



Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:

Yu, H;Cai, Y;Sinnott, RO;Yang, Z

Title:

ID-based dynamic replicated data auditing for the cloud

Date:

2019-06-10

Citation:

Yu, H., Cai, Y., Sinnott, R. O. & Yang, Z. (2019). ID-based dynamic replicated data auditing for the cloud. CONCURRENCY AND COMPUTATION-PRACTICE & EXPERIENCE, 31 (11), <https://doi.org/10.1002/cpe.5051>.

Persistent Link:

<https://hdl.handle.net/11343/284968>

ID-based dynamic replicated data auditing for the cloud

Haiyang Yu^{1,2}, Yongquan Cai¹, Richard O. Sinnott^{2,*} and Zhen Yang¹

¹ Faculty of Information, Beijing University of Technology, Beijing 100124, P. R. China

² School of Computing and Information Systems, University of Melbourne, Melbourne 3010, VIC, Australia

SUMMARY

As an essential component of cloud computing, cloud storage provides flexible data storage services for individuals and organizations. By storing multiple replicas of data in servers, cloud storage providers (CSPs) can improve availability and stability of the cloud storage service. To ensure that all data replicas of a cloud user are intact and completely stored in the CSP, many multi-replica cloud auditing schemes have been proposed. However, such schemes are predominantly based on the public key infrastructure (PKI), which incurs complex certificate management. In addition, existing schemes do not consider support for sector-level dynamic auditing. In this paper, we propose a fine-grained dynamic multi-replica data auditing scheme that has the following features: 1) it uses ID-based cryptography to eliminate the cost of certificate management; 2) it supports efficient sector-level dynamic operations on cloud user data, and 3) it optimizes the challenge algorithm to reduce the computational cost of the third party auditor (TPA). We show that the proposed scheme is provably secure based on a random oracle model. The performance analysis and experiments show the efficiency of the proposed scheme.

KEY WORDS: data integrity, replicated data auditing, dynamic operation, ID-based cryptography.

1. INTRODUCTION

1.1. Background

Cloud storage is now widely accepted as a key capability offering massive storage to cloud users in an on-demand service model [1]. By leveraging cloud storage, individuals and enterprises can access their data from anywhere and avoid capital expenditure required for local storage. However, cloud storage is a double-edged sword, which brings benefits as well as security threats in terms of data integrity and availability. Firstly, cloud user data may be deleted or corrupted by attackers. For

This is the author manuscript accepted for publication and has undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the Version of Record. Please cite this article as doi: [10.1002/cpe.5051](https://doi.org/10.1002/cpe.5051)

instance, increasing DDOS attacks on cloud service provider (CSP) can bring cloud services down, which could cause data loss and corruption. Second, management fault or system outages may lead to serious data damage. According to the Cloud Security Alliance (CSA) [2], the Amazon Elastic Compute Cloud (EC2) crash caused permanent corruption of a range of data sets from multiple organizations. Third, since a cloud user cannot take direct control of its cloud data, e.g. as it might if it was on their own local server, the CSP may hide data loss incidents to maintain a good reputation. In some cases, it may even deliberately delete some rarely used data to save storage space.

To ensure the integrity and availability of cloud data, most CSPs support data backup mechanisms whereby multiple data replicas are stored in the cloud. However, since a given CSP cannot be fully trusted, it may store fewer data copies than a given cloud user's requirement. Therefore, an auditing scheme that verifies the integrity of all data replicas would be highly beneficial to cloud users.

Recently, a number of multi-replica cloud auditing schemes have been proposed to check the integrity of cloud users' replicas with high probability. However, the majority of them are based on public key infrastructure (PKI), which introduces several drawbacks. First, PKI-based schemes lead to complicated certificate management such as certificate generation, revocation and distribution. Second, PKI-based schemes require the third party auditor (TPA) to check the validity of public key certificates of cloud users before performing cloud data auditing. This increases the computational overheads of the TPA, especially with the increasing number of cloud users.

In order to solve these problems, ID-based cloud auditing schemes have been proposed by introducing ID-based cryptography (IBC) [3]. Compared with PKI, IBC does not require public key certificates hence it eliminates the abovementioned certificate management overheads. Furthermore, due to the certificate-free approach, IBC does not require certificate verification of a cloud user's public key during auditing. This greatly reduces the computation burden of the TPA.

Cloud users not only store and access their data files, but they also need to update them. Thus, a major concern in cloud auditing is to support dynamic data operations such as modification, insertion and deletion of data over cloud storage infrastructure. In existing dynamic auditing schemes, these operations are at block level. Sectors cannot be updated individually because any update in sectors requires a cloud user to recalculate the homomorphic verifiable tag (HVT) of the corresponding data block. This incurs prohibitive computation costs on the user side, especially in multi-replica storage systems. Therefore, to relieve the cloud user of the computation overhead, it is necessary to introduce a fine-grained (sector-level) dynamic update mechanism into the multi-replica cloud auditing scheme.

1.2. Related work

Ateniese et al. [4] first proposed the concept of provable data possession (PDP) to check the integrity of remote data. PDP used RSA-based signatures to construct HVTs.

It utilized a sampling strategy to check data integrity with high probability. Juels and Kalisk [5] presented the notion of proof of retrievability (POR), which checked the integrity of remote data and retrieves corrupted data files. Ateniese et al. [6] also focused on a scalable and efficient PDP supporting all dynamic data operations with the exception of data insertion.

Erway et al. [7] first proposed a dynamic provable data possession (DPDP) scheme supporting all dynamic data operations. Their scheme adopted a rank-based authenticated skip list to manage dynamic information of data blocks. However, due to the removal of index information in HVTs, their scheme required extra auxiliary authentication information (AAI) during verification. In addition, their scheme did not support public auditability. Wang et al. [8] proposed another fully dynamic auditing scheme by introducing Merkle Hash Trees (MHTs). However, their scheme could not verify the indices of data blocks, which meant that the cloud storage server could pass verification with valid data blocks in any position. Zhu et al. [9] proposed a dynamic auditing scheme associated with an index hash table. However, these schemes as well as a number of cloud auditing schemes [10, 11, 12, 13, 14] only considered dynamic auditing in single replica storage systems, and hence they lacked the ability to deal with data backup and recovery when data corruption happened.

For integrity verification of multiple replicas, Curtmola et al. [15] first proposed multi-replica provable data possession (MR-PDP) to verify the integrity of multiple copies of cloud user's data stored on multiple servers. However, their scheme could not support dynamic auditing. Mukundan et al. [16] presented a replicated data integrity verification scheme supporting verification of dynamic multiple replicas. However, as with Wang et al.'s scheme [8], their scheme could not verify the position of data blocks due to the removal of index information in HVTs. Barsoum and Hasan [17] proposed tree-based dynamic multi-copy provable data possession (TB-DMCPDP), as an extension of MHT-based dynamic PDP. Zhang et al. [18] proposed a provable multiple replication data possession scheme with full dynamics by modifying the MHT to support position verification. Liu et al. [19] proposed a dynamic multi-replica auditing scheme by organizing all replicas of each data block into a replica sub-tree, which reduced the communication cost of both data update and integrity verification. However, the scheme required dynamic adjustment of the Merkle hash tree after each dynamic operation. Furthermore, despite of communication cost optimization, the size of AAI was still dependent on the size of the challenged data, which was impractical when considering a large amount of distributed data stored in the cloud. Dynamic multi-replica schemes based on the skip list [20] or the MHT incurred heavy storage and communication overheads. This was because these dynamic data structures have to store the hash codes of all replicas and transfer the AAI during verification. Barsoum and Hasan [17, 21] proposed a map-based provable multicopy dynamic data possession (MB-PMDDP) scheme by introducing a novel dynamic data structure named map-version table (MVT). This scheme only stored the dynamic information of one replica and could eliminate the AAI, which reduced the storage, communication and computation overheads. However, their scheme could not support fine-grained dynamic operations. Their

scheme also gave rise to security issues that might cause HVT forgery.

Existing data integrity auditing schemes [22, 23, 24] mostly relied on PKI, which involved certificate management and efficiency problems. To address these issues, Zhao et al. [25] first introduced IBC into cloud auditing and proposed an ID-based PDP scheme. Wang et al. [26] proposed an ID-based distributed PDP scheme. They further extended the scheme to support multi-cloud storage scenarios [27]. However, this scheme was not always sound since cloud storage providers could deceive third party auditors (TPAs) by storing hash codes and HVTs of data blocks whilst deleting the corresponding data blocks. Furthermore, the combiner in the scheme was not practical in reality. Yu et al. [28, 29] proposed two schemes that combined the traditional PDP scheme and the ID-based signature (IBS). However, the schemes were inefficient when dealing with multiple users and CSPs. Wang et al. [30] proposed an ID-based proxy-oriented cloud auditing scheme that allowed a proxy to help cloud users generate HVTs and upload data files. However, the approach did not support either dynamic auditing or multi-replica auditing.

Based on the current literature identified above, we identify that dynamic multi-replica auditing schemes based on IBC are limited. Moreover, there is almost no research supporting sector-level dynamic operations on data. To tackle this, we propose a novel ID-based fine-grained dynamic multi-replica auditing scheme with overall efficiency as a core requirement.

1.3. Main contributions

Our contributions can be summarized as follows:

- we propose an ID-based dynamic replicated data auditing (ID-DRDA) scheme; we introduce ID-based cryptography (IBC) into the scheme to eliminate complex key and certificate management whilst optimizing the challenge algorithm to improve the efficiency of the TPA;
- the proposed scheme supports fine-grained dynamic data operations at both the block level and the sector level;
- we present a performance comparison between our scheme and existing state of the art approaches through a range of experiments that show the efficiency of our scheme;
- we prove the security of the proposed scheme based on a random oracle model; we show that the CSP can pass the auditing with a valid proof only if it stores intact data blocks of all replicas.

1.4. Paper organization

The remainder of the paper is structured as follows. The problem statement is presented in Section 2. We give the detailed description of the proposed scheme in Section 3. Section 4 provides the security analysis of the proposed scheme. The performance analysis and the experiment evaluation are presented in Section 5. Finally, we draw conclusions on the work as a whole in Section 6.

2. PROBLEM STATEMENT

2.1. System model

The cloud storage auditing model in this paper consists of four entities as shown in Figure 1. This includes: 1) a cloud user who stores a large amount of data in the cloud and has limited resources; 2) a cloud service provider (CSP) with significant computation and storage resources offered on a pay for use to cloud users; 3) a third party auditor (TPA) with the necessary expertise to perform auditing services on the CSP on behalf of cloud users; 4) a public key generator (PKG) that generates system parameters and issues private keys for cloud users according to their identities.

In our system, the PKG first sets up the system parameters and generates a private key for each cloud user. A cloud user uploads its data and HVTs to the CSP and updates its data frequently. The TPA challenges the CSP and verifies the proof generated by the CSP.

Definition 1 (Algorithms in the proposed scheme).

$Setup() \rightarrow (params, mpk, msk)$. The *Setup* algorithm is run by the PKG to set up the whole system. It outputs parameters $params$, a master public key mpk , and a master private key msk .

$KeyExtract(mpk, msk, params, ID) \rightarrow sk$. The *KeyExtract* algorithm is run by the PKG to extract the cloud user's private key from its identity. It takes the master public key mpk , the master private key msk , system parameters $params$ and the cloud user's identity ID as input. It outputs the cloud user's private key sk .

$ReplicaGen(F, N_i) \rightarrow \tilde{F}$. The *ReplicaGen* algorithm is run by the cloud user to generate replicas of its data. It takes as input the file F and a replica number N_i and outputs all data replicas $\tilde{F}$.

$TagGen(sk, \tilde{F}) \rightarrow \Phi$. The *TagGen* algorithm is run by the cloud user to generate HVTs. It takes as input the cloud user's private key sk and data replicas $\tilde{F}$, and outputs the set of HVTs Φ .

$Challenge(F_{inf}) \rightarrow chal$. The *Challenge* algorithm is run by the TPA to randomly check the cloud user's data blocks that are stored in the CSP. The input is the information of files F_{inf} . It outputs a challenge $chal$ that contains the keys used for generating the indices and coefficients of challenged data blocks.

$Response(chal, \tilde{F}, \Phi) \rightarrow P$. The *Response* algorithm is run by the CSP to respond with the data proof to the TPA. It takes as input the challenge $chal$, the replica $\tilde{F}$

and the set of HVTs Φ and outputs a data proof P .

$Verify(P, chal, mpk, pk, F_{inf}) \rightarrow \{1, 0\}$. The *Verify* algorithm is run by the TPA to verify the integrity of challenged data blocks. It takes as input the data proof P , the challenge $chal$, the master public key mpk , the cloud user's public key pk , and the file information F_{inf} . The output is 1 if the verification equation holds true or 0 otherwise.

2.2. Design goals

To realize the ID-DRDA scheme for the cloud using the aforementioned model, our design should achieve the following goals.

- *Certificate elimination*: the ID-DRDA scheme should eliminate complex key and certificate management for all entities involved in the ID-DRDA scheme.
- *Fine-grained dynamic auditing*: the proposed scheme should allow a cloud user to efficiently update its data at the sector level.
- *Lightweight*: the scheme should improve the efficiency of both the cloud user and the TPA.
- *Storage correctness*: there should not exist a dishonest CSP that can pass the TPA's auditing unless it stores all of the cloud user's files intact.
- *Privacy-preserving*: the TPA in the proposed scheme cannot retrieve any content from the auditing proof generated by the CSP.

2.3. Security requirements

We assume that the PKG is a trusted entity and always honestly provides the private keys for cloud users. The CSP is a semi-trusted entity that may ignore data corruption issues caused by attacks or management faults to maintain its reputation. It may deliberately delete cloud user's rarely used data files to save storage space and thereby achieve further profit. The TPA is an honest-but-curious entity that honestly performs integrity checking services. It may try to obtain the content of cloud user's files.

Definition 2 (Unforgeability). Following the security definition from prior auditing schemes [4, 7, 31], the security of the ID-DRDA scheme can be formulated as a game between an adversary A and a challenger C . The detailed cloud auditing game is given as follows.

- *Setup*: C runs the *Setup* algorithm to set up the whole system and outputs parameters $params$, the master public key mpk , and the master private key msk . It forwards the parameters $params$ and master public key mpk to A . It keeps the master private key msk secret.
- *Queries*: A adaptively sends a number of queries to C in probabilistic polynomial time.
 - 1) *Extract queries*: A sends an identity ID to C to query the corresponding private key. C then executes the *KeyExtract* algorithm to generate the

- cloud user's private key sk and forwards it to A .
- 2) *ReplicaGen* queries: A selects a file F to query C for the replicas. C runs the *ReplicaGen* algorithm to generate a set of replicas $\tilde{F}$ and subsequently sends it to A .
 - 3) *TagGen* queries: A adaptively queries C for the HVTs of arbitrary data blocks. C runs the *TagGen* algorithm to create the corresponding HVTs and forwards them to A .
 - 4) *Hash* queries: A adaptively queries the hash functions. The hash values generated from the hash function are forwarded to A by C .
 - 5) *Dynamic operations* queries: A adaptively chooses data sectors to query C for dynamic operations. C returns the associated dynamic update requests to A .
- *Output*: In the end, adversary A outputs an HVT σ' of a new data block b' with new identity ID' . The new data block b' and identity ID' should not be queried in the Query phase. A wins the game if σ' is a valid HVT for the data block b' .

The proposed auditing scheme is secure if the probability that any probabilistic polynomial time adversary A wins the game is negligible.

3. THE PROPOSED SCHEME

3.1. Overview

The main challenge in the design of the ID-DRDA scheme is the scalability problem in dynamic auditing. Many existing auditing schemes adopt data fragment techniques [32] to reduce the storage cost associated with HVTs. However, the side effect of adopting this technique is that auditing schemes can only support block-level dynamic operations. Any change in a sector will result in the HVT re-generation of the entire block. To address this, our method utilizes the homomorphic property of the HVT to update the HVT based on the differences between new and old sectors. This reduces the computation cost of re-calculating the HVT. In this way, we efficiently extend our scheme to support sector-level dynamic auditing.

To improve the efficiency of multi-replica auditing systems, we integrate ID-based cryptography into the ID-DRDA scheme. In our scheme, public keys are generated based on identities of the involved entities whilst private keys are generated by the PKG. Thus, compared with PKI-based multi-replica auditing schemes, there exists no computation cost associated with validating certificates or communication cost incurred in obtaining certificates. In addition, we redesign the HVT based on ID-based signatures to alleviate the computational burden for both cloud users and the TPA.

Although the TPA checks all copies of challenged data blocks to ensure the integrity of all replicas, this incurs redundant computation cost within the *Response* and *Verify* algorithms. Instead, we propose a novel sampling strategy in which replicas

are considered as trivial data blocks. n copies of a cloud user's m data blocks will be considered as $n \times m$ individual data blocks. As a result, the TPA in ID-DRDA only needs to challenge certain number z of data blocks from $n \times m$ data blocks rather than $z \times n$ blocks for existing schemes.

To deal with the HVT forgery problem [32], we adapt each HVT with a random value chosen by the cloud user. The random value is also kept by the TPA and used for validating data proofs. As a result, the CSP cannot forge a valid HVT based on two replicas with the same hash values in their HVT.

3.2. Preliminaries

Bilinear Pairings. Let G_1 and G_T be two multiplicative cyclic groups of large prime order q . Let g_1 and g_2 be the generators of G_1 and G_T , respectively. A bilinear map is a map $e: G_1 \times G_1 \rightarrow G_T$ with the following three properties:

- 1) Bilinear: for $u, v \in G_1$ and $a, b \in \mathbb{Z}_q^*$, $e(u^a, v^b) = e(u, v)^{ab}$;
- 2) Non-degenerate: $\exists g_1, g_2 \in G_1$ such that $e(g_1, g_2) \neq 1$;
- 3) Computable: there exists an efficient algorithm to compute the map e .

Modified Map-Version Table. A Map-version table (MVT) is a dynamic data structure stored in the TPA [21]. It stores the dynamic information of data blocks needed for the *Verify* algorithm. Compared to MVT, our modified map-version table (MMVT) stores one more column containing random values to address the aforementioned security problems with existing multi-replica cloud auditing schemes. An MMVT consists of four columns: index number i , original block number BN_i , version number VN_i , and random value c_i . Figure 2 shows examples of changes in the MMVT due to different dynamic operations on data blocks.

In existing auditing schemes, authenticated data structures such as the Merkle Hash Tree (MHT) and skip list are also used to support dynamic data operations. Compared with these approaches, the MMVT has advantages in reducing the dynamic storage cost and the communication cost during verification. Let m be the number of data blocks in a replica and n be the number of replicas. The storage cost of the MMVT is $O(m)$ while that of authenticated data structures is $O(nm)$. This is because the MMVT only keeps one tuple (i, BN_i, VN_i, c_i) for all replicas of a data block b_i while the MHT and skip list need to store all hash codes of the replicas. In addition, the communication cost in the verification process can be reduced since there is no AAI needed to be transferred when utilizing the MMVT as the dynamic data structure.

3.3. Basic ID-DRDA scheme

The algorithms of the ID-DRDA scheme are illustrated as follows.

- *Setup*: Let G_1 and G_T be two multiplicative cyclic groups of prime order q . Let $e: G_1 \times G_1 \rightarrow G_T$ be a bilinear map and let g be a generator of group G_1 . Let $H: (0,1)^* \rightarrow G_1$ and $h: (0,1)^* \rightarrow Z_q^*$ be two cryptographic hash functions. Let f be a pseudo-random function (PRF) and π be a pseudo-random permutation (PRP). PKG randomly generates a master private key $x \in Z_q^*$ and a master public key $Y = xg \in G_1$.

PKG keeps master private key x private and publishes system parameters $\{e, H, h, Y, f, \pi\}$.

- *Key Extract*: A cloud user sends its identity ID to the PKG. The PKG computes the cloud user's public key $Q_{ID} = H(ID)$ and its private key $d_{ID} = xQ_{ID} = xH(ID)$. Following this, the PKG forwards the private key d_{ID} to the cloud user via a secure channel.
- *Replica Generation*: A cloud user splits a file F into m data blocks $F = \{b_1, b_2, \dots, b_m\}$ and each data block b_{jk} is further divided into s sectors $b_j = \{b_{j1}, b_{j2}, \dots, b_{js}\}$. By using a symmetric encryption algorithm E_K , the cloud user generates n replicas $\tilde{F} = \{\tilde{F}_i\}_{i \in [1,n]} = \{\tilde{b}_{ijk}\}_{i \in [1,n], j \in [1,m], k \in [1,s]}$, where $\tilde{b}_{ijk} = E_K(i || b_{jk})$.
- *Tags Generation*: The cloud user randomly chooses s numbers $(r_1, r_2, \dots, r_s) \in Z_q^*$. It computes s elements $\{u_k\}_{k \in [1,s]}$, where $u_k = r_k \cdot d_{ID} \in G_1$. It then calculates $R_k = r_k \cdot H(ID)$ for $k \in [1, s]$. The cloud user computes a HVT for each data block $\tilde{b}_{ij}$ as $\sigma_{ij} = \sum_{k=1}^s u_k \cdot \tilde{b}_{ijk} + h(ID_F || BN_j || VN_j || c_j || i) \cdot d_{ID}$, where ID_F is the identifier of file F . BN_j denotes the original index of the data block b_j . VN_j denotes the version number of the data block b_j . c_j is a random value of each data block b_j .
- *Challenge*: The TPA considers cloud user data replicas as $n \times m$ data blocks. It picks the number of challenged data blocks z and three random keys $k_1, k_2, k_3 \in Z_q^*$. It sends the challenge $chal = (z, k_1, k_2, k_3)$ to the CSP.
- *Response*: Upon receiving the challenge $chal$, for $1 \leq l \leq z$, the CSP computes $i = \pi_{k_2}(l)$, $j = \pi_{k_1}(l)$, and $v_{ij} = f_{k_3}(l)_{l \in [1,z]}$ to generate a challenge set $Q = \{(i, j, v_{ij})\}$. It then computes

$$\sigma = \sum_{(i,j,v_{ij}) \in Q} v_{ij} \cdot \sigma_{ij}$$

$$\mu_k = \sum_{(i,j,v_{ij}) \in Q} v_{ij} \cdot \tilde{b}_{ijk}$$

to generate a data proof $P = \{\sigma, \{\mu_k\}_{k \in [1,s]}\}$. Finally, the CSP responds to the

TPA with data proof P .

- *Verify*: Upon receiving the data proof P from the CSP, the TPA generates a challenge set Q using the same aforementioned process. It then verifies the integrity of challenged data blocks by calculating the following equation:

$$e(\sigma, g) = e\left(\sum_{k=1}^s R_k \mu_k + \sum_{(i,j,v_{ij}) \in Q} h_{ij} v_{ij} Q_{ID}, Y\right), \quad (1)$$

where $h_{ij} = h(ID_F || BN_j || VN_j || c_j || i)$. If the verification equation holds, the TPA reports 1 to the cloud user, otherwise 0.

The new sampling strategy in the *Challenge* algorithm reduces the computation cost of the TPA whilst achieving the same verification probability compared with existing auditing schemes. Suppose that a cloud user has n copies of m data blocks in the cloud and each block has s sectors. Let ρ be the corruption probability of each sector. In our sampling strategy, the TPA challenges z data blocks from $n \times m$ data blocks. Therefore, the verification probability can be calculated as

$$P = 1 - (1 - \rho)^{zs}.$$

For $\rho = 0.1\%$ and $s = 50$, to achieve 95% verification probability, the TPA needs to check 60 data blocks. It should choose 93 data blocks for achieving 99% verification probability.

3.4. Fine-grained dynamic auditing

The fine-grained dynamic auditing supports sector-level operations including modification and deletion operations. Sector-level insertion operations can be omitted in auditing schemes with fixed size data blocks. This is because inserting sectors into a block can be considered as the combination of modifying sectors and inserting a new block. Thus, it is not necessary to support sector-level insertion operations.

Modification: Assume that a cloud user modifies the sectors $\{\tilde{b}_{ijk}\}_{i \in [1,n]}$ in all copies $\{\tilde{F}_i\}_{i \in [1,n]}$. It should follow the steps below:

- The cloud user first calculates the differences of sectors $\Delta \tilde{b}_{ijk} = \tilde{b}'_{ijk} - \tilde{b}_{ijk}$, for $i \in [1,n]$.
- It then computes the differences of HVTs $\Delta \sigma_{ij} = u_k \cdot \Delta \tilde{b}_{ijk} + h(ID_F || BN_j || VN_j || c_j || i)^{-1} \cdot h(ID_F || BN_j || VN'_j || c_j || i) \cdot d_{ID}$ for $i \in [1,n]$.
- Finally, it sends a modification request $\langle ID_F, Mod, j, \{\tilde{b}_{ijk}\}_{i \in [1,n]}, \{\Delta \sigma_{ij}\}_{i \in [1,n]} \rangle$ to the CSP.

Upon receiving the modification request, the CSP does the following steps to update the data sectors and the HVTs:

- It first replaces $\tilde{b}_{ijk}$ with $\tilde{b}'_{ijk}$ in each replica $\tilde{F}_i$.
- It generates a new HVT for each updated data block $\sigma'_{ij} = \sigma_{ij} + \Delta \sigma_{ij}$, $i \in [1,n]$.

Finally, the TPA checks the integrity of the updated data blocks after the dynamic operations.

Deletion: Suppose that a cloud user wants to delete sector $\{\tilde{b}_{ijk}\}_{i \in [1,n]}$ in all replicas. This requires the following steps:

- Compute the differences between the HVTs $\Delta\sigma_{ij} = u_k \cdot (-\tilde{b}_{ijk}) + h(ID_F || BN_j || VN_j || c_j || i)^{-1} \cdot h(ID_F || BN_j || VN'_j || c_j || i) \cdot d_{ID}$ for $i \in [1, n]$.
- Send a deletion request $\langle ID_F, Del, j, \{\Delta\sigma_{ij}\}_{i \in [1,n]} \rangle$ to the CSP.

After receiving the deletion request, the CSP takes the following steps to delete the data sectors in all replicas and update the corresponding HVTs:

- The CSP sets $\tilde{b}_{ijk}$ in each replica $\tilde{F}_i$ to empty.
- It calculates a new HVT for each updated data block $\sigma'_{ij} = \sigma_{ij} + \Delta\sigma_{ij}$.

Finally, the TPA checks the integrity of the updated data blocks after the dynamic operations.

4. SECURITY ANALYSIS

Theorem 1 (Correctness). If every entity performs honestly and correctly in the proposed scheme, the verification equation will hold.

Proof. The correctness of the ID-DRDA scheme is proved below:

$$\begin{aligned}
 e(\sigma, g) &= e\left(\sum_{(i,j,v_{ij}) \in Q} v_{ij} \sigma_{ij}, g\right) \\
 &= e\left(\sum_{(i,j,v_{ij}) \in Q} v_{ij} \left(\sum_{k=1}^s u_k \tilde{b}_{ijk} + h_{ij} d_{ID}\right), g\right) \\
 &= e\left(\sum_{(i,j,v_{ij}) \in Q} \sum_{k=1}^s u_k \tilde{b}_{ijk} + \sum_{(i,j,v_{ij}) \in Q} v_{ij} h_{ij} d_{ID}, g\right) \\
 &= e\left(\sum_{k=1}^s u_k \left(\sum_{(i,j,v_{ij}) \in Q} \tilde{b}_{ijk}\right) + \sum_{(i,j,v_{ij}) \in Q} v_{ij} h_{ij} d_{ID}, g\right) \\
 &= e\left(\sum_{k=1}^s R_k \mu_k + \sum_{(i,j,v_{ij}) \in Q} h_{ij} v_{ij} Q_{ID}, Y\right)
 \end{aligned}$$

where $h_{ij} = h(ID_F || BN_j || VN_j || c_j || i)$.

Theorem 2. The CDH problem can be solved if the probabilistic polynomial time adversary A can forge a valid HVT with non-negligible probability.

Proof. Assume that the adversary A can forge a valid HVT with non-negligible probability, then we can construct an algorithm C_1 which can solve the CDH problem. For input $(g, ag, bg) \in G_1$, C_1 will output abg by interacting with A .

Setup: Let G_1 and G_T be two multiplicative cyclic groups. C_1 executes the *Setup* algorithm to create the parameters and sets $Y = ag \in G_1$. It then returns $\{e, Y, f, \pi\}$ to A . Let $v \in Z_q^*$ be the index of the challenged cloud user.

H hash queries: A sends H hash queries to C_1 with the identity ID_o . If the record of ID_o is in the query list, C_1 returns the corresponding record to A . Otherwise, if $o \neq v$, C_1 returns a random value $H(ID_o) = r_o g \in G_1$ to A . If $o = v$, C_1 returns the value $H(ID_o) = r_o bg \in G_1$ to A . Finally, it adds the record to the query list.

Extract queries: A sends an identity ID_o to C_1 to query the corresponding cloud user's private key. If $o \neq v$, C_1 computes the private key $d_{ID} = r_o ag$ and forwards it to A . Otherwise C_1 outputs fail and aborts the game.

ReplicaGen queries: A selects a file F to query C_1 for the replicas with the identity ID_o . C_1 runs the *ReplicaGen* algorithm to generate a set of replicas $\tilde{F} = \{\tilde{b}_{ij}\}_{i \in [1, n], j \in [1, m]}$ and sends it to A .

TagGen queries: A queries C_1 for the HVT with $(ID_{File}, BN_j, VN_j, c_j, \{\tilde{b}_{ijk}\}_{i \in [1, n]})$ and the identity ID_o . If $o = v$, C_1 terminates the game. Otherwise, it will follow the steps below.

- 1) If the ID_o does not exist in H hash list, C_1 makes the query to H hash queries. Otherwise, C_1 returns the record in the list to A .
- 2) Then C_1 computes s random values $u_1, u_2, \dots, u_s \in G_1$. It computes the HVT $\sigma_{ij} = \sum_{k=1}^s u_k \cdot \tilde{b}_{ijk} + h_{ij} r_o Y$ and sends it to A , where $h_{ij} = h(ID_{File} || BN_j || VN_j || c_j || i)$.

Dynamic operations queries: A chooses a data sector to query C_1 for modification or deletion. If $o = v$, C_1 aborts the game. Otherwise, it follows the procedure of the modification operation or deletion operation on the cloud user side and returns the dynamic update request to A .

Output: In the end, adversary A outputs an HVT σ' of a new data block b' with identity ID' . The new data block and identity should not be queried at the Query phase. A wins the game if it satisfies three conditions below:

- 1) σ' is a valid HVT of the data block b'
- 2) The new data block b' and the cloud user's identity ID' are not queried in *TagGen* queries.
- 3) $ID' = ID_v$.

Therefore, C_1 can get the following equation:

$$\sigma' = \sum_{k=1}^s u'_k \cdot b' + h'_{ij} r' bY \quad (2)$$

Then, C_1 can obtain

$$abg = \frac{\sigma' - \sum_{k=1}^s u'_k \cdot b'}{h'_{ij} r'} \quad (3)$$

Finally, we can conclude that A can solve the CDH problem in probabilistic polynomial time with non-negligible probability, which breaks the difficulty of solving the CDH problem.

5. EVALUATION

In this section, we first analyze the performance of the proposed scheme by comparing it with existing dynamic multi-replica cloud auditing schemes. We further implement our scheme and Barsoum's scheme and consider them in the experimental evaluation. The variables and cryptographic operations are given in Table 1.

5.1. Performance Analysis

A qualitative comparison between the proposed scheme and existing multi-replica dynamic schemes is shown in Table 2. As seen, [21] and the proposed scheme are far more efficient than [12] and [20] considering the storage and communication costs. In [21] and our scheme, the storage overhead is independent of the number of replicas n . Moreover, transferring AAI from the CSP to the TPA during verification is eliminated. This is because [21] and our scheme adopt the MVT while [12] and [20] utilize the MHT and skip lists respectively. Different from [21], the proposed scheme introduces ID-based signature (IBS) and proposes a fine-grained dynamic update mechanism.

Table 3 shows a comprehensive performance comparison between the proposed scheme (referred to as ID-DRDA) and [21] (referred to as MB-PMDDP). It illustrates the theoretical analysis of computation, communication, and dynamic operation costs of the two schemes. Let the order of the group be 160 bits and the order of the base field be 512 bits. From Table 3, we can see that the ID-DRDA scheme has much less computation costs compared to the MB-PMDDP scheme. In the *TagGen* algorithm, the ID-DRDA scheme utilizes the cryptographic hash function H_z , which is much more efficient than the Map-To-Point hash function H_G in the MB-PMDDP scheme. In the *Response* algorithm, both schemes need to calculate a data proof P which contains the aggregated HVT σ and data sectors $\{\mu_k\}_{k \in [1,s]}$. However, the ID-DRDA scheme requires less computation since the optimized *Challenge* algorithm results in fewer data blocks needing to be checked. For the same reason, the ID-DRDA scheme also reduces the computation costs in the *Verify* algorithm.

The communication overheads of the ID-DRDA scheme are slightly higher than

those in the MB-PMDDP scheme in the *Challenge* algorithm since the ID-DRDA scheme sends one more random key $k_3 \in Z_q^*$. In the *Response* algorithm, the communication cost of the ID-DRDA scheme is greatly reduced compared to the MB-PMDDP scheme. This is because the communication cost of the MB-PMDDP scheme increases linearly with the number of replicas n .

The cost of dynamic operations of the two schemes is demonstrated in Table 3. As seen, the communication overheads of both schemes are linearly dependent on n . In the ID-DRDA scheme, modifying a data sector requires the cloud user to send a modification request containing the dynamic update information $Info = (ID_F, Mod, j)$, differences of HVTs $\{\Delta\sigma_{ij}\}_{i \in [1, n]}$ and new data sectors $\{\tilde{b}_{ijk}\}_{i \in [1, n]}$. Thus, the communication cost is $Info + 1024n + 160tn$. For deleting a data sector, it sends a deletion request that includes the dynamic update information $Info = (ID_F, Mod, j)$ and the differences of HVTs $\{\Delta\sigma_{ij}\}_{i \in [1, n]}$. Therefore, the communication cost can be given as $Info + 1024n$. The update request of the MB-PMDDP scheme includes dynamic update information $Info = (ID_F, Mod, j)$, aggregated HVT σ'_j , and new data blocks $\{\tilde{b}'_{ij}\}_{i \in [1, n]}$, which can be described as $Info + 1024 + 160sn$.

For the computation cost of dynamic operations, both schemes use nE_K operations to build distinct data replicas. The cloud user only needs to use $n(t + 1)E_G + 2nH_z + ntM_G + n(I_z + S_z)$ operations to calculate the differences of the HVTs whereas the computation cost of generating new HVTs in the MB-PMDDP scheme is $n(s + 1)E_G + nH_G + snM_G + (n - 1)M_G$. It is thus easy to see that the ID-DRDA scheme incurs far fewer computational overheads than the MB-PMDDP scheme.

5.2. Experimental Evaluation

We conduct the experiments on an Ubuntu 14.04.4 Linux system with an Intel Core i5 CPU at 2.30GHz and 4GB DDR3 RAM. All algorithms are implemented in the C language using the GMP library (version 6.1.0) and the PBC library (version 0.5.14). We use type A pairing parameters, where the order of the group is 160 bits and the order of the base field is 512 bits.

In our experiments, we compare the ID-DRDA scheme and the MB-PMDDP scheme from the perspective of proof response time, verification time and dynamic operations time. We do not consider the computation cost of the pre-processing procedure that creates distinct data replicas. This is because this procedure only needs to be done once for the entire life-time of the scheme and this may last for a very long time.

Proof response time: Figure 3 shows the comparison of proof response time of the ID-DRDA and MB-PMDDP schemes. It is easy to see that the proposed scheme is slightly more efficient than Barsoum's scheme. The proof response time of MB-PMDDP scheme is linearly related to the number of replicas whereas that of the ID-DRDA scheme is independent of the number of replicas. This is because

irrespective of how many copies exist in the cloud, the ID-DRDA scheme only challenges a certain number of data blocks across all replicas.

Verification time: The comparison of the verification time between two schemes is shown in Figure 4. The verification time of both schemes is almost independent of the number of data copies. However, the ID-DRDA scheme incurs much less verification time than the MB-PMDDP scheme since the proposed scheme has less exponential and multiplication operations in group G . Furthermore, the hash function H_z is much more efficient compared with the Map-To-Point hash function H_G .

Dynamic operation costs: We compare the computation cost of dynamic operations of the two schemes on the cloud user side in Figure 5 (10 replicas) and Figure 6 (20 replicas). As seen from the two figures, the dynamic operations in two schemes with 10 replicas or 20 replicas have a similar performance pattern. The computation time of dynamic operations in the MB-PMDDP scheme remains unchanged with the increase of the percentage of updated sectors in a single block, whereas the dynamic computation time of the ID-DRDA scheme increases linearly. However, irrespective of the percentage of updated sectors, the ID-DRDA scheme is much more efficient than the MB-PMDDP scheme. This is because our scheme reduces the computation costs by only calculating updated data sectors rather than all sectors in a data block.

As shown in Figure 5 and Figure 6, in the ID-DRDA scheme, the sector modification operation has slightly more computation time than the deletion operation due to the fact that the modification operation has one more step when calculating the differences between data sectors.

6. CONCLUSIONS

In this paper, we propose a novel ID-based replicated data auditing scheme (referred to as ID-DRDA) that supports fine-grained dynamic operations. The ID-based signature helps eliminate complex certificate management and the associated cost of certificate verification. By optimizing the *Challenge* algorithm, we improve the efficiency of the proposed scheme. We extend the map-version table (MVT) to improve the security of existing multi-replica cloud auditing schemes. We further extend the proposed scheme to support dynamic operations at not only the block level but also the sector level, thereby reducing the computation costs on the cloud user side. Under the harness of the CDH problem, the security analysis demonstrates that the proposed scheme is provably secure in the random oracle model. The theoretical and experimental performance analysis shows that the proposed scheme is more efficient than existing multi-replica auditing schemes.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under grant 61170221 and supported by the China Scholarship Council.

REFERENCES

- [1] Sookhak M, Gani A, Talebian H, et al. Remote data auditing in cloud computing environments. *ACM Computing Surveys* 2015; **47**(4): 1-34. DOI:10.1145/2764465.
- [2] Cloud computing vulnerability incidents: a statistical overview. Cloud Security Alliance. May 2013. Available at: https://cloudsecurityalliance.org/group/cloud-vulnerabilities/#_downloads. Accessed December 21, 2017.
- [3] Shamir A. Identity-based cryptosystems and signature schemes. *Lecture Notes in Computer Science* 1984; **21**(2): 47-53.
- [4] Ateniese G, Burns R, Curtmola R, et al. Provable data possession at untrusted stores. *Proceedings of the 14th ACM Conference on Computer and Communications Security*, 2007; 598-609. DOI: 10.1145/1315245.1315318.
- [5] Juels A, Kaliski BS. PORs: Proofs of retrievability for large files. *Proceedings of the 14th ACM Conference on Computer and Communications Security*, 2007; 584-597. DOI: 10.1145/1315245.1315317.
- [6] Ateniese G, Di Pietro R, Mancini LV, Tsudik G. Scalable and efficient provable data possession. *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, 2008; 1-10. DOI:10.1145/1460877.1460889.
- [7] Erway CC, Küpçü A, Papamanthou C, Tamassia R. Dynamic provable data possession. *ACM Transactions on Information and System Security (TISSEC)*, 2015; **17**(4): 213-222. <https://doi.org/10.1145/2699909>.
- [8] Wang Q, Wang C, Li J, Ren K, Lou W. Enabling public verifiability and data dynamics for storage security in cloud computing. *European Symposium on Research in Computer Security*, 2009; 355-370. DOI: 10.1007/978-3-642-04444-1_22.
- [9] Zhu Y, Wang H, Hu Z, Ahn GJ, Hu H, Yau SS. Dynamic audit services for integrity verification of outsourced storages in clouds. *Proceedings of the 2011 ACM Symposium on Applied Computing*, 2011; 1550-1557. DOI: 10.1145/1982185.1982514.
- [10] Sookhak M, Akhunzada A, Gani A, Khurram Khan M, Anuar NB. Towards dynamic remote data auditing in computational clouds. *The Scientific World Journal* 2014; 2014: 12. DOI: 10.1155/2014/269357.
- [11] Shen W, Yu J, Yang G, Zhang Y, Fu Z, Hao R. Access-authorizing and privacy preserving auditing with group dynamic for shared cloud data. *KSII Transactions on Internet and Information Systems* 2016; **10**(7).
- [12] Wang J, Chen X, Huang X, You I, Xiang Y. Verifiable auditing for outsourced database in cloud computing. *IEEE Transactions on Computers* 2015; **64**(11): 3293-3303. DOI: 10.1109/TC.2015.2401036.

- [13]Zhang J, Meng H, Yu Y. Achieving public verifiability and data dynamics for cloud data in the standard model. *Cluster Computing* 2017; **20**(1): 1-13. DOI: 10.1007/s10586-017-0804-9.
- [14]Yao G, Li Y, Lei L, Wang H, Lin C. An Efficient Dynamic Provable Data Possession Scheme in Cloud Storage. *Proceedings of the 11th International Conference of Green, Pervasive, and Cloud Computing*, 2016; 63-81. DOI: 10.1007/978-3-319-39077-2_5.
- [15]Curtmola R, Khan O, Burns R, Ateniese G. MR-PDP: Multiple-replica provable data possession. *Proceedings of the 28th International Conference on Distributed Computing Systems*, June, 2008; 411-420.
- [16]Mukundan R, Madria S, Linderman M, Rome NY. Replicated data integrity verification in cloud. *IEEE Data Engineering Bulletin* 2012; **35**(4): 55-64.
- [17]Barsoum AF, Hasan MA. On verifying dynamic multiple data copies over cloud servers. *IACR Cryptology ePrint Archive*, 2011; 447.
- [18]Zhang Y, Ni J, Tao X, Wang Y, Yu Y. Provable multiple replication data possession with full dynamics for secure cloud storage. *Concurrency and Computation: Practice and Experience* 2016; **28**(4): 1161-1173. DOI: 10.1002/cpe.3573.
- [19]Liu C, Ranjan R, Yang C, Zhang X, Wang L, Chen J. MUR-DPA: top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud. *IEEE Transactions on Computers* 2015; **64**(9): 2609-2622. <https://doi.org/10.1109/TC.2014.2375190>
- [20]Abo- alian A, Badr NL, Tolba MF. Integrity as a service for replicated data on the cloud. *Concurrency and Computation: Practice and Experience* 2017, **29**(4): 1-21. DOI: 10.1002/cpe.3883.
- [21]Barsoum AF and Hasan MA. Provable multicopy dynamic data possession in cloud computing systems *IEEE Transactions on Information Forensics and Security* 2015; **10**(3): 485-497. DOI: 10.1109/TIFS.2014.2384391.
- [22]Yu Y, Au MH, Mu Y, et al. Enhanced privacy of a remote data integrity-checking protocol for secure cloud storage *International Journal of Information Security* 2015; **14**(4): 307-318. DOI: /10.1007/s10207-014-0263-8.
- [23]Xu C, Zhang Y, Yu Y, Zhang X, Wen J. An efficient provable secure public auditing scheme for cloud storage *KSII Transactions on Internet and Information Systems* 2014; **8**(11): 4226-4241.
- [24]Jiang T, Chen X, Ma J. Public integrity auditing for shared dynamic cloud data with group user revocation. *IEEE Transactions on Computers* 2015; **65**(8): 2363-2373. DOI: 10.1109/TC.2015.2389955.
- [25]Zhao J, Xu C, Li F, Zhang W. Identity-based public verification with privacy-preserving for data storage security in cloud computing. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 2013; **96**(12): 2709-2716. DOI: 10.1587/transfun.E96.A.2709.
- [26]Wang H, Wu Q, Qin B, Domingo-Ferrer J. Identity-based remote data possession checking in public clouds. *IET Information Security* 2014; **8**(2): 114-121. DOI: 10.1049/iet-ifs.2012.0271.

- [27]Wang H. Identity-based distributed provable data possession in multicloud storage. *IEEE Transactions on Services Computing* 2015; **8**(2): 328-340. DOI: 10.1109/TSC.2014.1.
- [28]Yu Y, Zhang Y, Mu Y, Susilo W, Liu H. Provably secure identity based provable data possession. *International Conference on Provable Security*, 2015; 310-325. DOI: 10.1007/978-3-319-26059-4_17.
- [29]Yu Y, Xue L, Au MH, et al. Cloud data integrity checking with an identity-based auditing mechanism from RSA. *Future Generation Computer Systems* 2016; **62**(2016): 85-91. DOI:10.1016/j.future.2016.02.003.
- [30]Wang H, He D, Tang S. Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud. *IEEE Transactions on Information Forensics and Security* 2016; **11**(6): 1165-1176. DOI: 10.1109/TIFS.2016.2520886.
- [31]Shacham H, Waters B. Compact proofs of retrievability. *Journal of Cryptology* 2013; **26**(3): 442-483. DOI: 10.1007/s00145-012-9129-2.
- [32]Zhu Y, Hu H, Ahn GJ, Yu M. Cooperative provable data possession for integrity verification in multicloud storage. *IEEE transactions on parallel and distributed systems* 2012; **23**(12):2231-2244. DOI:10.1109/TPDS.2012.66.

TABLES

Table 1. Notations

Notation	Description	Notation	Description
H_z	Cryptographic hashing in Z_p	P	Bilinear pairing
H_G	Hashing to group G	c	Number of challenged data blocks
E_G	Exponentiation in G	s	Number of sectors
M_G	Multiplication in G	t	Number of updated sectors
A_z	Addition in Z_p	n	Number of replicas
M_z	Multiplication in Z_p	m	Number of data blocks
S_z	Subtraction in Z_p	I_z	Inversion in Z_p

Table 2. Comparison of existing multi-replica dynamic cloud auditing schemes

	Public auditing	ID-based	AAI required	Fine-grained Dynamics	Storage of Dynamic Data Structure
Wang [12]	Yes	No	Yes	No	$O(nm)$
Abo- alian [20]	Yes	No	Yes	No	$O(nm)$
Barsoum [21]	Yes	No	No	No	$O(m)$
The proposed scheme	Yes	Yes	No	Yes	$O(m)$

Table 3: Comparison of the ID-DRDA scheme and the MB-PMDDP scheme

Costs		ID-DRDA	MB-PMDDP
Computation	TagGen	$nm(s+1)E_G + nmH_Z + nmsM_G$	$nm(s+1)E_G + nmH_G + nmsM_G + (n-1)M_G$
	Response	$cE_G + (c-1)M_G + (c-1)sA_Z + csM_Z$	$cE_G + (c-1)M_G + (c-1)snA_Z + csnM_Z$
	Verify	$2P + (s+1)E_G + cH_Z + sM_G + cM_Z + (s-1)A_Z$	$2P + (c+s+1)E_G + cH_G + (c+s-1)M_G + s(n-1)A_Z$

Communication	Challenge		$480 + \log_2 c$	$320 + \log_2 c$
	Response		$1024 + 160s$	$1024 + 160sn$
Dynamic Operation	Communi- cation	Mod Request	$Info + 1024n + 160tn$	$Info + 1024 + 160sn$
		Del Request	$Info + 1024n$	
	Cloud user computation		$nE_K + n(t + 1)E_G + 2nH_z + ntM_G + n(I_z + S_z)$	$nE_K + n(s + 1)E_G + nH_G + snM_G + (n - 1)M_G$