**Certificateless Public Integrity Checking of Group Shared Data on Cloud Storage**

Jiguo

**Abstract**

Cloud storage service supplies people with an efficient method to share data within a group. The cloud server is not trustworthy, so lots of remote data possession checking (RDPC) protocols are proposed and thought to be an effective way to ensure the data integrity. However, most of RDPC protocols are based on the mechanism of traditional public key infrastructure (PKI), which has obvious security flaw and bears big burden of certificate management. To avoid this shortcoming, identity-based cryptography (IBC) is often chosen to be the basis of RDPC. Unfortunately, IBC has an inherent drawback of key escrow. To solve these problems, we utilize the technique of certificateless signature to present a new RDPC protocol for checking the integrity of data shared among a group. In our scheme, user's private key includes two parts: a partial key generated by the group manager and a secret value chosen by herself/himself. To ensure the right public keys are chosen during the data integrity checking, the public key of each user is associated with her unique identity, for example the name or telephone number. Thus, the certificate is not needed and the problem of key escrow is eliminated too. Meanwhile, the data integrity can still be audited by public verifier without downloading the whole data. In addition, our scheme also supports efficient user revocation from the group. The security of our scheme is reduced to the assumptions of computational Diffie-Hellman (CDH) and discrete logarithm (DL). Experiment results exhibit that the new protocol is very efficient and feasible.

Index Terms-Remote data checking, cloud storage, certificateless signature, data shared in group

**1. InTRODUCTION**

LOUD storage service offers user an efficient way to - share data and work as a team. Once someone of the team uploads a file to the server, other members are able to access and modify the file by Internet. Many real applications such as Dropbox for Business [1] and TortoiseSVN [2] are used in many companies for their staff to work together. The most important problem of such applications is whether the cloud server provider (CSP) can ensure the data to be kept intact [3]. In fact, the CSP is not fully trustworthy and the failure of software or hardware is inevitable in some way, so serious accidents of the data corruption may occur at any time. Therefore, the user needs to audit the CSP to confirm the data on the cloud server is original.

To ensure the integrity of stored data, a great number of RDPC schemes are proposed [4], [5], [6], [7], [8], [9], [10], , . In these

* J. Li is with the College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China 350117, and also with State Key Laboratory of Cryptology, P.O. Box 5159, Beijing, 100878, China, Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, China, College of Computer and Information, Hohai University, Nanjing, China

[E-mail:ljg1688@163.com](mailto:E-mail:ljg1688@163.com).

* H. Yan is with the College of Computer and Information, Hohai University, Nanjing, Jiangsu 211100, China. E-mail: [pxy\_hao@163.com](mailto:pxy_hao@163.com).
* Y. Zhang is with the College of Mathematics and Informatics, Fujian Normal University, Fuzhou, Fujian 350117, China, and with the College of Computer and Information, Hohai University, Nanjing, Jiangsu 211100, China. [E-mail:zyc\_718@163.com](mailto:E-mail:zyc_718@163.com).

Manuscript received 16 Apr. 2017; revised 22 Dec. 2017; accepted 2 Jan. Date of publication ; date of current version 3 Feb. 2021 .

(Corresponding author: Jiguo Li.)

Digital Object Identifier no. 10.1109/TSC.2018.2789893 schemes, each data block generates an authentication tag which is bound with the block. By checking the correctness of the tags, the verifier is able to learn the status of the data. However, most of these schemes only focus on checking the integrity for personal data [4], [5], [6], [7], [8], [9], [10], [11], , [31], [32], which is not valid under the situation of data shared in a group. When data is shared among multiple users, some new challenges appear which are not well solved in the RDPC schemes for personal data. For example, block tags may be generated by any group user, and different group user will output different tags even if the block is the same one. Moreover, when a group user updates a block, it should regenerate the tag again. When auditing the data integrity, all the authentication tags generated individually need to be aggregated and the information of all the generators for these tags will be involved in. It brings great complexity for the checking scheme. Furthermore, the group is dynamic, any group member may initiatively leave or be fired from the group at any time, so the user revocation is also an important problem that must be addressed. More specifically, once a user is revoked, he should not be allowed to access or modify the data and all his public/private keys are invalid. Under this situation, it is impossible to check the correctness of the tags made by revoked user. Thus, all the tags made by revoked user should be renewed by other normal user. The traditional method is to download the blocks signed by revoked user from the CSP, calculate the new tags and upload the new tags to the cloud again. It will increase heavy computation and communication cost for the normal user. Therefore, this task should be performed by the CSP rather than the normal user. How to design an efficient and secure method to outsource the task is a challenge issue. Besides, public verification is an attractive feature of the data integrity checking work. That is, the integrity of shared data can be verified by not only the data owner but also everyone who is interested in the cloud data. It is very important for RDPC protocol to support public verification under current open environment.

Until now, lots of schemes [22], [23], [24], [25], [26], [27], [28] have been presented for the integrity verification of data shared in group. However, most of existing RDPC schemes [22], [23], [24], [25], [26], [28] are based on PKI. Although PKI is widely used and occupies an important position in public key cryptography, there are still some security threats in it. For example, the security of PKI is based on the trustworthy of certificate authority (CA), but it is not an easy work to ensure the trustworthiness of CA. Besides, the management of certificate such as distribution, storage, revocation and verification is also a big burden. To avoid these problems, some ID-based RDPC schemes [27], [28] are proposed. Unfortunately, ID-based RDPC schemes suffer from key escrow problem. Namely, the private key generator (PKG) generates all the private keys for the users. If is untrusted, the scheme is not secure either. Thus, ID-based RDPC schemes may be restricted to small, closed settings. Compared with PKI and IBC, certficateless cryptography [33] solves the problems of certificate management and key escrow at the same time. To construct certificateless RDPC scheme is a good method for cloud data integrity checking.

**1.1. Motivation and Contributions**

In this paper, we mainly focus on the integrity checking for data shared within a group. Suppose there is a scenario that a software engineer starts an open source project and calls on volunteers from the world to join the project. They work as a temporary team. All the codes of the project are stored on certain cloud server so that all the team members upload and modify the source code by Internet. The team may be very big, so it should be set up and managed efficiently. The volunteers may leave the team at any time, so the problem of user revocation from the should be considered. The most important thing is that there need some way to guarantee the integrity of source codes on cloud sever.

Motivated by such requirement, we propose a new RDPC scheme for data shared in a group. Different from previous work, our scheme is based on the certifcateless signature technique to avoid the problems of certificate management and key escrow. In our scheme, the group creator generates the partial key for each group user on behalf of key generation centre. Each user selects a secret value privately. The private key of each group user contains two parts: a partial key and a secret value. All the data blocks are signed by group user to get corresponding authentication tags. During the data verification, all the tags are aggregated to decrease the computation and communication cost. Based on and DL assumptions, we prove the security of our scheme. Besides, our scheme supports public verification and efficient user revocation. We implement our scheme and perform some experiments. The experiment results indicate that our scheme has good efficiency.

**2. Related Work**

The first RDPC protocol for remote data checking was proposed by Deswarte et al. [4], in which a RSA-based hash function was utilized to generate the authentication tag of the data. Following it, a great number of provable data possession (PDP) [5] and proof of retrievability (POR) [29] schemes were proposed to solve the issue for data integrity verification.

Ateniese et al. [5] first presented PDP model and initially introduced the technique of probabilistic integrity checking for the remote data. However, the first PDP scheme was only suitable for static data. To meet dynamic operations of the data block, Ateniese et al. [6] proposed another scalable and efficient PDP scheme by symmetric encryption, which supported block appending, updating and deleting. Sebé et al. [7] presented a PDP protocol based on the hard problem of factoring large integers. Erway et al. [8] utilized the authenticated skip list to provide a fully dynamic PDP scheme, which supported data owner to insert, append, modify and delete data blocks.

Based on the technique of random masking and the homomorphic linear authenticator, Wang et al. [9] presented a public verification PDP scheme with property of privacy-preserving. To support the public auditability and data dynamics, Wang et al. [10] utilized merkle hash tree (MHT) to present a dynamic scheme for cloud data checking. The scheme was fully dynamic and allowed anyone to verify the file integrity with public keys. MHT was also used in schemes [11], [12] to implement data dynamic. However, due to the computation complexity of the MHT, this scheme caused heavy computation cost and communication cost. To overcome this shortcoming, Yang and Jia [13] introduced a linear index table to support data dynamic. Yan et al. [14] further optimized the implementation of linear index table and provided an efficient RDPC scheme. Feng et al. [15] presented a public remote integrity checking scheme, which could protect the user identity on file level to reduce the storage and communication cost.

Zhu et al. [16] provided a cooperative PDP scheme for the multi-cloud setting, in which the data blocks were stored on different cloud servers. To improve the security, Wang [17] proposed another identity-based PDP scheme for multicloud setting without certificate management. Recently, Wang et al. [18] presented an incentive and unconditionally anonymous identity-based public PDP scheme. In order to reduce the computation cost of data owner, Wang et al. [19] presented a proxy-oriented PDP scheme which moved the work of tag generation from data owner to proxy. To address the problem of key escrow and certificate management, two PDP scheme based on certificateless [20] and certificatebased cryptography [21] were proposed respectively.

All the schemes mentioned above focused on the integrity verification for personal data. In 2012, Wang et al. [22] proposed a protocol for checking the integrity of data shared in a group. They utilized the technique of group signature to generate each authentication tag so as to preserve the tag generator's privacy. Wang et al. [23] proposed another PDP scheme for group data which supported the group user's joining and leaving. Based on broadcast encryption and group signature techniques, Liu et al. [24] provided a PDP scheme for group data. To improve the efficiency, Wang et al. [25] presented another scheme based on ring signature technique. However, these two schemes didn't solve the problem of user revocation. To address this issue, Wang et al. [26] used proxy re-signature technique to propose a new scheme with user revocation. Yu et al. [27] presented a PDP scheme without paring, which also supported dynamic group. Yuan and Yu [28] proposed a PDP scheme based on polynomial-based authentication tags, which aimed to solve the problem of multi-user modification for blocks. All these schemes rely on traditional PKI mechanism which has security risk and big burden for certificate management. Besides, schemes of [27], [28] also suffer from the problem of key escrow. Thus, there still exists big limitation for the schemes to be used in real applications.

PoR [29] is another direction for auditing the integrity of remote data on cloud server. To enhance the efficiency, Shacham and Waters [30] proposed two compact PoR schemes based on technique of short signature [34]. Furthermore, lots of PoR schemes [31], [32] were presented for higher efficiency or better security.

With the development of cloud computing, how to share the data from the cloud server attracts more concern. In order to ensure the security and privacy, and obtain flexibly finegrained file access control, attribute based encryption (ABE) scheme [35], [36], [37], [38], [39], [40], [41], [42] as a new cryptographic primitive was presented and applied in cloud storage system. In ABE scheme, an encryptor associates ciphertext with a set of attributes. The authority issues the users' different private keys that are associated with access policy on attributes. Li et al. [35], [36] modeled collusion attack executed by existing users with revoked users and constructed efficient CP-ABE schemes with user revocation. Recently Wang et al. [37] provided an anonymous distributed fine-grained access control scheme with verifiably outsourced decryption in public cloud. In ABE scheme, sensitive documents should be encrypted prior to outsourcing for privacy requirements, which hinders efficient query processing like keyword-based document retrieval. In order to address this issue, Li et al. [38], [39] presented ABE schemes with keyword search function. ABE schemes may have sensitive information and leak the privacy of the encryptor because access policy is sent to the decryptor along with the ciphertext. ABE scheme with hidden access policy [40] and privacy preserving scheme [41] can overcome the above issue. To ensure cloud service provider correctly stores ciphertext, some ABE schemes [42], [43] with efficient verifiable outsourced decryption were proposed. Recent research mainly concentrates on verifiability of outsourced decryption for the authorized users. How to guarantee the correctness of outsourced decryption for unauthorized users still remains a challenging problem. Recently Li et al. [44] presented a full verifiability for outsourced decryption in ABE, which can simultaneously verify the correctness of transformed ciphertext for the unauthorized users and authorized users.

**2.1. Organizations**

The remainder of the paper is organized as follow. Section 2 introduces the preliminaries and Section 3 gives the detailed construction of our scheme. The security proof and performance analysis are demonstrated in Sections 4 and 5. We conclude the paper in Section

**3. PRELIMINARIES**

In this section, we introduce the preliminary knowledge used throughout this paper.

**3.1. Bilinear Maps**

and are two multiplicative cyclic groups with large prime order is a generator of is a bilinear map with the following properties:

1. Computability: for can be computed efficiently.
2. Bilinearity: for , it has .
3. Non-degeneracy: such that .

**3.2. Complexity Assumption**

Definition 1 (Computational Diffie-Hellman (CDH) problem). Suppose is a multiplicative cyclic groups. is a generator of . Given the tuple with the unknown elements , the problem is to compute .

Definition 2 (CDH assumption). For any probabilistic polynomial time algorithm , the advantage for to solve the problem in is negligible, which can be defined as:

Definition 3 (Discrete Logarithm (DL) problem). Assume that is a multiplicative cyclic group. is a generator of . Given the values of with unknown element , the DL problem is to output a.

Definition 4 (DL assumption). For any PPT algorithm , the advantage for to solve the problem in is negligible, which can be defined as:

Notably, denotes a negligible value in the above definitions.

**3.3. System Model**

Referring to the papers [25], [26], [27], [28], the system model of our scheme is composed of three major entities: user group, cloud service provider (CSP) and public verifier. The user group includes numbers of users, who can upload, access and update the data shared within the group, and honestly execute the protocol. Without loss of generality, the original creator of the group plays the role of group manager, who sets up the system and generates partial keys for general group users. CSP owns powerful storage and computational abilities to supply cloud users with data storage service. In our scheme, the shared data is divided into many blocks and each block is attached with an authentication tag. Thus, the CSP stores all the blocks and the corresponding tags for cloud user. The data verifier is a person who checks the integrity of the data on CSP. Due to the feature of public verification, anyone could be the verifier in our scheme.

The Fig. 1 shows the relationships and the interactions among the three entities of the system. As most previous works [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], we assume the CSP is semi-trusted. That is, the CSP can honestly execute the protocol, but may cheat the verifier about the incorrectness of the data so as to keep its reputation or get extra benefits.

**3.4. Outline of RDPC Scheme**

Definition 5. The RDPC scheme is composed of eight algorithms:

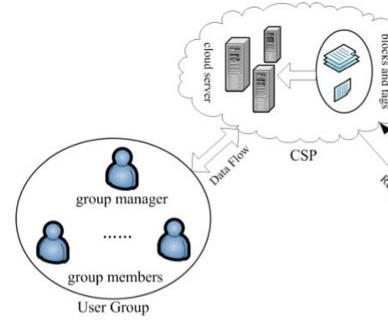


Fig. 1. System model of our scheme.

Setup. This algorithm inputs the security parameter , outputs the master key and the system public parameters params. This algorithm is executed by the manager of the group.

PartialKeyGen: This algorithm is performed by the group manager to extract the partial key for group users. It takes the master key and the identity of the user as the inputs, outputs the 's partial key .

SecretValueGen. The group user executes this algorithm to generate the secret value. The algorithm randomly selects as the secret value for the user . Thus, the private key of group user contains two components: secret value and partial key.

PublicKeyGen. The algorithm is performed by group user to generate the public key. It inputs the 's secret value and outputs the s public key .

TagGen. The group user runs this algorithm to generate authentication tag for a block. This algorithm inputs the 's partial key , the secret value and the block , and outputs the tag of the .

Challenge. The algorithm is performed by the verifier. It inputs the count of challenged blocks and outputs the challenge information chal.

ProofGen. CSP runs this algorithm to obtain the possession proof. It takes the challenged file , tag set of all the blocks and the challenge information chal as inputs, and outputs the integrity proof .

Verify. The verifier runs this algorithm to verify the proof . It inputs the proof , the challenge information chal and the public key set of all the group users. If is correct, the algorithm outputs 1 , otherwise it outputs 0 .

**4. Security Model**

Since our new scheme introduces the idea of certificateless cryptography [33], we consider three adversaries namely and in our security model. Both and try to forge the tag of the block. But the differences of them are that can't access the master key of the system but can substitute the user's public key with any other value, and is able to get the master-key but can't substitute the user's public key. The aims to forge the data integrity proof to cheat the verifier. Referring to [20], [26], we define the security of our scheme by three games which involves a challenger and the adversaries and respectively. Three security games are described as follows.

Game I. This game is played by and .

Setup. C performs the Setup algorithm to obtain the master key and the public parameters. keeps the master key privately and sends the public parameters to .

Queries. can make different queries to for polynomial times. responses the queries to as follows.

1. Hash Query. adaptively makes hash function queries to responses the hash values to .
2. Partial Key Query. adaptively chooses different and submits it to for querying the partial key of the ID. executes the PublicKeyGen algorithm to obtain the partial key for the and sends it to .
3. Secret Value Query. adaptively chooses different and submits it to for querying the secret key of the ID. C runs the SecretValueGen to generate the secret value for this and sends it to .
4. Public Key Query. adaptively selects different and submits it to for querying the public key of the ID. C performs the algorithm of PublicKeyGen to compute the public key for the and sends it to .
5. Public Key Replacement. can repeatedly choose a value to substitute the public key of any ID.
6. Tag Query. adaptively chooses the tuple and submits it to for querying the tag of the block generated by the ID. By the algorithm TagGen, generates the tag for and sends the tag to .

Forge. Finally, outputs a forged tag for the with the identity and the public key .

If the conditions listed below are satisfied, wins the game:

1. The forged tag is valid for the block with the identity and the public key .
2. doesn't query the private key of the user with .
3. does not query the partial key of the and replace the public key of the simultaneously.
4. does not make the tag query for the with the identity and the public key .

Game II. This game is played by and .

Setup. performs the Setup algorithm to obtain the master key and the public parameters. sends the public parameters and the master key to .

Queries. can make different queries to for polynomial times. responses the queries to as follows.

1. Hash Query. adaptively asks hash function queries to responses the hash values to .
2. Secret Value Query. adaptively chooses different and submits it to for querying the secret value of the ID. C executes the SecretValueGen to obtain the secret value for the and sends it to .
3. Public Key Query. adaptively chooses different and submits it to for querying the public key of the ID. executes the algorithm of PublicKeyGen to compute the public key for the and returns it to .
4. Tag Query. adaptively chooses the tuple and submits it to for querying the tag for the block generated by the . By the algorithm TagGen, generates the tag for and returns it to .

Forge. Finally, outputs a forged tag for the with the identity . If the conditions listed below are satisfied, wins the game:

1. The forged tag is valid for the block with the identity
   1. doesn't query the secret value for the .
   2. doesn't query the tag for the with the identity .

Definition 6. If for any PPT adversary or , the probability that wins the Game I and Game II is negligible, the single tag of block is existentially unforgeable.

Game III. The challenger interacts with the adversary in this game. Here, is regarded as the untrusted CSP. If the data is corrupted, tries to cheat the verifier that the data is kept intact. From the Definition 6, we know that any adversary can't forge the tag of single block without the right private key. Thus, in this game we just focus on the issue that whether can forge the integrity proof on incorrect data to pass the verification. Inspired by [26], the procedure of game III is defined below.

Setup. generates the public parameters, the master key and private keys for all users. keeps all the private keys and the master key privately, but sends the public parameters to .

Tag-Query. adaptively chooses the tuple and sends it to for querying the tag of generated by the . generates the tag of and returns it to by the algorithm TagGen.

Challenge. generates a random challenge chal, sends chal to and requests to give the corresponding data possession proof for chal.

Forge. For the challenge chal, generates a proof and gives it to . If can pass the integrity verification and the block information in is wrong, wins the game.

Definition 7. If any PPT adversary can win the Game III about a set of blocks with only negligible probability, the probability to forge the integrity proof without correct data is negligible.

**5. OUR NeW SCHEME**

**5.1. Construction of our Scheme**

We suppose there are users in one group. Let represent the unique identity of the user for . Without loss of the generality, we set to be the group manager, who will set up the system and generate the partial secret keys for other users. We suppose the file is split into blocks, represented as , and each block is an element in . The detailed construction for the scheme is demonstrated as follows:

Setup. Let and be multiplicative cyclic groups with primer order is a bilinear map and is a generator of . selects two secure hash functions and , a pseudo-random permutation (PRP) and a pseudo-random function (PRF) randomly chooses as the master secret key and calculates keeps the master secret key privately and publishes the public system parameters params , PartialKeyGen. When receiving the identity of the group user , the group manager computes the and returns the to the as the partial private key. It is noted that the is unique identity in the group.

SecretValueGen. randomly chooses and sets as the secret value and keeps it private.

PublicKeyGen. uses the secret value to compute the public key .

TagGen. Each user in the group can access the blocks and generate tags for them with his private key. Suppose the user wants to generate the tag for the block , the equation for computing tag is , where , and denotes the unique file identity. The group manager maintains a public log file which stores the information of the and the identity of tag generators. After adding or modifying data blocks, the user should add or update the corresponding and the identity of tag generators in the log file. The user uploads the blocks and the tags to CSP. CSP can check the validation of the tag by

Challenge. The verifier randomly picks the count of challenged blocks and the two random values , where are seeds for PRP and PRF respectively. The verifier sends chal to the CSP as the challenge information.

ProofGen. When the CSP receives the challenge information chal , the CSP computes the set , , in which for . According to tag generator of each challenged block, the CSP splits the set into subsets in which the subset is the collection of the tags generated by the user . Let denote the count of the element in . We can get and for . For each subset , the CSP computes by and . The CSP returns as the final proof, where .

Verify. When receiving the proof , the verifier calculates and divides the challenge set according to the tag generators as the CSP does. By searching in the file, the verifier gets all of the challenged blocks. Finally the verifier checks whether the Equation (2) holds:

If the Equation (2) holds, verifier outputs 1 , otherwise outputs 0 . If the CSP and the verifier honestly run the protocol, we can check the correctness of the protocol by the following equality: theorems.

**5.2. Supporting User Revocation**

If any user leaves the group, the secret keys and public key of should be claimed to be invalid. Thus, the secret key of should be removed from the tags generated by the . Otherwise, the validation of these tags can't be verified and the integrity of the file can't be checked either. The traditional method for regenerating tags of the revoked users is to download the blocks from CSP, re-generate the tags and then upload the new tags again. It inevitably increases the communication and the computation cost for the users. Therefore, to relieve these potential overhead, our scheme designs to outsource the tag update work to the CSP and reduces the communication cost. We use the algorithm ReTagGen to update the tags generated by revoked users. The construction of ReTagGen is shown as follows, in which the is the revoked user, is another valid user in the group. Notably, is the group creator and manager, we consider should not be revoked.

ReTagGen: This algorithm contains three interactions among and CSP. We suppose there is no collusion among and CSP, and the secure channel is used during the interactions. Besides, it is required that and CSP are online simultaneously among the revocation procedure.

1. CSP randomly chooses a value and sends to by secure channel.
2. computes and sends to .
3. calculates and sends to CSP.
4. When receiving the , CSP calculates . Using the Equation (1) to retrieve and check all the tags-block pairs generated by . Then the CSP transforms the tags for the block as

,where is the valid tag of for the generator .

**6. SeCURITY PROOF**

通过以下三个定理证明了该方案是安全的。

**6.1. Security Proof**

定理1。如果一个概率多项式时间敌手 在时间t内，最多分别以次发起-哈希询问、部分私钥询问、秘密值询问、公钥询问、-哈希询问和标签询问后，以的优势赢得2.5节中定义的游戏I，那么就有一个模拟者 可以在时间内，以 的概率攻破CDH问题。

证明。给定一个CDH问题的实例 . 如果敌手 以不可忽略的优势赢得游戏I，模拟者 就能通过的能力以不可忽略的概率计算出的值。模拟了与的每个交互步骤，如下所示。

初始化。 产生公开参数，并设置 ，其中主密钥是隐匿的未知值 。

-哈希询问。 自适应地对任意身份 做-哈希询问。 维护一个-哈希询问的列表 。 如果 存在于 检索元组 并返回 给。否则， 随机选择一个值 并抛出一个硬币 。假设 的概率是 ，则 的概率是 .。如果 计算。如果 设置 回复 给 并将运足 添加到中。

部分私钥询问。 自适应地执行对任意身份的部分私钥询问。 首先检查是否存在列表中。如果不存在，对身份做-哈希询问。注意，如果，中止。还会维护一个部分私钥询问的列表。

1. 如果 存在于 检查是否对应的值 。如果 从中搜索元组，对于，计算 并更新元组中的 。如果 终止。对于 直接得到 。 然后 返回 给。
2. 如果 不存在与 从中搜索元组 ，如果，计算 。如果 终止。最后， 返回 给，并添加心得元组 到中。

秘密值询问。 自适应地对任意身份 做秘密值询问。检查是否元组 已经在 中存在。如不存在， 对做 -哈希询问。检查 是否在 中存在。

1. 如果存在于 检查是否对应的值 .。如果 随机选择一个值 并设置 , 。然后 使用 and 更新元组。 如果 在元组中检索它并返回给 。
2. 如果 不存在于 随机选择一个值 并设置 , 。最后， 添加元组 到中并返回 给。

公钥询问。 自适应地对任意身份 做公钥询问。

1. 如果元组 在 中存在， 检查是否 。 如果 ， 随机选择一个值 并设置 。 返回 并更新元组中的 。如果 ， 直接把它返回给 。
2. 如果 中不包含元组 , 随机选择 并设置 把元组 , 添加到 并发送 给 .

公钥替换。 自适应地用 进行公钥替换。

1. 如果元组 存在于 ，将该元组更新为 。
2. 如果 中不包括该元组 , 添加一个新的元组 到中。

-哈希询问。 自适应地执行-哈希询问， 对于 同样为-哈希询问维持一个列表 。 如果 中包含了 检索 并返回 给。 否则, 选择一个随机值 并发送 给。然后 将 插入到 。

标签询问。 自适应地执行标签询问，使用 首先检查是否 存在列表 中，对于。 如果 终止。否则， 从中获得 、 并从中获得 。然后 通过TagGen算法计算 的标签并返回给 。

伪造。最后， 输出一个元组 , ) 其中 是由身份使用公钥对块 伪造的标签。

分析。如果 赢得游戏I， 可以根据等式（1）得到 。 在中检索元组 。如果 ， 终止并输出“失败”。 否则， 在 中检索 以及在中检索 。 根据上面提到的验证等式， 可以得到 . 。 因此我们可以得到 。 现在，我们评估 输出正确的结果的概率。显然，如果 在上述的过程中不终止， 和 进行了完美的交互。 另外我们可以知道 -哈希询问、秘密值询问、公钥询问、公钥替换、 哈希询问都可以被无意外地完美执行。 的终止只发生在部分私钥询问和标签询问的过程中。因此 与完美模拟交互过程而不终止的概率 高于 。因此， 以 的高铝输出 正确的值。相应的时间代价为 。□

定理2： 如果一个概率多项式时间敌手 在时间t内最多分别以次发起-哈希询问、秘密值询问、公钥询问、-哈希询问和标签询问后，以 的概率赢得节2.5中定义的游戏II，那么就有一个模拟者 可以在时间 内，以 的概率攻破CDH问题。

证明：给定一个CDH实例 。 如果敌手 以不可忽略的优势赢得游戏II， 模拟者 可以通过的能力以不可忽略的概率计算出 的值。at non-negligible probability by the capability of 模拟了与的每个交互步骤如下。

初始化： 随机选择一个值 作为主密钥并产生公开参数。 返回主密钥和公开参数给 。

-哈希询问： 自适应地对任意身份发起 -哈希询问。 为-哈希询问维护一个列表 。如果 存在于 检索元组 并返回 给。 否则，选择一个随机值 并计算 回复 给 并添加 到中。

秘密值询问：因为 知道主密钥, 不需要进行部分私钥询问。 自适应地对任意身份 做秘密值询问。 maintains a list for the SecretValue-Query. first checks whether exists in .

1. If the doesn't exist in randomly chooses a value and flips a coin . Suppose the probability of is , so the probability of is . If calculates and adds the tuple to . Then outputs 'fail' and aborts. If calculates and adds the tuple , ) to . Then returns to .
2. If the exists in checks the corresponding value . If outputs 'fail' and abort. Otherwise, retrieves the and returns it to . (It is sure that exists in when ).

PublicKey-Query. adaptively executes the PublicKey- Query for any identity .

1. If the tuple exists in returns to directly.
2. If the does not contain the tuple , randomly selects a value and flips a coin . Suppose the probability of is , so the probability of is . If calculates . If sets and calculates adds a new tuple into and returns to . for . If aborts and outputs 'fail'. Otherwise, calculates the and gets the from , the from . Then computes the tag for by the algorithm TagGen and returns it to .

Forge. Finally, outputs a tuple in which is the forged tag on the block for the identity .

Analysis. If , wins the game II, can get according to the verification Equation (1). Then retrieves the (ID', ) from . If aborts and outputs 'fail'. Otherwise, gets the from , from and from . According to the above equation, obtains . . Thus, we can derive . Now, we evaluate the probability for outputting the right result. Obviously, if does not abort in the process above, performs a perfect interaction simulation with . Further, we can see that the -Query, PublicKeyQuery and -Query are executed completely without additional conditions. The abortion of only happens in the procedures of SecretValue-Query and Tag-Query. Therefore, the probability that perfectly simulates the interactions with without abortion is higher than . Thus, outputs the right value with the probability . The corresponding time cost is

Theorem 3. If the DL assumption holds, the adversary wins the Game III only at negligible probability.

Proof. Let the challenge information be . If outputs the integrity proof and wins the Game III at non-negligible probability, we can get the verification equation:

where denotes the count of the group subset in the challenge. Assume the real proof for chal is , we also gets verification equation: , Because wins the game III, there exists but . According to the two equations above, it has . Define for each , we know that . Based on this conclusion, the DL problem can be solved as follows. Give two element that , we will compute the value . Let , in which and are randomly selected from . We can get the following equality



at least one is not zero. is a random value in , so the probability of is only . Therefore, we can output the right value of with non-negligible probability .

**6.2. Error Detection Probability**

Suppose of blocks on CSP have been tampered. CSP randomly selects of blocks to generate the proof. In order to detect the data corrupt, must be satisfied. Therefore, the probability of error detection is equal to the probability of . Let denote error detection probability, it has . It indicates that: . From the above equation, we can see that greater number of challenged blocks will cause higher error detection probability. By the analysis of [5], for 1 percent tampered blocks in the file, 300 challenged blocks will make and 460 blocks will make . Thus, our scheme can achieve a high error detection probability.

In this section, we give the performance analysis and experiment results of our scheme.