Security Analysis and Improvement of a Dynamic-Hash-Table Based Auditing Scheme for Cloud Storage

Qiang Ma[1], Ti Guan[1], Yujie Geng[1], Jing Wang[2], and Min Luo[2(✉)]

[1] State Grid Shandong Electric Power Company, Jinan, China
[2] School of Cyber Science and Engineering, Wuhan University, Wuhan, China
mluo@whu.edu.cn

Abstract. Cloud storage has emerged as a promising solution to the scalability problem of massive data management for both individuals and organizations, but it still faces some serious limitations in reliability and security. Recently, Tian et al. proposed a novel public auditing scheme for cloud storage (DHT-PA) based on dynamic hash table (DHT), with which their scheme achieves higher efficiency in dynamic auditing than the state-of-the-art schemes. They claimed that their scheme is provably secure against forging data signatures under the CDH assumption. Unfortunately, by presenting a concrete attack, we demonstrate that their scheme is vulnerable to the signature forgery attack, i.e., the cloud service provider (CSP) can forge a valid signature of an arbitrary data block. Thus, a malicious cloud service provider can pass the audit without correct data storage. The cryptanalysis shows that DHT-PA is not secure for public data verification. The purposed of our work is to help cryptographers and engineers design/implement more secure and efficient identity-based public auditing schemes for cloud storage by avoiding such kind of attacks.

Keywords: Cloud storage · Public auditing · Dynamic hash table · Auditing security

1 Introduction

With the explosive growth of data in today's world, the significance of cloud storage service is more and more highlighted [1]. Taking the advantages of elastic storage, ubiquitous access and affordable management, cloud storage providers have attracted an increasing number of individuals and organizations to enjoy this service, such as Microsoft Skydrive, Amazon S3 and Google cloud storage [2–4]. By shifting the data from their local storage system to the remote cloud server, individuals and organizations can greatly relieve themselves from the burden of data management and maintenance. Regardless of these benefits, outsourcing the local data to a remote cloud server still faces some security and privacy challenges. For example, the cloud infrastructure may suffer from some inevitable

D. Wang et al. (Eds.): SPNCE 2020, LNICST 344, pp. 277–285, 2021.
https://doi.org/10.1007/978-3-030-66922-5_19

failures that leads to a data corruption, but the cloud service provider (CSP) may hide the accident to avoid financial loss [5]. Therefore, maintaining the integrity and privacy of cloud data is a key point for prompting the serviceability of cloud storage.

To address the security issues, many public auditing schemes have been proposed to verify the integrity of cloud data, which allow an honest-but-curious public auditor (also called trusted public auditor, TPA) verify the integrity of outsourced data periodically without downloading the entire data file from the remote cloud server. Ateniese et al. [6] first presented the notion of Provable Data Possession (PDP) to check the storage correctness of cloud data without downloading the whole file. On the basis of Ateniese et al.'s conception, Shacham and Waters [7] proposed an improved PDP scheme with Boneh-Lynn-Shacham (BLS) signature, which is widely adopted to construct auditing schemes with additional requirements, such as privacy preserving [8,9] and efficient dynamic auditing [10,11].

Note that a secure public auditing scheme should enable an external auditor to check the storage correctness of cloud data without learning any content of the data, as the introduced TPA is credible but curious. Otherwise, the TPA can reconstruct the while file by collecting all data blocks after several auditing procedures, so that the data copyright of the owner may be violated. Wang et al. [12] is the first to come up with a privacy-preserving auditing scheme by using the random masking technique. Later, there are many other improved privacy-preserving public auditing protocols have been proposed for higher efficiency, such as [13–16].

As for the dynamic data auditing, Erway et al. [17] first came up with a dynamic provable data possession (DPDP) scheme by utilizing a ranked-based skip list, but it cannot support public auditing. Then, Wang et al. [18] proposed a dynamic public auditing scheme with Merkle Hash Tree (MHT). However, both the two dynamic auditing schemes would arouse heavy computation and communication costs during the verification and updating processes. In view of these problems, Zhu et al. [19] came up with an efficient dynamic public auditing scheme (IHT-PA) based on an index-hash table (IHT) by storing the auditing metadata in the side of TPA rather than CSP. However, Tian et al. [20] pointed out that IHT-PA is still inefficient in updating procedure, although it can efficiently support dynamic auditing to some degree.

To get a better tradeoff between the dynamic properties and auditing efficiency, Tian et al. [20] presented a new public auditing scheme (DHT-PA) by exploiting the dynamic hash table (DHT) and Boneh-Lynn-Shacham (BLS) signature to achieve dynamic auditing and batch auditing. Tian et al. proved that DHT-PA is much more efficient than IHT-PA at the time of updating data blocks and files. They also claimed that DHT-PA is secure in terms of resisting the signature forgery attack and proof forgery attack. However, we demonstrate that their scheme is vulnerable to signature forgery attack, i.e., the CSP can forge a valid signature of any data block, with which the CSP can further generate a forged auditing proof to pass the TPA's verification. By providing a new math-

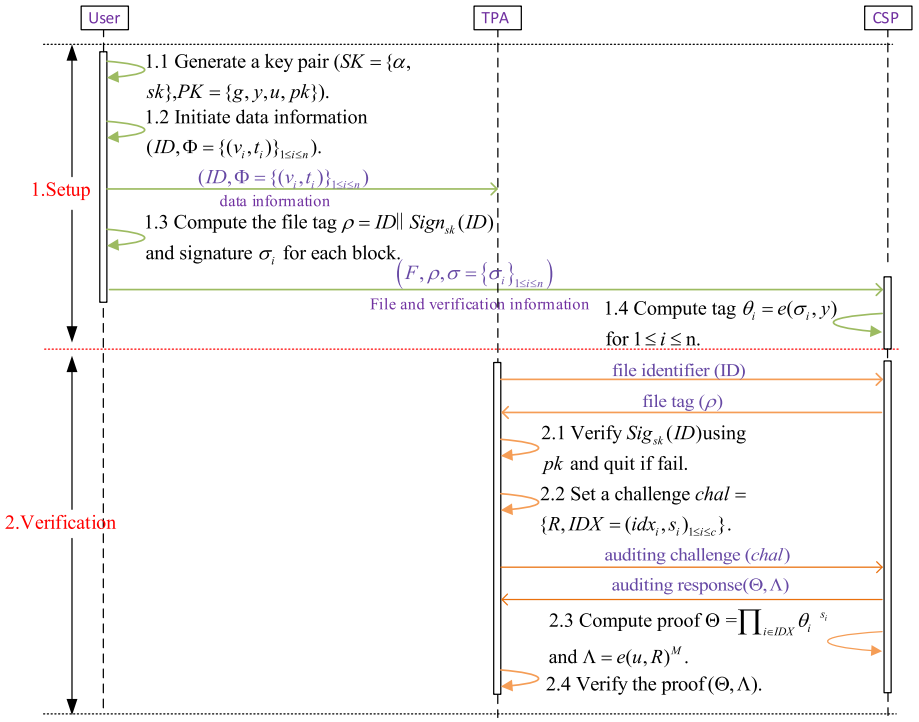

Fig. 1. The auditing process of Tian et al.'s DHT-PA scheme

ematical attack, our work is helpful for cryptographers and engineers to design and implement more secure and efficient identity-based public auditing schemes for cloud storage.

The remainder of this paper is organized as follows: In Sect. 2, we concisely review the scheme proposed by Tian et al. [20]. In Sect. 3, we demonstrate that Tian et al.'s scheme is vulnerable to signature forgery attack, and propose a probable fix to this weakness in Sect. 4. At last, we draw some conclusions for this paper in Sect. 5.

2 Review of DHT-PA

In this section, we give a brief review on Tian et al.'s scheme (DHT-PA) about achieving public dynamic data auditing for cloud storage.

To start with, some definition are presented. $e : G_1 \times G_1 \rightarrow G_2$ is viewed as a bilinear map, where G_1 and G_2 are two additive cyclic groups with the same prime order p. $H : \{0,1\}^* \rightarrow G_1$ is a secure hash function. Let $F = \{m_1, m_2, \cdots, m_n\}$ denote the outsourced file, which is divided into n blocks.

For the sake of simplicity, we will only describe the first auditing part of DHT-PA with setup phase and verification phase as shown in Fig. 1. And the more details for readers can be referred to [20].

2.1 Setup Phase

1) Key initiation: $(SK = \{\alpha, sk\}, PK = \{g, u, y, pk\})$ is a key pair generated by the user, where g and u are two different elements in G_1, (sk, pk) generated for computing file tags.
2) Data information initiation: Let ID be the unique identifier of F. And $\boldsymbol{\Phi} = \{v_i, t_i\}_{1 \le i \le n}$ denotes the latest version information of data blocks, where v_i, t_i are the version and timestamp of block m_i respectively. Then, the user sends $(ID, \boldsymbol{\Phi})$ to the TPA as a delegation of data auditing.
3) Signature Generation: The user first computes the signature for each data block m_i as follows:

$$\sigma_i = H(v_i \| t_i) \cdot u^{m_i + H(v_i \| t_i)}, 1 \le i \le n \tag{1}$$

Then, the user calculates $\rho = ID \| Sig_{sk}(ID)$ as the file tag, where $Sig_{sk}(ID)$ is the signature of ID under the secret key sk. Finally, the user outsources $(F, \rho, \boldsymbol{\sigma})$ to the CSP before deleting them from the local storage, where $\boldsymbol{\sigma} = \{\sigma_i\}_{1 \le i \le n}$.
4) Tag Generation: Upon receiving the signatures σ_i, the CSP computes a tag for each data block as follows:

$$\theta_i = e(\sigma_i, y), 1 \le i \le n \tag{2}$$

After that, the CSP will store $(\rho, \theta_i)_{1 \le i \le n}$ along with the file $F = \{m_1, m_2, \cdots, m_n\}$.

2.2 Verification Phase

1) File identifier check: The TPA first verifies the file signature $Sig_{sk}(ID)$ using the public key pk after receiving the tag ρ. If the verification fails, TPA refuse the user's delegation; otherwise, the TPA launches a challenge for data auditing on behalf of the user.
2) Challenge: The TPA randomly selects a c-element subset $I = \{idx_1, idx_2, \cdots, idx_c\}$ from the set $\{1, 2, \cdots, n\}$ as the index set of the blocks to be checked. Then it sets $chal = \{R, (idx_i, s_i)\}_{i \in I}$ as the auditing challenge and sends it to the CSP, where s_i is a random number from Z_p, $R = y^r$ ($r \in Z_p$ is also a random number).
3) Proof generation: Upon receiving the challenge, the CSP starts to compute the corresponding proof: $\Theta = \prod_{i \in I} \theta_i^{s_i}$, $M = \sum_{i \in I} s_i m_i$ and $\Lambda = e(u, R)^M$. Next, it sends $\{\Theta, \Lambda\}$ back to TPA as the auditing proof.
4) Proof check: To perform the verification, the TPA first computes the value of $H = \prod_{i \in I} H(v_i, t_i)^{s_i}$, then it verifies the proof by checking the following equation:

$$\Lambda \cdot e(H \cdot u^{\sum_{i \in I} s_i H(v_i, t_i)}, R) \stackrel{?}{=} \Theta^r. \tag{3}$$

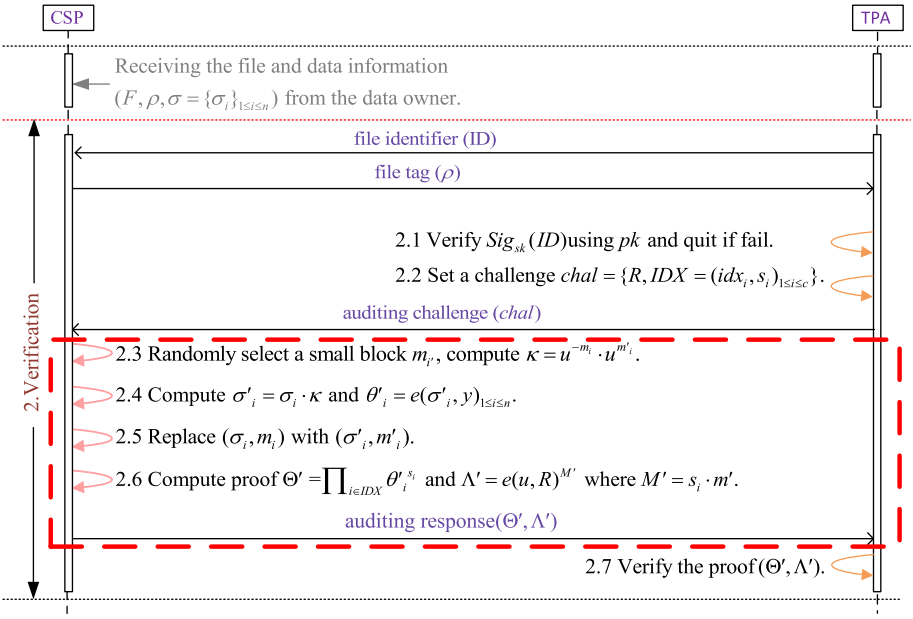

Fig. 2. The forgery attack of Tian et al.'s DHT-PA scheme

If holds, the cloud data is stored correctly; otherwise, the data loses its integrity on the remote node.

3 Cryptanalysis of Li et al.'s Scheme

Tian et al. claimed that their DHT-PA is secure because the CSP cannot keep $m_i^{'}$ instead of m_i to pass the audit. However, in this section, we will analyze the security of DHT-PA on verifying the integrity of the outsourced data, and demonstrate that DHT-PA is insecure against the signature forgery attack, i.e., the CSP can create a legal signature of an arbitrary data block $m_i^{'}$. In other words, the CSP can keep $m_i^{'}$ instead of m_i to pass the audit successfully as shown in Fig. 2. Some details about the attack are presented as below.

Assume that the file F to be outsourced is divided into n blocks, i.e., $F = m_1 \| m_2 \| \cdots \| m_n$. The signature of each data block m_i is denoted as σ_i. Let $\mathscr{A}$ denote the malicious CSP, and it can pass the verification even if it does not correctly store the data by executing the following steps:

1) $\mathscr{A}$ randomly retrieves a signature σ_i of the data block m_i. As the messages transmitted from a user to the CSP is over public channel, thus the step is easily to for an network adversary $\mathscr{A}$ as the way referred in [21,22].
2) $\mathscr{A}$ randomly selects another data block $m_i^{'}$ ($m_i^{'} \neq m_i$), and computes the value of $\kappa = u^{-m_i} u^{m_i^{'}}$ due to the fact that m_i, $m_i^{'}$ and u are public to the CSP.

3) $\mathscr{A}$ computes $\sigma_i^{'} = \sigma_i \cdot \kappa$, and outputs it as the signature of data block $m_i^{'}$. Since $\sigma_i = H(v_i \| t_i) \cdot u^{m_i + H(v_i \| t_i)}$, we would get

$$\begin{aligned}\sigma_i \cdot \kappa &= H(v_i \| t_i) \cdot u^{m_i + H(v_i \| t_i)} \cdot u^{-m_i} u^{m_i^{'}} \\ &= H(v_i \| t_i) \cdot u^{m_i^{'} + H(v_i \| t_i)} \\ &= \sigma_i^{'}\end{aligned}$$

Obviously, $\sigma_i^{'}$ is a valid signature on $m_i^{'}$ according to the above equation.

4) Replace $(m_i, \sigma_i)_{1 \le i \le n}$ with $(m_i^{'}, \sigma_i^{'})_{1 \le i \le n}$.
5) Upon receiving the auditing challenge, $\mathscr{A}$ computes the forged response proof: $\Theta^{'} = \prod_{i \in I} (\theta_i^{'})^{s_i}$, $M^{'} = \sum_{i \in I} s_i m_i^{'}$ and $\Lambda^{'} = e(u, R)^{M^{'}}$, where $\theta_i^{'} = e(\sigma_i^{'}, y)$.
6) $\mathscr{A}$ returns $(\Theta^{'}, \Lambda^{'})$ as auditing proof.

$\mathscr{A}$'s response can surely pass the TPA's verification, we prove it as below:

$$\begin{aligned}&\Lambda^{'} \cdot e(H \cdot u^{\sum_{i \in I} s_i H(v_i, t_i)}, R) \\ &= e(u, R)^{\sum_{i \in I} s_i m_i^{'}} e(\prod_{i \in I} H(v_i \| t_i)^{s_i} \cdot u^{\sum_{i \in I} s_i H(v_i, t_i)}, R) \\ &= e(u^{\sum_{i \in I} s_i m_i^{'}}, R) e(\prod_{i \in I} H(v_i \| t_i)^{s_i} \cdot u^{\sum_{i \in I} s_i H(v_i, t_i)}, R) \\ &= e(\prod_{i \in I} H(v_i \| t_i)^{s_i} \cdot u^{\sum_{i \in I} s_i (m_i^{'} + H(v_i, t_i))}, R) \\ &= e(\prod_{i \in I} (H(v_i \| t_i) \cdot u^{m_i^{'} + H(v_i, t_i)})^{s_i}, g^{r \cdot \alpha}) \\ &= e(\prod_{i \in I} (\sigma_i^{'})^{s_i}, g^{\alpha})^r \\ &= \prod_{i \in I} e(\sigma_i^{'}, y)^{s_i r} \\ &= (\Theta^{'})^r\end{aligned}$$

Hence, the proof $(\Theta^{'}, \Lambda^{'})$ provided by $\mathscr{A}$ can certainly pass the verification of TPA without being detected when it does not store the user's data correctly. In other words, a malicious CSP can hide the corrupted data blocks caused by hardware/software failures; and it also can replace the large data blocks with smaller ones or directly deletes the unfrequently accessed data for space saving. So DHT-PA is not secure as an auditing scheme.

4 Possible Countermeasure

In the above attack, $\mathscr{A}$ just uses the value of $\kappa = u^{-m_i} \cdot u^{m_i^{'}}$ to compute a legal signature $\sigma_i^{'}$ for another data block $m_i^{'}$, and then constructs a legal proof to pass

the TPA's audit. Therefore, to withstand this attack, we should prevent $\mathscr{A}$ from computing $\kappa = u^{-m_i} \cdot u^{m_i^{'}}$ to derive a valid signature. To achieve this goal, we can modify the *Signature Generation* step and *Tag Generation* step as follows.

Signature Generation: Given each data block m_i and public key u, the user generates a corresponding signature σ_i by following equation:

$$\sigma_i = (H(v_i \| t_i) \cdot u^{m_i + H(v_i \| t_i)})^{\alpha} \tag{4}$$

where α is the user's private key generated in *Key Initiation* step. Next, the user sends $(F, \rho, \boldsymbol{\sigma})$ to the CSP, where $\rho = ID \| Sig_{sk}(ID)$, $\boldsymbol{\sigma} = \{\sigma_i | 1 \leq i \leq n\}$.

Compared to the Eq. (1), we exploit the private key to sign the data block, with which $\mathscr{A}$ is not able to obtain the forged signature $\sigma_i^{'}$, because nobody knows the private key α except the data owner.

Tag Generation: Based on the received signature σ_i, the CSP generates a tag θ_i for each data block m_i, namely,

$$\theta_i = e(\sigma_i, g) \tag{5}$$

Compared to the Eq. (2), we replace the public key y with g which are both generated in *Key Initiation* step.

As for the verification phase, it does not need to have any modification. Now, we verify the correctness of Eq. (3) based on the Eq. (4) and Eq. (5) as follows:

$$\begin{aligned}
&\Lambda \cdot e(H \cdot u^{\sum_{i \in I} s_i H(v_i, t_i)}, R) \\
&= e(u, R)^{\sum_{i \in I} s_i m_i} e(\prod_{i \in I} H(v_i \| t_i)^{s_i} \cdot u^{\sum_{i \in I} s_i H(v_i, t_i)}, R) \\
&= e(u^{\sum_{i \in I} s_i m_i}, R) e(\prod_{i \in I} H(v_i \| t_i)^{s_i} \cdot u^{\sum_{i \in I} s_i H(v_i, t_i)}, R) \\
&= e(\prod_{i \in I} H(v_i \| t_i)^{s_i} \cdot u^{\sum_{i \in I} s_i (m_i + H(v_i, t_i))}, R) \\
&= e(\prod_{i \in I} (H(v_i \| t_i) \cdot u^{m_i + H(v_i, t_i)})^{s_i}, g^{r \cdot \alpha}) \\
&= e(\prod_{i \in I} (H(v_i \| t_i) \cdot u^{m_i + H(v_i, t_i)})^{\alpha \cdot s_i}, g^{r}) \\
&= e(\prod_{i \in I} (\sigma_i)^{s_i}, g)^{r} \\
&= \prod_{i \in I} e(\sigma_i, g)^{s_i r} \\
&= \prod_{i \in I} \theta_i^{s_i r} \\
&= (\Theta)^{r}
\end{aligned}$$

Remark. From the above correctness analysis, we can see that the proposed countermeasure can be used to audit the cloud data at the cost of only small

performance loss in computing block signatures. By adding a random exponent to the original tag, it will break the linear relationship between different message tags. And from this point, it may improve the security level of the original DHT-PA scheme by avoiding the attack described in Sect. 3.

5 Conclusion

In this paper, we reviewed the scheme DHT-PA proposed by Tian et al. [20], which is a public auditing scheme using the dynamic hash table to support dynamic auditing. Tian et al. claimed that DHT-PA is secure due to the unforgeability of data signatures and auditing proofs. However, the cryptanalysis of their DHT-PA scheme demonstrates that a malicious CSP can create a valid signature of any data block, so that it can pass the audit of TPA without correct data storage. Therefore, DHT-PA is not secure for practical application. To address the problem, we come up with a possible countermeasure to enhance the security of DHT-PA. And in the near future, we will be devoted ourselves to design a more secure and efficient public auditing scheme.

Acknowledgment. The work was supported by the National Key Research and Development Program of China (No. 2018YFC1604000) and the National Natural Science Foundation of China (Nos. 61972294, 61932016).

References

1. Wang, C., Wang, Q., Ren, K., Cao, N., Lou, W.: Toward secure and dependable storage services in cloud computing. IEEE Trans. Serv. Comput. **5**(2), 220–232 (2012)
2. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst. **25**(6), 599–616 (2009)
3. Liu, J.K., Au, M.H., Huang, X., Lu, R., Li, J.: Fine-grained two-factor access control for web-based cloud computing services. IEEE Trans. Inf. Forensics Secur. **11**(3), 484–497 (2016)
4. Li, Y., Yu, Y., Yang, B., Min, G., Wu, H.: Privacy preserving cloud data auditing with efficient key update. Future Gener. Comput. Syst. **78**, 789–798 (2016)
5. Libing, W., Wang, J., Zeadally, S., He, D.: Privacy-preserving auditing scheme for shared data in public clouds. J. Supercomput. **74**(11), 6156–6183 (2018)
6. Ateniese, G., et al.: Provable data possession at untrusted stores. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 598–609. ACM (2007)
7. Shacham, H., Waters, B.: Compact proofs of retrievability. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 90–107. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89255-7_7
8. Cui, H., Mu, Y., Au, M.H.: Proof of retrievability with public verifiability resilient against related-key attacks. IET Inf. Secur. **9**(1), 43–49 (2015)
9. Yu, Y., Zhang, Y., Ni, J., Au, M.H., Chen, L., Liu, H.: Remote data possession checking with enhanced security for cloud storage. Future Gener. Comput. Syst. **52**, 77–85 (2015)

10. Barsoum, A.F., Hasan, M.A.: Provable multicopy dynamic data possession in cloud computing systems. IEEE Trans. Inf. Forensics Secur. **10**(3), 485–497 (2015)
11. Zhang, Y., Ni, J., Tao, X., Wang, Y., Yong, Yu.: Provable multiple replication data possession with full dynamics for secure cloud storage. Concurr. Comput.: Pract. Exp. **28**(4), 1161–1173 (2016)
12. Wang, C., Chow, S.S.M., Wang, Q., Ren, K., Lou, W.: Privacy-preserving public auditing for secure cloud storage. IEEE Trans. Comput. **62**(2), 362–375 (2013)
13. Zhao, H., Yao, X., Zheng, X.: Privacy-preserving TPA auditing scheme based on skip list for cloud storage. IJ Netw. Secur. **21**(3), 451–461 (2019)
14. Yang, Z., Wang, W., Huang, Y., Li, X.: Privacy-preserving public auditing scheme for data confidentiality and accountability in cloud storage. Chin. J. Electron. **28**(1), 179–187 (2019)
15. Zhang, X., Zhao, J., Xu, C., Li, H., Wang, H., Zhang, Y.: CIPPPA: conditional identity privacy-preserving public auditing for cloud-based WBANs against malicious auditors. IEEE Trans. Cloud Comput. (2019)
16. Tian, H., Nan, F., Chang, C.-C., Huang, Y., Jing, L., Yongqian, D.: Privacy-preserving public auditing for secure data storage in fog-to-cloud computing. J. Netw. Comput. Appl. **127**, 59–69 (2019)
17. Erway, C., Küpçü, A., Papamanthou, C., Tamassia, R.: Dynamic provable data possession. In: Proceedings of the 16th ACM Conference on Computer and Communications Security-CCS 2009, p. 213. ACM Press (2009)
18. Wang, Q., Wang, C., Ren, K., Lou, W., Li, J.: Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans. Parallel Distrib. Syst. **22**(5), 847–859 (2011)
19. Zhu, Y., Ahn, G.-J., Hu, H., Yau, S.S., An, H.G., Hu, C.-J.: Dynamic audit services for outsourced storages in clouds. IEEE Trans. Serv. Comput. **6**(2), 227–238 (2013)
20. Tian, H., et al.: Dynamic-hash-table based public auditing for secure cloud storage. IEEE Trans. Serv. Comput. **10**(5), 701–714 (2015)
21. Libing, W., Wang, J., Kumar, N., He, D.: Secure public data auditing scheme for cloud storage in smart city. Pers. Ubiquitous Comput. **21**(5), 949–962 (2017)
22. Xu, Z., Wu, L., Khan, M.K., Choo, K.-K.R., He, D.: A secure and efficient public auditing scheme using RSA algorithm for cloud storage. J. Supercomput. **73**(12), 5285–5309 (2017)