Identity-based Data Possession Verification Scheme for Multi-Replica Storage in Cloud

Xiaodong Yang*, Xiuxiu Wang, Kaibing Li and Xixi Li

Department of Computer Science and Engineering, Northwest Normal University, room 9c102, Teaching Building No.9, 967 Anning East Road, Anning District, Lanzhou City, China

*Email: y200888@163.com; nuanyue1978297992@163.com

Abstract. In the cloud storage environment, how to efficiently and dynamically complete the integrity verification of multi-user and multi-replica data is a challenging problem. Based on the bilinear mapping signature mechanism and the multi-branch authentication tree, we propose an identity-based integrity verification scheme of multi-replica data for cloud, which solves the problem of large public key storage and management overhead in signature algorithm based on traditional public key cryptosystems. The proof shows that our scheme satisfies the robustness and has the function of protecting privacy. The results of analysis indicate that, compared with the existing similar schemes, our scheme has higher communication efficiency and computing performance.

1. Introduction

Recently, a large amount of data has been stored on Cloud Service Providers (CSP) with strong storage and powerful computing capabilities, as the rapid development of computer technologies such as artificial intelligence, data mining, and deep learning. However, cloud storage increasingly faces severe security challenges [1-3], such as malicious attacks, data leakage, privacy protection, data loss, etc. Therefore, cloud data integrity verification [4-5] is definitely of significance.

Zhu [6] and Cao at al. [7] put forward verification models for provable data ownership. Cha et al. [8] proposed a data integrity certification scheme based on a multi-branch authentication tree, which implements dynamic data operations, but an issue of users' privacy leakage. Liu et al. [9] proposed a Merkle hash tree based secure public auditing scheme for dynamic big data. This scheme has problems of low efficiency and high communication overhead.

The users' identity information is directly used as the public keys in the identity-based cryptosystem [10-12], such as phone number, IP address, name, which eliminates the use of public key certificates. Kang [13] et al. put an efficient identity-based data possession verification scheme, with privacy protection function and resistance to forgery attacks, but no supporting multiple copy operations. Wang et al. [14] referred to the index logic table and proposes a dynamically provable data ownership scheme based on Identity characteristics, which can resist hash storage attacks and delete attacks but not support multiple copies verification.

In view of these problems, based on the schemes [15] and [16], we propose an identity-based data possession mechanism of multi copies of cloud storage, which uses a multi-branch authentication tree. Our solution is not only correct but also satisfies robustness and privacy. By comparing with similar
schemes, it can be concluded that our scheme has lower time and communication overhead, which shows that our scheme has better performance.

2. Preliminary

2.1. Bilinear Pairings

Let $p$ be a large prime number. There are two cyclic groups $G_1$ and $G_2$ of order $p$ and a generator $g$ of $G_1$. If the following three conditions are all true, then $e: G_1 \times G_1 \to G_2$ is said to be a bilinear mapping [17].

- **Bilinear**: For any $a, b \in \mathbb{Z}_p^*$, the equation $e(g^a, g^b) = e(g, g)^{ab}$ holds.
- **Non-degeneration**: $e(g, g)^{ab} \neq 1_{G_2}$, where $1_{G_2}$ is the unit element of $G_2$.
- **Computability**: For any $g_1, g_2 \in G_1$, there is an effective algorithm to calculate $e(g_1, g_2)$.

2.2. Multi-branch Certification Tree

![Figure 1. Multi-branch certification tree.](image)

The multi-branch signature authentication tree has many branch factors [8]. As shown in Figure 1, the depth of the tree is two, the degree of output is $n$; each leaf node in the tree corresponds to the hash value of a data block; the value of a non-leaf node is obtained by the hash of all its child node values, such as, $R_i = H(H(m_{i1}) \parallel H(m_{i2}) \parallel \cdots \parallel H(m_{in}))$ and $R_0 = H(H(R_1) \parallel H(R_2) \parallel \cdots \parallel H(R_n))$.

![Figure 2. System model.](image)
3. Identity-based Data Possession Verification Scheme for Multi-Replica Storage in Cloud

3.1. The System Model of Our Scheme
The system model of our scheme includes four entities: trusted private key generator (PKG), CSP, and third-party auditor (TPA), the system model is shown in Figure 2.

- PKG: The functions of PKG are to generate system parameters and the private keys of users.
- Users: Users can upload data to the cloud and have the right to delete and update data.
- CSP: CSP provides users with data storage services. It needs to manage users’ data stored in the cloud and respond to cloud data verification tasks.
- TPA: TPA audits the integrity of cloud data and feedbacks back the audit results to users.

Firstly, through the setup algorithm, PKG establishes the system and generates the system public parameters and the system master key; through the key generation algorithm, generates the users’ private keys. Then, the user executes the replica generation algorithm, signature algorithm for replicas and the replica storage algorithm. Next, TPA generates challenge information, in order to respond to the challenge, CSP generates proof, and finally, performing verification algorithm, TPA can verify the proof of response.

3.2. Symbols

| Instructions | Description |
|--------------|-------------|
| $\omega$     | The master secret key of the system |
| $sk_i$       | The private key of user $U_i$ |
| $F_i$        | The $i$-th copy of file $F$ |
| $C_i$        | The ciphertext of each replica $F_i$ |
| $c_{ij}$     | The $j$-th block of $C_i$ |
| $m_{ij}$     | The result of $c_{ij}$ being randomized |
| $\tau$       | The signature of root node of the multi-branch tree |
| $\delta_{ij}$| The identity-based signature value of $m_{ij}$ |
| $chal$       | The challenge information |
| $T$          | The response message |

As shown in Table 1, we list some of the Symbols and their descriptions used in this scheme.

3.3. Scheme Description
First of all, we choose two cyclic groups $G_1$ and $G_2$ of prime order $p$, one generator $g$ of $G_1$, and one bilinear mapping $e:G_1\times G_1\rightarrow G_2$. We use 5 anti-collision hash functions $H_0: \mathbb{R} \rightarrow \{0,1\}^*$ (where $\mathbb{R}$ means message space) $H_i: \{0,1\}^* \rightarrow G_i$, $H_2, H_3: \{0,1\}^* \rightarrow \{0,1\}^*$, $H_4: G_1 \times \{0,1\}^* \rightarrow \{0,1\}^*$ and two pseudo-random functions $\psi, \vartheta: Z_p^* \times \{0,1\}^* \rightarrow Z_p^*$.

3.3.1. Setup. PKG executes this algorithm. It selects the $\omega \in Z_p^*$ as a master secret key of the system randomly, that is $sk = \omega$, calculating $pk = g^\omega$ as the system public key. After that, the public parameters are made public, which are $G_1, G_2, p, e, g, H_0, H_1, H_2, H_3, H_4, \psi$ and $\vartheta$. 

As shown in Table 1, we list some of the Symbols and their descriptions used in this scheme.
3.3.2. **Key Generation.** PKG calculates private key of user $U_i$ according to the user’s identity information $id \in \{0,1\}$ and sends the private key $sk_i = H_i(id)^\omega$ to $U_i$.

3.3.3. **Replica Generation.** The user performs the following operations in order to generate replicas:

- For file $F$, the user saves the file as $K$ copies $\{F_i\}_{i \leq K}$.
- The user encrypts each data block $F_i$ to obtain the ciphertext data $\{C_i\}_{i \leq K}$ of each copy. And $C_i$ is divided into $n$ blocks to get $\{c_{ij}\}_{i \leq K, j \leq n}$. By calculating $m_{ij} = c_{ij} + H_z(F \parallel c_{ij} \parallel i \parallel j)$, each ciphertext data block can be randomized. For each $\{m_{ij}\}_{i \leq K, j \leq n}$, we can easily calculate the original data file based on $c_{ij} = m_{ij} - H_z(F \parallel c_{ij} \parallel i \parallel j)$. The user sends $K$ and $\{M_i\}$ to the CSP.

3.3.4. **Signature Generation.** Randomly selecting $\eta \in \mathbb{Z}_p^*$, the user calculates $\theta = g^\eta$. The hash value of the data block is used to construct a multi-branch authentication tree based on multiple copies of data. Meanwhile, the signature of the data block and the signature value of the root node of the multi-branch authentication tree is generated. The constructed multi-branch tree is shown in Figure 3.

![Figure 3. Multi-branch authentication tree.](image)

- Using the hash function $H_0$, the user can generate a multi-branch authentication tree with a depth of $l$ and several child nodes $v$ for each duplicate. Taking Figure 3 as an example, we have $R_0 = H_0(H_0(R_1) \parallel H_0(R_2) \parallel \cdots \parallel H_0(R_K))$ and $R_i = H_0(H_0(m_{i1}) \parallel H_0(m_{i2}) \parallel \cdots \parallel H_0(m_{in}))$;
- Calculate the signature value of the root node of the authentication tree as $\tau = (H_0(R_0))^\omega$;
- Calculate the identity-based signature value of data block $\delta_0 = sk_i H_1(\tau \cdot H_i(id) \parallel R_0 \parallel m_{ij})^\eta$.

The user sends the signature value $\tau$ of $R_0$, the structure information $(l,v)$ of the authentication tree and the signature value $\{\delta_0\}_{i \leq K, j \leq n}$ of the data block to the CSP. At the same time, sends $\tau$ and $\{\delta_0\}$ to the TPA. Except $\tau$, the other local information is deleted.

3.3.5. **Replica Storage.** After receiving the data and signature from the user, the CSP performs the following operations:

- CSP constructs the same multi-branch authentication tree as the user’s, and calculates the value of root node $R_0$ and data node of each replica $\{R_i\}_{i \leq K}$.
- Verifying whether the equation $e(\tau, g) = e(H_0(R_0), pk)$ holds, the CSP can check the validity of the verified data block. If the equation holds, CSP continues to perform the following steps to store data.
- The main storage server saves the root node value $R_0$, the signature value $\{\delta_1\}, j=1, \ldots, n$ of each replica $M_i = \{m_{i1}, \ldots, m_{in}\}$ and the authentication tree node $R_1$. 


• The other replica data $M_i = \{m_{i1}, \ldots, m_{in}\}, i = 2, \ldots, K$ and signature sets $\{\delta_{ij}\}_{ij:0<i<j<2\ldots K}$ are saved by $K - 1$ replica servers.

3.3.6. Challenge. TPA performs the following steps to generate challenge information:
• Randomly selects $c$ numbers from $\{1, \ldots, n\}$ to form set $S = \{s_k\}_{k < c}, s_i \in \{1, \ldots, n\}.
• TPA randomly selects $\beta \in Z_p^*$ and $\alpha_i, \alpha_z \in Z_p^*$.
• Calculate $\chi = e(H_i(id), pk)^\beta$.
• Calculate $\{\varsigma_1 = \psi(\alpha_z, i) | i = 1, \ldots, K\}, \{\gamma_j = \psi(\alpha_z, j) | j = 1, \ldots, n\}$; Let $\varsigma = \{\varsigma_1\}, \gamma = \{\gamma_j\}$.
• Finally, the challenge information $chal = (S, \beta, \alpha_i, \alpha_z, \varsigma, \gamma, \chi)$ is sent to the CSP.

3.3.7. Proof. CSP calculates $\lambda = \sum_{i=1}^{K} \varsigma_i \gamma_i b_g, \delta = (\prod_{i=1}^{K} \delta_i) \gamma_j b_g, \text{and} \mu = \sum_{i=1}^{K} \varsigma_i \gamma_i H_z(F \| c_j \| i \| j)$ when receiving the challenge information, and then sends the response message $T = \{\lambda, \nu, \sigma\}$ to TPA.

3.3.8. Verification. After receiving the response, TPA checks whether the equation (1) is true:

$$\sigma = H_4(e((\tau \cdot H_z(id \| R_0 \| b_g) \sum_{i=1}^{K} \varsigma_i \gamma_i b_g, g^\beta) \cdot \chi^\nu)$$

If the equation is true, it means that the challenge data block is stored correctly and completely, then TPA will feed back the relevant information of the data storage to the user. Otherwise, return $\Gamma$.

4. Correctness Analysis
The analysis of correctness is mainly carried out by verifying whether Equation 1 is true, and the proof is as follows:

$$\sigma = H_4(e(\delta, g^\beta) \cdot \chi^{-\lambda})$$

$$= H_4$$

$$e(\prod_{i=1}^{K} \prod_{j=1}^{j \neq i} \delta_i \gamma_i b_g, g^\beta)$$

$$= H_4$$

$$= H_4$$

$$e(\sum_{j=1}^{K} \sum_{i=1}^{j \neq i} \delta_i \gamma_i b_g, g^\beta)$$

In which,

$$e(\prod_{i=1}^{K} \prod_{j=1}^{j \neq i} \delta_i \gamma_i b_g, g^\beta) = e(\prod_{i=1}^{K} \prod_{j=1}^{j \neq i} sk_i^{s_i} \cdot (\tau \cdot H_3(id \| R_0 \| b_g))^{\gamma_j b_g, g^\beta})$$

$$= e(\prod_{i=1}^{K} \prod_{j=1}^{j \neq i} sk_i^{s_i} \cdot (\tau \cdot H_3(id \| R_0 \| m_j))^{\gamma_j b_g, g^\beta})$$

$$= e(\prod_{i=1}^{K} \prod_{j=1}^{j \neq i} (\tau \cdot H_3(id \| R_0 \| m_j))^{\gamma_j b_g, g^\beta})$$

$$= e(\prod_{i=1}^{K} \prod_{j=1}^{j \neq i} (\tau \cdot H_3(id \| R_0 \| m_j))^{\gamma_j b_g, g^\beta})$$
It can be seen from the above equation transformation that equation 1 holds. Therefore, our scheme is correct and feasible which means CSP cannot be verified by forged evidence.

5. Safety Analysis

5.1. Soundness

**Theorem 1**: The scheme proposed in this paper satisfies soundness.

**Prove.** Peng et al. [15] has proved that the data integrity verification scheme satisfies soundness. Basing on the signature algorithm in [15], thus the signature algorithm of our new scheme in this paper is ensured satisfies soundness.

5.2. Privacy Protection

**Definition 1**: privacy protection

For any probability polynomial-time adversary \( \mathcal{A} \) (semi-trusted third-party verifier), the probability of \( \mathcal{A} \) winning the five-stage safe game is \( 1/2 \pm \varepsilon \). In our scheme, we describe the game between a and \( \mathcal{C} \) as follows:

*Initialization.* \( \mathcal{C} \) runs the initialization algorithm to obtain \( \text{params} \), the master private key \( sk \), and the corresponding master public key \( pk \), and then sends \( \text{params} \) and \( pk \) to \( \mathcal{A} \).

*The first stage query.* \( \mathcal{A} \) submits identity information \( id \), original document \( F \), the number of blocks \( n \), number of copies \( K \). Running the key extraction algorithm, the replica generation algorithm and signature algorithm, \( \mathcal{C} \) can obtain the private key \( sk_i \), copy file \( M_i = \{M_i\} = \{m_i\} \) of \( F \) and signatures \( \{\delta_i\} \). Next, to obtains \( \text{chal} \), \( \mathcal{A} \) does the challenge algorithm and sends \( \text{chal} \) to \( \mathcal{C} \). Correspondingly, \( \mathcal{C} \) generates a response \( T \) and sends it to \( \mathcal{A} \) through challenge algorithm. Repeatedly, selecting different \( \text{id} \), \( F \), \( n \), and \( K \), \( \mathcal{A} \) and \( \mathcal{C} \) perform these operations in polynomial time.

*Challenge.* \( \mathcal{A} \) submits \( \text{id}^*, n^*, K^* \) and the two original data files \( F_0 = \{f_0\}_{i \in \mathcal{G}_0} \) and \( F_1 = \{f_1\}_{i \in \mathcal{G}_1} \) to \( \mathcal{C} \). \( \mathcal{C} \) selects a random bit \( \beta \in \{0, 1\} \). \( \mathcal{C} \) gains \( sk_i^*, M_{\beta} = \{m_{\beta,i}\}_{i \in \mathcal{G}_{\beta}} \) of \( F_{\beta} \), and \( \{\delta_{\beta,i}\} \) by way of executing key generation algorithm, replica generation algorithm and signature algorithm. \( \mathcal{A} \) could be able to gain \( \text{chal}^* \) by challenge algorithm and submits it to \( \mathcal{C} \). Lastly, \( \mathcal{C} \) generates a response \( T_{\beta} \) and sends it to \( \mathcal{A} \).

*The second stage of inquiry.* Consistent with the first stage query.

*Guess.* \( \mathcal{A} \) outputs a guessed bit \( \beta \). when \( \beta=\beta \), \( \mathcal{A} \) wins the game.

**Theorem 2** The scheme proposed in this paper satisfies the characteristics of privacy protection.

**Proof:** There are two facts that are obvious, according to the game defined in Definition 1. In two query stages, there are same \( T \) for the same \( id \), \( n \), \( K \), and different \( F \). In the challenge process, respond \( T_0=T_1 \) holds. When \( id^*, n^*, K^*, F_0, F_1 \) and \( \text{chal}^* \) remains unchanged.

In other words, the response \( T \) clearly contain no information of \( F \). Because \( \mathcal{A} \) cannot know any information about the original data, the probability of \( \mathcal{A} \) winning the game is \( 1/2 \), and \( F_0, F_1 \) are completely indistinguishable. From this, theorem 2 can be proved.

6. Performance Analysis

We perform performance analysis by comparing the multiple copy integrity verification schemes in [12] and [13] with ours. We assume that \( n \) is message signatures with \( n \) identities, \( c \) represents the number of copies that need to be verified, and groups in all three schemes are \( G_1 \) and \( G_2 \) with same prime order \( p \).
6.1. Communication Cost
In the communication cost analysis, we only consider the cost occupied by the elements in the group. $G_1$, $G_2$ represent the size of the elements in a group $G_1$, $G_2$. The comparison result of the communication authentication overhead of each scheme is shown in Table 1.

**Table 2.** Communication overhead comparison.

|              | Challenge | Proof | Verification |
|--------------|-----------|-------|--------------|
| MUR-DPA [9]  | $cG_i$    | $cG_i$| $2ncG_i$     |
| Peng et al. [15] | $nG_i+nG_2$ | $nG_i+nG_2$ | $nG_i$ |
| ours         | $nG_i+nG_2$ | $nG_2$ | $nG_i$ |

As shown in Table 2, the communication cost of our scheme in completing a signature verification process is better than that of paper [12] and paper [13].

6.2. Time Cost
In the efficiency analysis process, we only consider the cost of the bilinear pairing operation and use $p_a$ to represent a bilinear pairing operation. We compare the time cost of the three schemes of a message authentication process, as shown in Table 2.

**Table 3.** Time cost comparison.

|              | Challenge | Proof | Verification |
|--------------|-----------|-------|--------------|
| MUR-DPA [9]  | $0p_a$    | $0p_a$| $2np_a$     |
| Peng et al. [15] | $np_a$ | $np_a$ | $np_a$ |
| ours         | $np_a$    | $np_a$ | $np_a$ |

From Table 3, it is clear that the computational cost of our scheme in completing a signature verification process is less than that of paper [12] and paper [13].

7. Conclusion
In this paper, we propose a multi-copy data integrity audit model based on a multi-branch authentication tree. The scheme uses identity-based signature verification mechanism. We proved the correctness and robustness of the scheme in the paper. Simultaneously, our program can protect user privacy. Through communication overhead analysis and efficiency analysis, it is evident that our solution has higher performance than similar solutions.

References
[1] Tie Q, Ning C, Keqiu L, Daji Q and Zhangjie Fu 2017 Heterogeneous ad hoc networks: architectures, advances and challenges. *J. Ad Hoc Networks* **55** 143-152.
[2] Yuqing Z, Xiaofei W, Xuefeng L and Ling L 2016 Overview of cloud computing environment security *J. Software Journal* **27**(06) 1328-48
[3] Mishra B, Mnaya P, and Behera S 2011 Security in vehicular adhoc networks: a survey *C. Proceedings of the 2011 International Conference on Communication, Computing & Security, Odisha, India: ACM.*
[4] Thokchom S and Saikia D K 2019 Privacy preserving and public auditable integrity checking on dynamic cloud data *J. International Journal of Network Security, 21**(2) 221-229.
[5] Ambika K and Moses M B 2020 An efficient SG-DACM framework for data integrity with user revocation in role based multiuser cloud environment *J. Computer Communications* **155** 84-92.
[6] Zhu Y, Hu H, Ahn G J and Yu M. 2012 Cooperative provable data possession for integrity verification in multi cloud storage J. IEEE Transactions on Parallel & Distributed Systems 23(12) 2231-44.

[7] Laicheng C, Wenwen H, Xian G and Tao F 2016 A scheme for verification on data integrity in mobile multi cloud computing environment J. Mathematical Problems in Engineering 2016(pt.8) pp 1-6.

[8] Zha Y X, Luo S S, Bian J C and Li W. 2015 Multiuser and multiple-replica provable data possession scheme based on multi-ranch authentication tree J. Journal on Communications 36(11) 80-91

[9] Liu C, Ranjan R, Yang C, Zhang X, Wang Land Chen J 2016 MuR-DPA: top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud J. IEEE Transactions on Computers 64(9) 2609-22.

[10] Shamir A 1984 Identity-Based cryptosystems and signature schemes C. Workshop on the Theory and Application of Cryptographic Techniques Springer, Berlin, Heidelberg.

[11] Shim K A 2010 An ID-based aggregate signature scheme with constant pairing computations J. Journal of Systems & Software 83(10) 1873-80

[12] Wang Z 2017 An identity-based data aggregation protocol for the smart grid J. IEEE Transactions on Industrial Informatics 13(5) 2428-35.

[13] Kang B, Si L, Jiang H, Chunqing L and Mingming X 2018 ID-based public auditing protocol for cloud data integrity checking with privacy-preserving and effective aggregation verification J. Security and Communication Networks 3 1-9.

[14] Wang F, Xu L, Wang H and Chen Z 2018 Identity-based non-repudiable dynamic provable data possession in cloud storage J. Computers & Electrical Engineering 2018 521-533.

[15] Peng S, Zhou F, Wang Q, Xu Z and Xu J 2017 Identity-based public multi-replica provable data possession J. IEEE Access 99 1.

[16] Yang X, Pei X, Wang M, Li T, and Wang C 2020 Multi-replica and multi-cloud data public audit scheme based on blockchain J. IEEE Access 99 1.

[17] Horng S J, Tzeng S F, Huang P H, Wang X, Li T and Khan M K. 2015 An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks J. Information Sciences 317 48-66