (such as 4MB, 512KB) data blocks and then deleting the duplicate blocks [42]. However, it is difficult for fixed-size block-level deduplication to deal with the situation of insertion of data in the file. Abadi et al. [36] propose a completely random scheme that avoids deterministic messages to generate tags directly and better guarantees the security of the data deduplication process. On the basis of [36], Jiang et al. [43] added static data deduplication decision trees and dynamic data deduplication decision trees and optimized duplicate detection operations. However, most of previous schemes realize data deduplication after the data is encrypted by users, and hence the computation and communication efficiencies remain to be improved. In [44], the authors proposed a data deduplication scheme based on Rabin fingerprinting, which is a preliminary version of the work given in Section 4.2 of this paper. In this paper, we significantly revise the preliminary scheme [44] and add more technical details as compared to the preliminary abstract [44]. First, we add Section 3 to describe a system architecture of secure deduplication based on Rabin fingerprinting over wireless sensing data in cloud computing. Second, we improve the basic construction to support randomized tags and provide detailed procedures of data deduplication using randomized tags in Section 4.3. Third, we present security analysis of both schemes in Section 5 and do extensive experiments to evaluate the proposed deduplication schemes in Section 6.

*Our Contribution*. The contributions of this paper can be summarized as follows. In order to tackle the security and efficiency drawbacks in the existing data deduplication technologies, we propose two secure data deduplication schemes based on Rabin fingerprinting over wireless sensing data in cloud computing. The first scheme is based on deterministic tags and the other one adopts random tags. Note that the randomized tag achieves more reliable security guarantees than the deterministic tag. In order to reduce the communication cost and the computation cost, data deduplication in the proposed schemes is realized before the data is outsourced to the cloud storage server. For the sake of practicability, we realize variable-size block-level deduplication of the data, which is enabled based on the technique of Rabin fingerprinting. In order to protect the outsourcing data from being accessed by unauthorized users, the data is encrypted by users based on convergent encryption technologies before outsourcing data to the cloud. Our security analysis shows that the proposed schemes are secure against both external attacks and internal attacks. Extensive experimental results indicate that the proposed data deduplication schemes are efficient in terms of the deduplication rate, the system operation time, and the tag generation time.

*Organization*. The rest of this paper is organized as follows. Notations and cryptographic backgrounds are reviewed in Section 2. The system model, the threat model and security requirements of a secure deduplication scheme are described in Section 3. We present the proposed two data deduplication schemes in Section 4. Section 5 gives the security analysis of the proposed schemes and Section 6 shows the performance evaluation. Finally, our concluding remarks are made in Section 7.

## 2. Preliminaries

In this section, we first explain notations used throughout this paper and then simply review some cryptographic

Table 1: Notation description.

| Notation | Meaning |
|---|---|
| $q$ | A prime. |
| $s \in_R S$ | $s$ is randomly chosen from the set $S$. |
| $g$ | A generator of a cyclic group of order $q$. |
| $(K_{\text{pub}}, K_{\text{pri}})$ | The public and secret key pair of a user. |
| $B_i$ | A data block. |
| $f(B_i)$ | The Rabin fingerprinting of $B_i$. |
| $K_i$ | The convergent key corresponding to $B_i$. |
| $C_i$ | The ciphertext corresponding to $B_i$. |
| $\tau_i$ | The random tag corresponding to $B_i$. |
| $\mathsf{B}(X)$ | The bitwise exclusive of $X$. |

backgrounds involved in the proposed data deduplication schemes.

*2.1. Notations.* In Table 1, we list notations mainly used in the description of the proposed data deduplication schemes.

*2.2. Rabin Fingerprinting.* The technique of Rabin fingerprinting is widely used for quick comparison and recognition of duplicate data. It is based on arithmetic modulo an irreducible polynomial over $\mathbb{Z}_2$ [45]. Let $S = [a_1, a_2, \ldots, a_n]$ be a bit string. We define a polynomial $S(t)$ of degree $n-1$ over $\mathbb{Z}_2$ as

$$S(t) = a_1 t^{n-1} + a_2 t^{n-2} + \cdots + a_{n-1} t + a_n. \quad (1)$$

Let $p(t) = b_1 t^k + b_2 t^{k-1} + \cdots + b_k t + b_{k+1}$ be an irreducible polynomial of degree $k$ over $\mathbb{Z}_2$. Given a fixed $p(t)$, the Rabin fingerprinting of $S(t)$ is defined as the polynomial $r(t) = S(t) \bmod p(t)$. The computation of Rabin fingerprinting is illustrated in Figure 1, where $[X_1, X_2, \ldots, X_\omega, X_{\omega+1}, X_{\omega+2}, \ldots]$ is a continuous string and each character $X_i$ is a tuple of 8 bits.

Note that a sliding window of width $\omega$ is used. Assume the starting point is $X_i$ which is represented by a polynomial $X_i(t)$; thus the Rabin fingerprinting value of the string $[X_i, X_{i+1}, \ldots, X_{i+\omega-1}]$ in the window is

$$r_i(t) = \left( \sum_{j=1}^{\omega} X_{i+j-1}(t)\, t^{8\omega-j} \right) \bmod p(t). \quad (2)$$

When the window slides forward 8 bits, $X_{i+1}$ becomes the starting point and then the Rabin fingerprinting value of the string $[X_{i+1}, X_{i+2}, \ldots, X_{i+\omega}]$ in the window is

$$r_{i+1}(t) = \left( \sum_{j=1}^{\omega} X_{i+j}(t)\, t^{8\omega-j} \right) \bmod p(t). \quad (3)$$

In fact, the Rabin fingerprinting algorithm computes a rolling checksum of the data [46]. The window of the data is configurable, but it is typically a few dozen bytes long. The Rabin module will read through a file and let the window slide over the data. When a byte is read, the fingerprint is recalculated. If the fingerprint is a special value, the Rabin module considers the corresponding window position to be a boundary. The data preceding this window position is taken to be a "block" of the file. For $1 \le i \le n$, let $B_i$ be a "block," and the fingerprint of the data block $B_i$ is defined as.

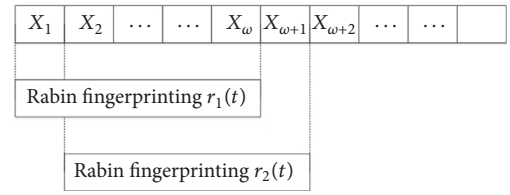


Figure 1: The computation of Rabin fingerprinting.

*2.3. Proof of Ownership.* A proof of ownership (PoW) protocol [47] enables a client to prove to the server that they own a given file. The server can derive a small metadata $T(M)$ from the data $M$. To prove the ownership of the data $M$, the user needs to send $T'$ and run a proof algorithm with the sever. Its ownership is accepted if and only if $T' = T(M)$ and the proof is correct [48].

*2.4. Convergent Encryption.* The notion of convergent encryption was proposed by Douceur et al. [33]. In order to ensure the confidentiality of outsourcing data in the data deduplication process, users first encrypt data and then upload ciphertexts. In practice, if traditional encryption mechanisms are adopted, different users have diverse encryption keys, which leads to that the same file will be encrypted to different ciphertexts by diverse users. This property poses a serious challenge to data deduplication form the point of efficiency. In convergent encryption, the key is derived from the outsourcing data, and hence the same data corresponds to the same ciphertext even if users are different. Therefore, CE makes it possible to realize secure data deduplication in ciphertexts. Figure 2 illustrates the process of a convergent encryption. A convergent encryption scheme consists of the following algorithms:

(i) $\mathsf{KenGen}_{\text{CE}}(M) \longrightarrow K$. The key generation algorithm generates a convergent key $K$ based on data $M$. For a secure use of convergent encryption, the convergent key should be unpredictable, which can be realized by introducing randomness based on the message authentication code (MAC). MAC is also known as the keyed hash function. It is a value obtained based on a secret key and a message digest, which is usually used to data source authentication and integrity checking. A MAC is defined as below.

  (a) $\mathsf{Hash}(M) \longrightarrow H_b$ is a hash algorithm, such as SHA-1 and SHA-256, which takes as input the message $M$ and outputs the hash value.

  (b) $\mathsf{keyHmac}(secret, H_b) \longrightarrow K$ is a message authentication code that takes as inputs the hash value $H_b$ and a random parameter *secret* and outputs a randomized convergent key $K$.

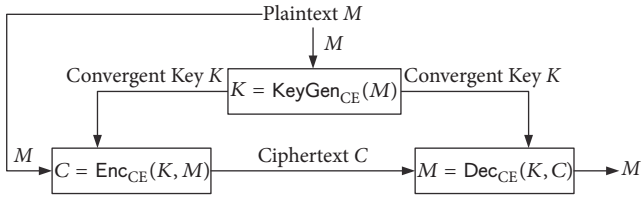


Figure 2: The convergent encryption process.

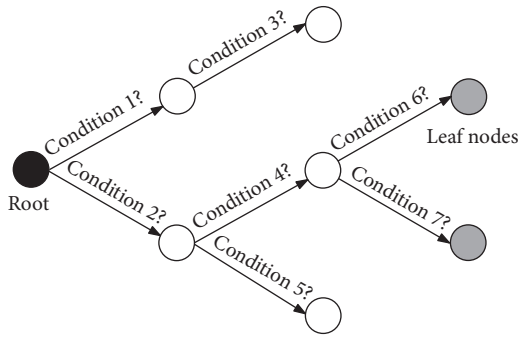


Figure 3: An example of decision tree.

(ii) $\mathsf{Enc}_{\mathrm{CE}}(K, M) \longrightarrow C$. It is a symmetric encryption algorithm that takes the convergent key $K$ and the data $M$ as inputs and outputs a ciphertext $C$.

(iii) $\mathsf{Dec}_{\mathrm{CE}}(K, C) \longrightarrow M$. It is the corresponding decryption algorithm that takes the convergent key $K$ and the ciphertext $C$ as inputs and outputs the original data $M$.

(iv) $\mathsf{TagGen}_{\mathrm{CE}}(M) \longrightarrow T_M$. The tag generation algorithm maps the original data $M$ to a tag $T_M$. Essentially, the Rabin fingerprinting of data is used as the tag in the deterministic tag based scheme and is used to generate tags for the random tag based scheme.

*2.5. Decision Trees.* As a predictive model, a decision tree is a tree-like structure, in which each internal node denotes a test on an attribute, each branch represents the test output, and each leaf node means a category. For example, as shown in Figure 3, a decision tree consists of nodes and branches. Typically, a decision tree begins with the root node and branches connect the nodes. A branch that originates from a decision node is called a decision branch. Note that different conditions are associated with different branches. A leaf node acts as a termination node, which indicates the final outcome of the branch.

## 3. Models and Security Goals

In this section, we first introduce the system model and then describe the threat model and security goals.

*3.1. System Model.* The system model is illustrated in Figure 4, in which three entities are involved, including a management server (MS), users, and a cloud storage server (CSS). In the model, users outsource their data to CSS and access the data later with the help of MS, while keeping the ability of data deduplication. The details are described as follows.

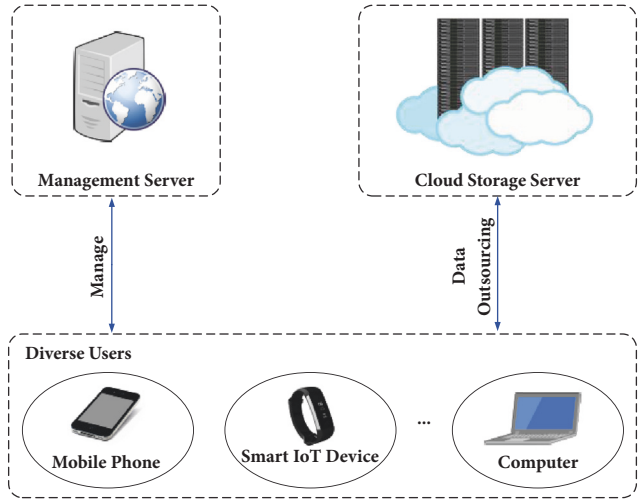


Figure 4: The system model.

*(i) MS.* It is trusted by users and manages secret keys and users' information. MS introduces a random secret parameter to generate randomized convergent keys for users.

*(ii) Users.* Users can compute the block fingerprints before data deduplication. They encrypt data and then upload ciphertexts to CSS. For recovering the data, they decrypt the corresponding ciphertext from CSS.

*(iii) CSS.* It is honest but curious and provides data storage service to users. It stores and manages user's unique data copies in the form of ciphertexts. In the subsequent random tag based deduplication scheme, CSS checks duplicate data based on a decision tree.

*3.2. Threat Model and Security Goals.* We consider both external attackers and internal attackers for the security of outsourcing data storage with data deduplication. For one thing, in the public channels, the external attackers are able to achieve partial of information on the data. An external attacker can access CSS by disguising as a legitimate user. For another, the internal attackers are honest but curious. They will follow the procedures of the proposed scheme and try to get confidential information as much as possible. The goal of the internal attackers is to obtain the contents of the data from CSS and obtain the randomized convergent keys from MS.

Considering the above threat model, we specify the following security goals. First, we need to ensure that the semantic security of encrypted data blocks. This requirement has been formalized in [49]. Therefore, the adversary does not have the ownership of the data because there is no convergent key to encrypt. Second, the convergent keys should be kept secure. The goal of the attackers is to get the other users' keys and the data block ciphertexts. We aim to guarantee the security of the keys' transmission and storage. Neither

external attackers nor internal attackers can obtain other convergent keys.

## 4. Data Deduplication Schemes Based on Rabin Fingerprinting

In this section, we propose two data deduplication schemes based on Rabin fingerprinting, including a deterministic tag based scheme and a random tag based scheme. In each scheme, three phases, system setup, file uploading, and file downloading, are performed for data outsourcing storage with deduplication. The proposed deduplication schemes perform block-level data deduplication before users' data encryption, in which the file blocks are generated based on Rabin fingerprinting.

*4.1. Overview of Our Schemes.* In the first scheme, the outsourcing data is first divided into many data blocks based on the Rabin fingerprinting technique. For each data block, a deterministic tag is generated based a hash function. With the tag, the cloud storage server can check whether the corresponding data block has already existed. If it exists, the user proves to the cloud server that it indeed has the ownership of the data block. Otherwise, the user encrypts the data block and uploads the generated ciphertext to the cloud server, in which the ciphertext is based on a convergent encryption and the convergent key is generated by the management server. The security of data in the deduplication process is ensured based on encryption techniques, and the convergent keys are also effectively managed. However, deterministic tags fail to meet the standard confidentiality requirement, such as semantic security. To be specific, if the plaintext can be listed, the attacker can learn the content of the plaintext by computing the tags and comparing the ciphertexts. If the tag is unpredictable, the above security drawback can be avoided. In the second scheme, the tag is randomly generated by the management server. The new scheme can support randomized tags and also allows decision tree based data duplicate detection. The decision tree supports the deletion and updating without needing expensive bilinear pairing operations. The randomized tags sacrifice efficiency to some extent but provide more reliable protection for data confidentiality in data deduplication systems.

*4.2. Data Deduplication with Deterministic Tags*

*4.2.1. System Setup.* In the system setup phase, necessary parameters are generated based on the following procedures:

(S1) Given a security parameter $1^\lambda$, MS specifies a convergent encryption scheme ($\mathsf{KeyGen}_{\mathrm{CE}}$, $\mathsf{Enc}_{\mathrm{CE}}$, $\mathsf{Dec}_{\mathrm{CE}}$, and $\mathsf{TagGen}_{\mathrm{CE}}$), an asymmetric encryption scheme ($\mathsf{KeyGen}_{\mathrm{AE}}$, $\mathsf{Enc}_{\mathrm{AE}}$, and $\mathsf{Dec}_{\mathrm{AE}}$), and a PoW algorithm. MS runs $\mathsf{KeyGen}_{\mathrm{AE}}$ to generate an asymmetric public and secret key pair ($K_{\mathrm{pub}}$, $K_{\mathrm{pri}}$) for each user. Note that $\mathsf{KeyGen}_{\mathrm{CE}}$ is realized based on keyHmac and $\mathsf{TagGen}_{\mathrm{CE}}$ is computed based on the Rabin fingerprinting.

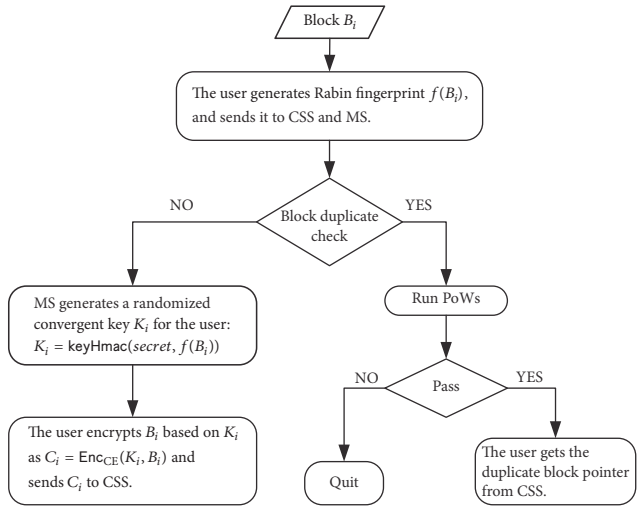


Figure 5: The uploading phase of deterministic tag based data deduplication.

(S2) The CSS initializes two types of storage systems: a fast storage system for efficient detection of duplicate data tags and a file storage system for storing encrypted outsourcing data.

(S3) MS initializes its local storage system for storing users' metadata and randomized convergent keys.

*4.2.2. File Uploading.* The uploading phase is shown in Figure 5. Suppose that a user uploads a file $F$ and then performs the block-level deduplication below:

(S1) The user sends a file-backup request to MS, including its authentication information. Then, MS performs an identity authentication. If passed, the following steps are performed.

(S2) Based on the Rabin fingerprinting technique, the user divides $F$ into a set of blocks denoted by $\{B_i\}_{1\le i\le n}$. The user computes each block fingerprint $f(B_i)$ and sends $f(B_i)$ as tags to CSS for duplicate checking.

(S3) In addition, the fingerprints $\{f(B_i)\}$ are sent to MS for generating convergent keys later.

(S4) Once the data block fingerprints $\{f(B_i)\}$ are received, CSS computes the data block signal vector $\sigma_B$ as follows:

(i) For each $i$, if an existing block fingerprint matches $f(B_i)$, CSS sets $\sigma_B[i] = 1$ to indicate "block duplicate."

(ii) Otherwise, CSS sets $\sigma_B[i] = 0$ to indicate "no block duplicate." CSS stores $f(B_i)$ into the fast storage system.

After the data deduplication is fulfilled, CSS returns the signal vector $\sigma_B$ to the user.

(S5) After receiving $\sigma_B$, the user checks if $\sigma_B[i] = 1$. If it is, the user runs a PoW algorithm to prove to CSS

that it owns the data block $B_i$. If CSS accepts the proof, it directly returns the corresponding pointer of $B_i$ to the user. At the same time, the user stores the block pointer of $B_i$ which is not needed to upload. In the other cases, the protocol is terminated and the involved entities quit the protocol.

(S6) Otherwise, the user sends $\sigma_B$ to MS. Upon receiving $\sigma_B$, MS checks if $\sigma_B[i] = 0$. If it is, MS generates the convergent key $K_i = \mathsf{keyHmac}(secret, f(B_i))$, where *secret* is a randomly chosen parameter. MS sends the randomized convergent key $K_i$ corresponding to the nonduplicate block to the user. The user computes a ciphertext $C_i = \mathsf{Enc}_{\mathrm{CE}}(K_i, B_i)$ and uploads $C_i$ to CSS.

*4.2.3. File Downloading.* Suppose that a user intends to download a file $F$. The user first sends a downloading request to MS, including its authentication information. If the authentication is successfully verified, the following procedures are performed:

(S1) MS encrypts the randomized convergent key $K_i$ by computing $C_{k_i} = \mathsf{Enc}_{\mathrm{AE}}(K_{\mathrm{pub}}, K_i)$, which is then sent to the user.

(S2) Upon receiving the ciphertext $C_{k_i}$, the user decrypts it based on its secret key $K_{\mathrm{pri}}$ to get the randomized convergent key $K_i$, that is, $K_i = \mathsf{Dec}_{\mathrm{AE}}(K_{\mathrm{pri}}, C_{k_i})$. Subsequently, the user obtains the encrypted data block $\{C_i\}$ from CSS.

(S3) The user decrypts the corresponding ciphertext $C_i$ by computing $B_i = \mathsf{Dec}_{\mathrm{CE}}(K_i, C_i)$, based on $K_i$, and then restores the file $F$.

### 4.3. Data Deduplication with Randomized Tags

*4.3.1. System Setup.* The details are the same to those in the deterministic tag based scheme. Besides, MS specifies a cyclic group of prime order $q$ with generator $g$.

*4.3.2. File Uploading.* Suppose that a user intends to outsource the file $F$. The tag corresponding to the data block $B_i$ is $\tau_i = (g^{r_i}, g^{r_i f(B_i)}, s_i)$, where $r_i$ is randomly chosen from $\mathbb{Z}_q^*$ and $f(B_i)$ is the data fingerprint. The value of $s_i$ can be 0 or 1. If $s_i = 1$, it means the corresponding data block of the tag in the decision tree has not been deleted. When the data block is deleted, $s_i$ is set to be 0, which means there is no corresponding data block in CSS. The data uploading process is illustrated in Figure 6.

(S1) The user sends a file-backup request to MS, including its authentication information. Then, MS performs an identity authentication. If passed, the following steps are performed:

(S2) Based on the Rabin fingerprinting technique, the user divides $F$ into a set of blocks denoted by $\{B_i\}_{1 \le i \le n}$. The user computes each block fingerprint $f(B_i)$ and sends $f(B_i)$ to MS.

(S3) Upon receiving the data backup request, CSS first iterates through the tag nodes in the order of the decision tree. If $s_i = 0$, it traverses the next node's tag until $s_i = 1$ or a leaf node. If $s_i = 1$, CSS returns the tag $\tau_i = (g^{r_i}, g^{r_i f(B_i)}, s_i)$ to the user. Note that the root node tag of the decision tree is $\tau_0 = (g^{r_0}, g^{r_0 f(B_0)}, s_0)$ and $s_i$ has a default value 1.

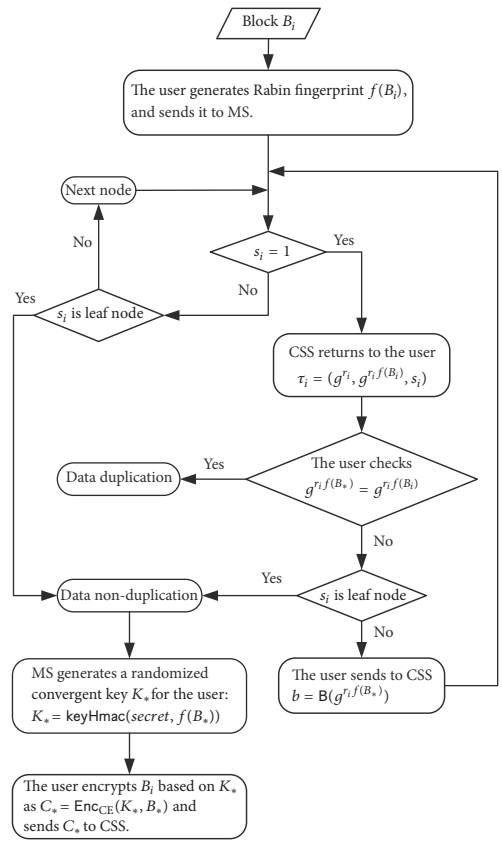


Figure 6: The uploading phase of random tag based data deduplication.

(S4) Once the user receives the tag sent by CSS, the user calculates $g^{r_i f(B_*)}$ and verifies that $g^{r_i f(B_*)}$ is equal to $g^{r_i f(B_i)}$.

(i) If $g^{r_i f(B_*)} = g^{r_i f(B_i)}$, the user sends "data duplication" to CSS and skips to the step (S6).

(ii) Otherwise, the user calculates $b = \mathsf{B}(g^{r_i f(B_*)})$ and sends it to CSS.

(S5) The server moves the pointer to the next node in the decision tree based on the result of $b = \mathsf{B}(g^{r_i f(B_*)})$.

(i) If $b = 0$, CSS will move the pointer to the left node.

(ii) If $b = 1$, the pointer will be moved to the right node and the above step (S3) is performed again.

If the decision tree pointer has not found a duplicate node after moving to the leaf node, CSS will send "data non-duplication" instruction to the user and skips to the following step (S7).

(S6) Once the user receives a "data duplication" instruction for a block $B_*$, it runs a PoW protocol with CSS to prove its ownership of the block. If passed, then CSS will return to the user a pointer to the duplicate data block $B_*$. The user then stores the pointer and the data block $B_*$ does not need to be uploaded.

(S7) Once the user receives the "data non-duplication" associated with the data block $B_*$, MS generates the convergent key $K_* = \mathsf{keyHmac}(secret, f(B_*))$, where *secret* is a randomly chosen parameter, and sends the randomized convergent key $K_*$ to the user. The user will run the encryption algorithm $C_* = \mathsf{Enc}_{\mathrm{CE}}(K_*, B_*)$ to compute the ciphertext $C_*$ and upload it to CSS. At the same time, the user chooses $r_* \in_R \mathbb{Z}_q^*$, generates a corresponding tag $\tau_* = (g^{r_*}, g^{r_* f(B_*)}, s_*)$, and sends the tag to CSS.

(S8) Upon receiving the tag $(g^{r_*}, g^{r_* f(B_*)}, s_*)$ of the block $B_*$ from the user, CSS computes $b = \mathsf{B}(g^{r_i f(B_*)})$. If $b = 0$, the tag $(g^{r_*}, g^{r_* f(B_*)}, s_*)$ will cover the left node with $s_* = 0$, or be placed on the left leaf node. If $b = 1$, the tag $(g^{r_*}, g^{r_* f(B_*)}, s_*)$ will cover the right node with $s_* = 0$ or be placed on the right leaf node.

*4.3.3. File Downloading.* Suppose that a user intends to download a file $F$. The user first sends a downloading request to MS, including its authentication information. If the authentication is successfully verified, the following procedures are performed:

(S1) MS encrypts the randomized convergent key $K_i$ by computing $C_{k_i} = \mathsf{Enc}_{\mathrm{AE}}(K_{\mathrm{pub}}, K_i)$, which is then sent to the user.

(S2) Upon receiving the ciphertext $C_{k_i}$, the user decrypts it based on its secret key $K_{\mathrm{pri}}$ to get the randomized convergent key $K_i$, that is, $K_i = \mathsf{Dec}_{\mathrm{AE}}(K_{\mathrm{pri}}, C_{k_i})$. Subsequently, the user obtains the encrypted data block $\{C_i\}$ from CSS.

(S3) The user decrypts the corresponding ciphertext $C_i$ by computing $B_i = \mathsf{Dec}_{\mathrm{CE}}(K_i, C_i)$, based on $K_i$, and then restores the file $F$.

## 5. Security Analysis of the Proposed Schemes

The proposed two data deduplication schemes differ in the tags. The first scheme adopts deterministic tags and the second scheme uses random tags. The involvement of random parameters in the tag generation makes the second scheme more secure. In the following, we only show that the deterministic tag based scheme is secure against both external attacks and internal attacks.

*5.1. Security against External Attacks.* In data deduplication systems, external attackers must be prevented from accessing data. For instance, the transmitted data between the user and CSS may be obtained by an external attacker. After selecting the range of a dictionary, the attacker can obtain data corresponding to metadata by the way of brute-force dictionary attack. In particular, an external attacker may maliciously modify and destroy users' transmitted data in order to compromise both the integrity and the availability of the data. In the proposed deduplication scheme, random information is added to the convergent key by MS, which randomizes the convergent key and alleviate the key compromise risk. The randomization of the convergent key makes offline brute-force attack very difficult. Because each user first encrypts outsourcing data and then transmits data ciphertexts in the system, it is impossible for external attackers to get the original data without needing the relevant key.

*5.2. Security against Internal Attacks.* In order to issue cross-border operations, attackers often try to hide their own identities. For example, an attacker may disguise as other legitimate users to violate the privacy of other users. To prevent the internal attackers, the secure deduplication system realizes the identity authentication when a user initially communicate with MS which stores and manages the convergent keys to prevent unauthorized information read. At the side of CSS, if a user aims to access a file, a PoW protocol is required to be performed between the user and CSS. The user can prove to CSS its ownership of the file. The proposed deduplication scheme can effectively prevent attackers from accessing any files and keys beyond their ownership. In the random tag based scheme, besides the security of the first scheme, it also avoids the use of deterministic tags during duplicate data detection. Accordingly, even if an attacker obtains a tag, the randomness of the tag makes it possible to obtain the corresponding convergent key, which further improves the system security.

## 6. Performance Evaluation

In this section, we evaluate the performance of the proposed Rabin fingerprinting based data deduplication systems. We also compare the trivial deduplication rate of the fixed-size block scheme and our Rabin fingerprinting based schemes.

*6.1. Simulation Environment.* The hardware used in the simulation is a 64-bit Lenovo 80ER laptop with Windows 7 Home Basic operating system, and its CPU is Intel(R) Core(TM) i5-5300U CPU @2.30GHz. The simulation code is written with Java language by using the MyEclipse development platform. In our experiments, the data samples are 5400 different journal articles from the China national knowledge infrastructure, and they are about 15.9 GB in size.

*6.2. Experimental Results and Analysis*

*6.2.1. The Optimized Sliding Step Size.* In this section, we aim to find the optimized sliding step size of the data deduplication scheme based on the Rabin fingerprinting. The optimized sliding step size enables a better performance of the data deduplication system. Specifically, given a fixed-size data set, we set the upper bound of the data block size as 8 KB and the sliding window size of the Rabin fingerprinting as 64 KB. Then, when the window sliding step size varies from 1 B to 20 B, we test the running time and the

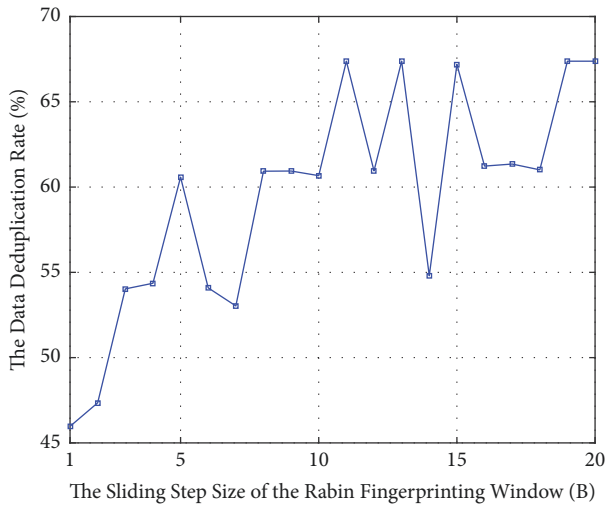


Figure 7: The variation of the deduplication rate with the sliding step size.

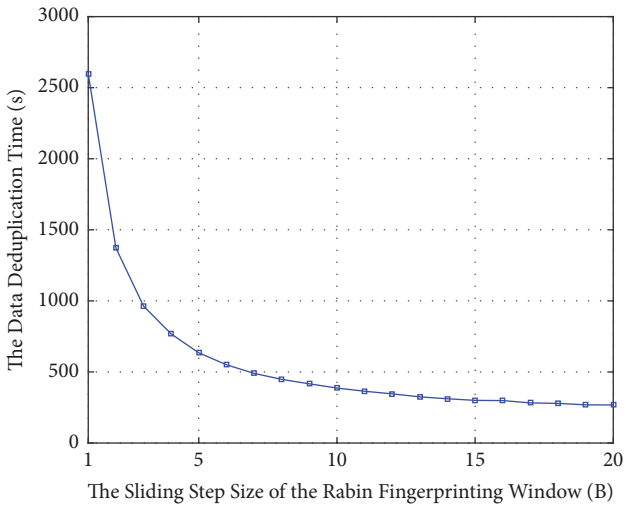


Figure 8: The variation of the data deduplication time with the sliding step size.

deduplication rate of the Rabin fingerprinting, respectively. Note that the data deduplication rate is defined as the ratio of the remaining nonduplicate data size after data deduplication to the total data size. The smaller the ratio, the better the data deduplication effect.

Figure 7 illustrates the variation of the data deduplication rate with the sliding step size of the window. Figure 8 shows the variation of the data deduplication time with the sliding step size of the window. We can see from Figure 7 that the deduplication rate has the optimal value when the Rabin fingerprinting has a window sliding step of 1 B and more than 50% of the duplicate data is removed. In this case, however, the deduplication time is the longest as shown in Figure 8. As the sliding step size increases, the data deduplication rate fluctuates between fixed values and it tends to be steady between several given sliding step size. When the window sliding step size is 1 B, the time of the deduplication

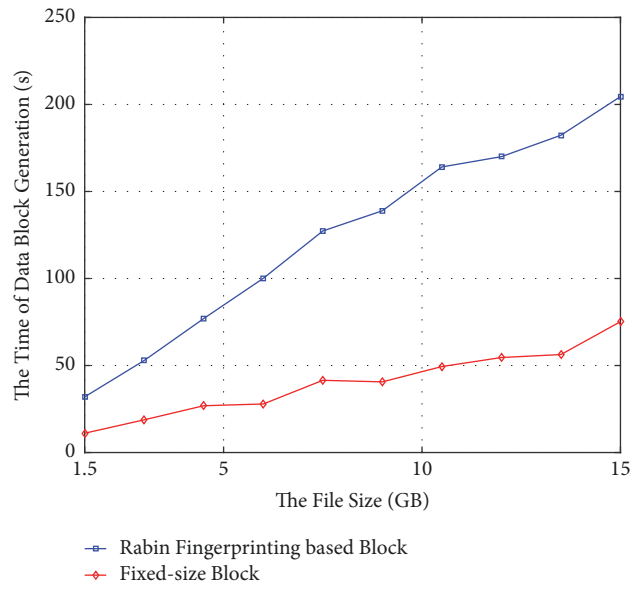


Figure 9: The time of data block generation with the file size.

based on Rabin fingerprinting is the longest. The longer the sliding window moves, the less time it takes for data to be deduplicated. Generally, in order to ensure the effect of data deduplication based on Rabin fingerprinting and reduce the system operation time, we exploit a sliding window of 64 KB and a sliding step size of 18 B in the following experiments.

*6.2.2. The Performance Comparison of Rabin Fingerprinting Based Scheme and Fixed-Size Block Scheme.* Figure 9 shows that the time for the data block generation varies with the file size. At the same time, we compare the block generation time of the fixed-size block deduplication scheme and the Rabin fingerprinting based scheme. Given the test file of the same size, the time required for the fixed-size block scheme is smaller than that based on the Rabin fingerprinting. Nevertheless, we will show that the total system operation time of the Rabin fingerprinting based scheme is optimal, later. Figure 10 compares the time of data deduplication based on the Rabin fingerprinting and data deduplication based on fixed-size blocks. The deduplication time does not include the block generation time, and the performance of these two schemes is compared from the perspective of data deduplication. It can be seen from Figure 10 that the data deduplication time of both schemes increases with the file size. If the test file is given, the deduplication efficiency of the Rabin fingerprinting based scheme is obviously better than that of the fixed-size block scheme. The comparison reflects the advantage of the Rabin fingerprinting in data deduplication.

In the subsequent simulation, the fixed-size block algorithm is first used to divide the test files into fixed-size data blocks of sizes 4, 8, 16, 32, 64, 128, 256, 512, 1024, and 2048 KB, respectively. Then, the Rabin fingerprint algorithm is used to divide the same files into variable-size data blocks with the upper bound limit of sizes 4, 8, 16, 32, 64, 128, 256, 512, 1024,

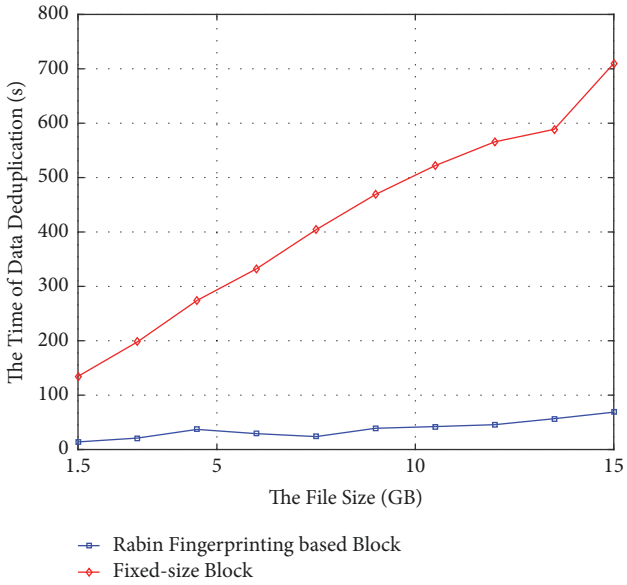


Figure 10: The time of data deduplication with the file size.

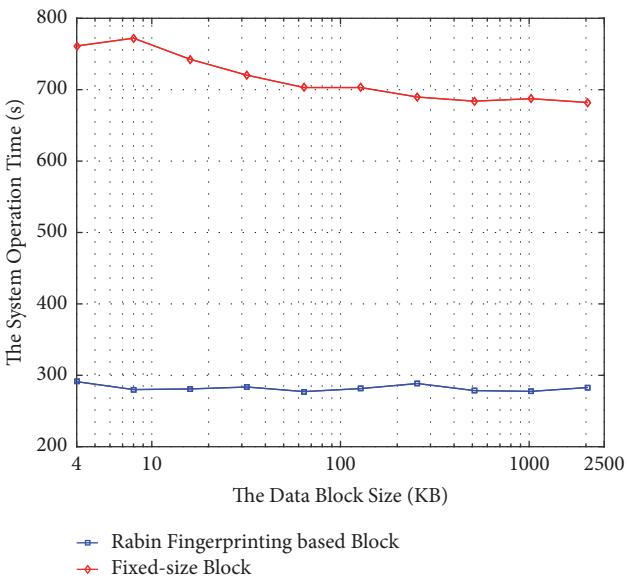


Figure 11: The system operation time with the block size.

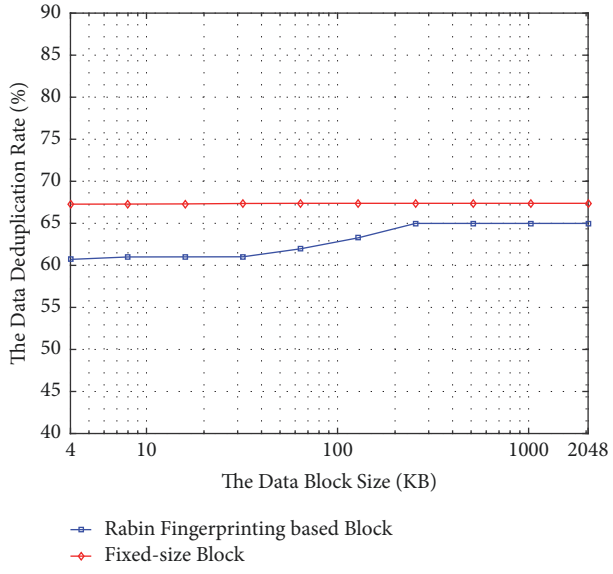


Figure 12: The data deduplication rate with the block size.

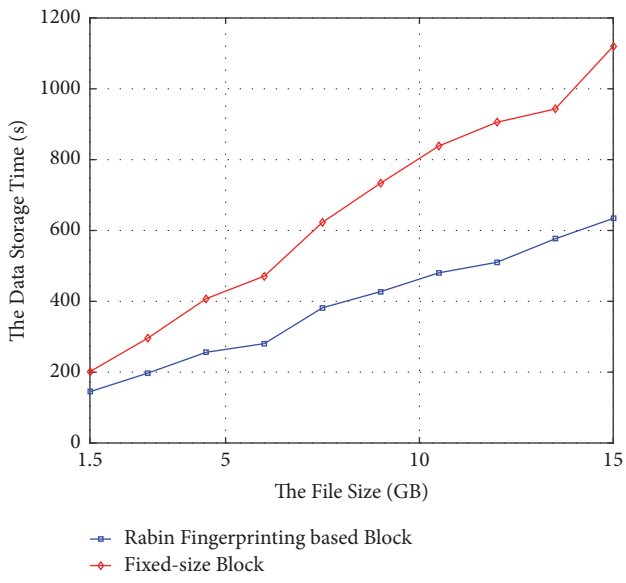


Figure 13: The data storage time with the file size.

and 2048KB, respectively. Finally, the data deduplication time and the deduplication rate are tested and compared.

Figure 11 shows the comparison of the total running time of the data deduplication system based on the Rabin fingerprinting and the fixed-size block-level data deduplication system. For the sake of clarity, the horizontal axis adopts a logarithmic scale. We can see that the performance of the data deduplication system based on the Rabin fingerprinting is better. Therefore, the use of the characteristics of duplicate data can be quickly found by using the Rabin fingerprinting, which makes the data deduplication more efficient. Figure 12 shows the data deduplication rate of the Rabin fingerprinting based scheme and the fixed-size block scheme. It can be seen that the duplicate data detection rate of the former is better than the latter. With the increase of the data block size, the deduplication rate becomes inferior. However, the data deduplication rate of the system based on the Rabin fingerprinting is always better than the fixed-size block scheme. Figure 13 is a comparison of the overall system data storage performance based on the Rabin fingerprinting and fixed-size block level, respectively. From this figure, we can see that the overall storage performance of the data deduplication scheme based on Rabin fingerprinting is better than that of the fixed-size block data deduplication system. And with the increase of the data volume of the file, the increase trend of the storage time of the fixed-size block-level data deduplication system is faster than that of the Rabin fingerprinting deduplication system.

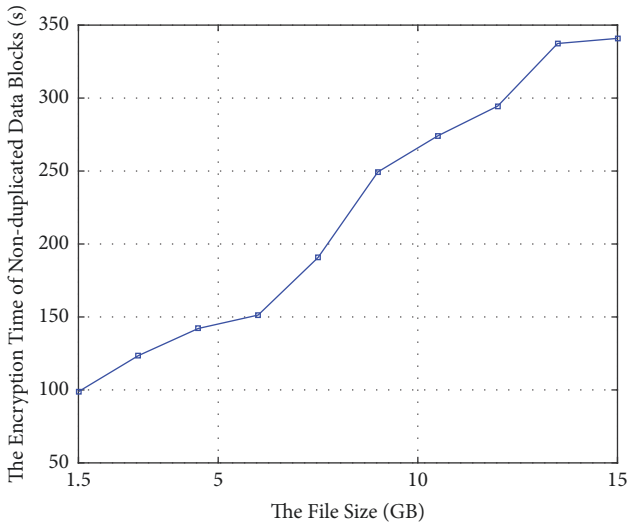


Figure 14: The encryption time of the nonduplicated data blocks with the file size.

*6.2.3. The Encryption Time of Nonduplicated Data Blocks.* Besides the deduplication performance, we also consider the cost of encryption. As shown in Figure 14, the encryption time of the nonduplicated data after deduplication operation increases with the file size. Based on Figure 10, we know that the fixed-size block-level data deduplication scheme will generate more data blocks, and hence the data block encryption time will be longer than that of the Rabin fingerprinting based scheme. In particular, the fixed-size block-level data deduplication scheme needs to encrypt data before performing data deduplication. Put another way, duplicate data is also encrypted, which further increases the encryption overhead of the system.

*6.2.4. The Performance Comparison of Deterministic Tags and Random Tags.* In the above analysis, the schemes are of deterministic tags. In the following, we test and analyze the performance of the data deduplication systems based on the Rabin fingerprint algorithm with deterministic tags and random tags. In Figure 15, we show the performance comparison of tag generation in the data deduplication scheme based on deterministic tags and random tags.

It can be seen from Figure 16 that the time for generating random tags is much longer than that of the deterministic tags. With the increase of the number of uploaded files, the time cost of generating random tags will also increase and its rising trend is obvious. In Figure 16, we compare the storage performance of the two types of schemes. As can be seen from Figure 16, the larger the number of uploaded files, the greater the total data deduplication time of the two deduplication schemes. Generally, the random tag based deduplication system is more secure and the deterministic tag based scheme is more efficient.

## 7. Conclusions and Future Work

In this paper, we proposed two secure data deduplication schemes based on Rabin fingerprinting. The schemes are realized, respectively, based on deterministic tags and random tags. In our schemes, data deduplication is enabled before the data is outsourced to the cloud storage server, and hence both the communication cost and the computation cost are reduced. In particular, we realized variable-size block-level deduplication by using Rabin fingerprinting. The data confidentiality is kept based on convergent encryption technologies. Our security analysis showed that the proposed schemes can resist offline brute-force dictionary attacks. Our simulation results indicated that the proposed schemes are practical in terms of the efficiency.

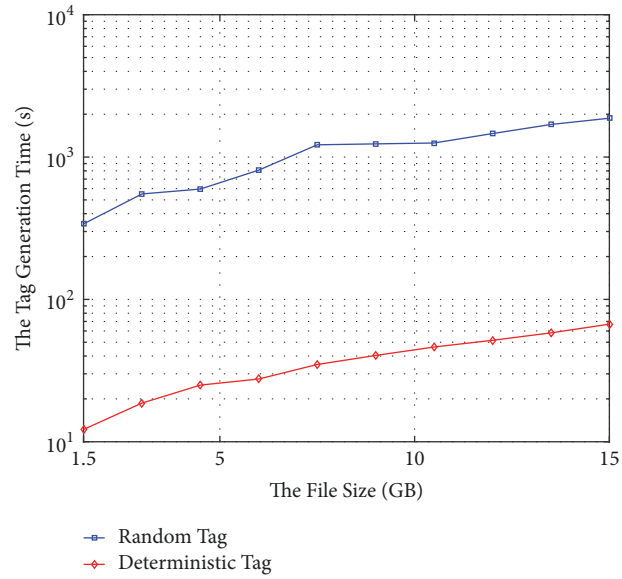


Figure 15: The tag generation time with the file size.

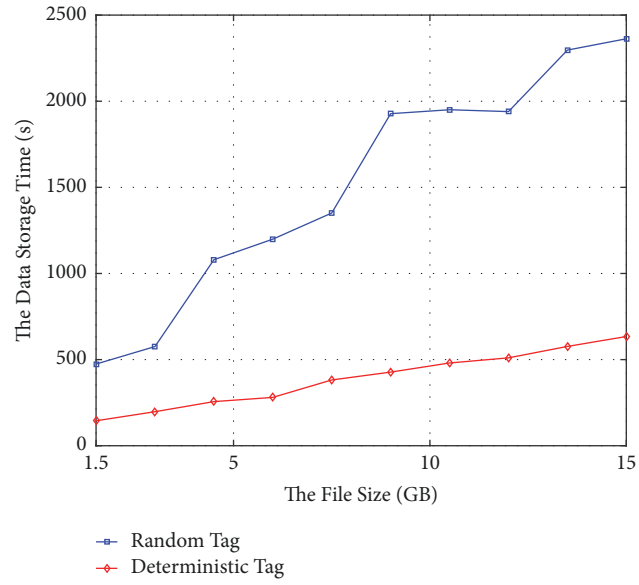


Figure 16: The storage cost comparison of random tag scheme and deterministic tag scheme.

In the future research, it would be interesting to design decentralized block-level data deduplication schemes with fine-grained access control.

## Data Availability

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

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work is supported by National Key R&D Program of China (no. 2017YFB0802000), the National Natural Science Foundation of China (nos. 61772418, 61472472, and 61402366), and the Natural Science Basic Research Plan in Shaanxi Province of China (nos. 2018JZ6001 and 2015JQ6236). Yinghui Zhang is supported by New Star Team of Xi'an University of Posts and Telecommunications (2016-02).

## References

[1] H. Huang, T. Gong, N. Ye, R. Wang, and Y. Dou, "Private and secured medical data transmission and analysis for wireless sensing healthcare system," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1227–1237, 2017.

[2] Y. Zhang, X. Chen, J. Li, and H. Li, "Generic construction for secure and efficient handoff authentication schemes in EAP-based wireless networks," *Computer Networks*, vol. 75, pp. 192–211, 2014.

[3] Q. Han, Y. Zhang, X. Chen, H. Li, and J. Quan, "Efficient and robust identity-based handoff authentication in wireless networks," in *Proceedings of the in International Conference on Network and System Security*, pp. 180–191, Springer, 2012.

[4] Y. Zhang, J. Li, D. Zheng, P. Li, and Y. Tian, "Privacy-preserving communication and power injection over vehicle networks and 5G smart grid slice," *Journal of Network and Computer Applications*, 2018, http://dx.doi.org/10.1016/j.jnca.2018.07.017.

[5] Y. H. Zhang, X. F. Chen, H. Li, and J. Cao, "Identity-based construction for secure and efficient handoff authentication schemes in wireless networks," *Security and Communication Networks*, vol. 5, no. 10, pp. 1121–1130, 2012.

[6] X. Xu, S. Fu, Q. Cai et al., "Dynamic resource allocation for load balancing in fog environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 6421607, 15 pages, 2018.

[7] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New publicly verifiable databases with efficient updates," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 546–556, 2015.

[8] Y. Zhang, A. Wu, and D. Zheng, "Efficient and privacy-aware attribute-based data sharing in mobile cloud computing," *Journal of Ambient Intelligence & Humanized Computing*, vol. 9, no. 4, pp. 1039–1048, 2018.

[9] Z. Wu, L. Tian, P. Li, T. Wu, M. Jiang, and C. Wu, "Generating stable biometric keys for flexible cloud computing authentication using finger vein," *Information Sciences*, vol. 434, pp. 431–447, 2016.

[10] L. Yang, Z. Han, Z. Huang, and J. Ma, "A remotely keyed file encryption scheme under mobile cloud computing," *Journal of Network and Computer Applications*, vol. 106, pp. 90–99, 2018.

[11] C. Xiang, C. Tang, Y. Cai, and Q. Xu, "Privacy-preserving face recognition with outsourced computation," *Security & Communication Networks*, vol. 20, no. 9, pp. 3735–3744, 2016.

[12] Y. Zhang, M. Yang, D. Zheng, P. Lang, A. Wu, and C. Chen, "Efficient and secure big data storage system with leakage resilience in cloud computing," *Soft Computing*, 2018, http://dx.doi *Soft Computing*, 2018, .org/10.1007/s00500-018-3435-z.

[13] Y. Zhang, P. Lang, D. Zheng, M. Yang, and R. Guo, "A secure and privacy-aware smart health system with secret key leakage resilience," *Security and Communication Networks*, vol. 2018, Article ID 7202598, pp. 1–13, 2018.

[14] C. Esposito, M. Ficco, F. Palmieri, and A. Castiglione, "Smart cloud storage service selection based on fuzzy logic, theory of evidence and game theory," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 8, pp. 2348–2362, 2016.

[15] Y. Zhang, D. Zheng, Q. Li, J. Li, and H. Li, "Online/offline unbounded multi-authority attribute-based encryption for data sharing in mobile cloud computing," *Security and Communication Networks*, vol. 9, no. 16, pp. 3688–3702, 2016.

[16] Z. Li, Y. Dai, G. Chen, and Y. Liu, "Toward network-level efficiency for cloud storage services," in *Content Distribution for Mobile Internet: A Cloud-based Approach*, pp. 167–196, Springer, 2016.

[17] Y. Zhang, D. Zheng, R. Guo, and Q. Zhao, "Fine-Grained Access Control Systems Suitable for Resource-Constrained Users in Cloud Computing," *Computing and Informatics*, vol. 37, no. 2, pp. 327–348, 2018.

[18] X. Chen, J. Li, J. Weng, J. Ma, and W. Lou, "Verifiable computation over large database with incremental updates," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 10, pp. 3184–3195, 2016.

[19] H. Kwon, C. Hahn, D. Koo, and J. Hur, "Scalable and reliable key management for secure deduplication in cloud storage," in *Proceedings of the IEEE International Conference on Cloud Computing*, pp. 391–398, 2017.

[20] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2130–2145, 2018.

[21] J. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet of Things Journal*, 2018, http://dx.doi.org/10 .1109/JIOT.2018.2842773.

[22] J. Xiong, Y. Zhang, X. Li, M. Lin, Z. Yao, and G. Liu, "RSE-PoW: a role symmetric encryption PoW scheme with authorized deduplication for multimedia data," *Mobile Networks and Applications*, vol. 23, no. 3, pp. 650–663, 2018.

[23] Y. Zhang, J. Shu, X. Liu, J. Li, and D. Zheng, "Comments on a large-scale concurrent data anonymous batch verification scheme for mobile healthcare crowd sensing," *IEEE Internet of Things Journal*, Article ID 2862381, 2018, http://dx.doi.org/10.1109/JIOT.2018.2862381.

[24] J. Li, X. Chen, X. Huang et al., "Secure distributed deduplication systems with improved reliability," *IEEE Transactions on Computers*, vol. 64, no. 12, pp. 3569–3579, 2015.

[25] X. Li, J. Li, and F. Huang, "A secure cloud storage system supporting privacy-preserving fuzzy deduplication," *Soft Computing*, vol. 20, no. 4, pp. 1437–1448, 2016.

[26] H. Wang, Z. Zheng, L. Wu, and P. Li, "New directly revocable attribute-based encryption scheme and its application in cloud storage environment," *Cluster Computing*, vol. 20, no. 3, pp. 2385–2392, 2017.

[27] Y. Zhang, J. Li, X. Chen, and H. Li, "Anonymous attribute-based proxy re-encryption for access control in cloud computing," *Security and Communication Networks*, vol. 9, no. 14, pp. 2397–2411, 2016.

[28] J. Li, J. Li, X. Dongqing, and Z. Cai, "Secure auditing and deduplicating data in cloud," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 8, pp. 2386–2396, 2016.

[29] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Information Sciences*, vol. 379, pp. 42–61, 2017.

[30] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Computers & Security*, vol. 72, pp. 1–12, 2018.

[31] X. Xu, X. Zhao, F. Ruan et al., "Data placement for privacy-aware applications over big data in hybrid clouds," *Security and Communication Networks*, vol. 2017, pp. 1–15, 2017.

[32] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channels in cloud services: Deduplication in cloud storage," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 40–47, 2010.

[33] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in *Proceedings of the 22nd International Conference on Distributed Systems*, pp. 617–624, IEEE, Austria, July 2002.

[34] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 296–312, Springer, 2013.

[35] J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 6, pp. 1615–1625, 2014.

[36] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, "Message-locked encryption for lock-dependent messages," in *Advances in Cryptology – CRYPTO 2013*, vol. 8042 of *Lecture Notes in Computer Science*, pp. 374–391, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[37] J. Li, C. Qin, P. P. Lee, and J. Li, "Rekeying for encrypted deduplication storage," in *Proceedings of the 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 618–629, Toulouse, France, June 2016.

[38] Y. Zhang, R. H. Deng, X. Liu, and D. Zheng, "Blockchain based efficient and robust fair payment for outsourcing services in cloud computing," *Information Sciences*, vol. 462, pp. 262–277, 2018.

[39] Y. Zhang, R. H. Deng, X. Liu, and D. Zheng, "Outsourcing service fair payment based on blockchain and its application in cloud computing," *IEEE Transactions on Services Computing*, Article ID 2864191, 2018, http://dx.doi.org/10.1109/TSC.2018.2864191.

[40] Y. Zhang, R. H. Deng, J. Shu, K. Yang, and D. Zheng, "TKSE: Trustworthy Keyword Search Over Encrypted Data With Two-Side Verifiability via Blockchain," *IEEE Access*, vol. 6, pp. 31077–31087, 2018.

[41] J. Li, Y. K. Li, X. Chen, P. P. Lee, and W. Lou, "A hybrid cloud approach for secure authorized deduplication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1206–1216, 2015.

[42] S. Wang, "Use of gpu architecture to optimize rabin fingerprint data chunking algorithm by concurrent programming," Tech. Rep., California State University, Long Beach, ProQuest Dissertations Publishing, 2016, https://books.google.com.sg/books?id=GyttnQAACAAJ.

[43] T. Jiang, X. Chen, Q. Wu, J. Ma, W. Susilo, and W. Lou, "Secure and efficient cloud data deduplication with randomized tag," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 3, pp. 532–543, 2017.

[44] H. Su, D. Zheng, and Y. Zhang, "An efficient and secure deduplication scheme based on rabin fingerprinting in cloud storage," in *Proceedings of the IEEE International Conference on Computational Science and Engineering*, pp. 833–836, 2017.

[45] A. Z. Broder, "Some applications of Rabin's fingerprinting method," in *Sequences II*, pp. 143–152, Springer, 1993.

[46] K. R. Jayaram, C. Peng, Z. Zhang, M. Kim, H. Chen, and H. Lei, "An empirical analysis of similarity in virtual machine images," in *Proceedings of the the Middleware 2011 Industry Track Workshop*, pp. 1–6, Lisbon, Portugal, December 2011.

[47] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, pp. 491–500, Chicago, Illinois, USA, October 2011.

[48] Y. Zhao and S. S. Chow, "Towards proofs of ownership beyond bounded leakage," in *Proceedings of the International Conference on Provable Security*, Provable Security, pp. 340–350, Springer, 2016.

[49] L. P. Cox, C. D. Murray, and B. D. Noble, "Pastiche: Making backup cheap and easy," *ACM SIGOPS Operating Systems Review*, vol. 36, pp. 285–298, 2002.