**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

**2 储备知识**

在本节中，我们将介绍在本文中使用的储备知识。

**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:

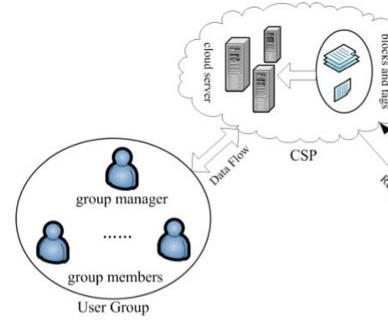


Fig. 1. System model of our scheme.

2.4 Outline of RDPC Scheme

定义5：RDPC方案由八种算法组成：

Setup: 输入安全参数，输出主密钥，和系统公开参数 。该算法由群组的管理员执行。

PartialKeyGen: 由群组管理员执行，为群组用户生成部分私钥。以主私钥和用户的身份作为输入，输出的部分私钥。

SecretValueGen: 群组用户执行这个算法来生成秘密值。算法随机选择作为用户的秘密值。因此，群组用户的私钥由两部分组成：秘密值和部分私钥。

PublicKeyGen：该算法由群组用户执行来生成公钥。输入用户的秘密值，输出的公钥。

TagGen：群组用户执行此算法，为一个数据块生成认证标签。输入的部分私钥，秘密值，和数据块，输出的标签。

Challenge：该算法由验证者执行。输入挑战的块数，输出挑战消息。

ProofGen：CSP执行该算法，获得占有证据。以被挑战的文件，所有数据块的标签集合，以及挑战信息作为输入，输入完整性证据。

Verify：验证者执行该算法来验证完整性证据。它以证据，挑战信息和所有群组用户的公钥集合作为输入。如果是正确的，输出1，否则输出0。

**2.5 安全模型**

由于我们的新方案引入了无证书密码学[33]的思想，因此我们在我们的安全模型中考虑了三个对手，即、和。和都试图伪造数据块的标签。但不同之处在于，不能访问系统的主密钥，但可以用任何其他值替换用户的公钥，而能够获得主密钥，但不能替换用户的公钥。的目的是伪造数据完整性证据来欺骗验证者。参考[20]，[26]，我们通过三个游戏来定义我们的方案的安全性，其中分别涉及到一个挑战者和对手，和。对三个安全游戏的描述如下：

游戏I：由和来完成。

初始化：运行Setup算法，获得主密钥和公开参数。自己保留主密钥，把公开参数发送给。

询问：可以在多项式时间内对做不同的询问。按照如下方式对的询问进行回应：

哈希询问：适应性的向做哈希询问。向回应哈希值。

部分私钥询问。自适应的选择不同的并提交给来询问的部分私钥。执行PartialKeyGen算法获得的私钥并发给。

秘密值询问。自适应地选择不同的并提交给来询问的秘密值。运行SecretValueGen算法生成秘密值并发送给。

公钥查询：自适应地选择不同的并提交给来询问的公钥。运行PublicKeyGen算法计算公钥并发送给。

公钥替换：重复地选择一个值来替换任意的公钥。

标签询问：自适应地选择元组并提交给，以询问块由生成的标签。通过TagGen算法，生成的标签并发送给。

伪造：最后，输出数据块由身份和公钥生成的标签。

如果满足以下条件，将获胜。

1）伪造的标签是在身份和公钥下对数据块的有效标签。

2）没有询问过的私钥。

3）没有同时询问的部分私钥并替换了的公钥。

4）没有在身份和公钥下对做过标签询问。

游戏2：由和来完成。

初始化：运行Setup算法，获得主密钥和公开参数。把主密钥和公开参数都发送给。

询问：可以在多项式时间内对做不同的询问。按照如下方式对的询问进行回应：

哈希询问。适应性的向做哈希询问。向回应哈希值。

秘密值询问。自适应地选择不同的并提交给来询问的秘密值。运行SecretValueGen算法生成秘密值并发送给。

公钥询问。自适应地选择不同的并提交给来询问的公钥。运行PublicKeyGen算法计算公钥并发送给。

标签询问。自适应地选择元组并提交给，以询问块 由生成的标签。通过TagGen算法，生成的标签并发送给。

伪造。最后，输出数据块由身份标签。

如果满足以下条件，将获胜。

1）伪造的标签是在身份下对数据块的有效标签。

2）没有询问过的秘密值。

3）没有在身份下对做过标签询问。

定义6：如果对于任意的多项式时间敌手（或），赢得游戏I和游戏II的概率是可忽略的，数据块的单个标签是存在性不可伪造的。

游戏III： 挑战者 和敌手 i在游戏中进行交互。在这里， 被认为是不可信的CSP。如果数据损坏， 会试图欺骗验证者数据是保持完整的。从定义6中，我们知道任何敌手在没有正确的私钥的情况下都不能伪造单个块的标签。因此，在这个游戏中，我们只关注 是否可以在不正确的数据上伪造完整性证据来通过验证。受[26]的启发，游戏3的过程定义如下。

初始化：为所有用户生成公共参数、主密钥和私钥。将所有私钥和主密钥私有，但将公共参数发送给。

标签询问：自适应地选择元组并发送给，询问由ID生成的的标签。通过TagGen算法生成的标签并发送给。

挑战。 生成一个随机的挑战消息，把发送给，并要求给出对应的数据拥有证据。

伪造。对于挑战消息，AIII生成证据并把它交给。如果可以通过完整性验证，并且中的块信息是错误的，获胜。

定义7。如果任意的多项式时间敌手只能以可忽略的概率赢得关于一个数据块的几个的游戏3，那么在没有正确数据的情况下，伪造完整性证据的概率是可忽略的。

**3 我们的方案**

**3.1 方案构造**

我们假设在一个群组中存在个用户，让表示用户的唯一标识。在不丧失一般性的情况下，我们将设置为群组管理员，他将设置系统并为其他用户生成部分密钥。我们假设文件被分成个块，表示为，每个块都是中的一个元素。方案详细实施如下：

Setup算法：令和是素数阶的乘法群。是一个双线性映射，是的生成元。选择选择两个安全哈希函数 和，一个伪随机排列(PRP) 和一个伪随机函数(PRF) 。随机选择作为主私钥，并计算。自己保留主私钥并公开系统参数。

PartitialKeyGen算法：当收到群组用户的身份，管理员计算，返回给作为它的部分私钥。注意在群组中是唯一的。

SecretValueGen算法：随机选择并设置作为秘密值并保密。

PublicKeyGen算法： 使用秘密值 计算公钥 。

TagGen算法：群组中的每个用户都可以访问文件块，并使用其私钥为它们生成标签。假设用户想要为块生成标签，计算标签的方程是，其中，表示唯一的文件标识。群组管理员维护一个公共日志文件，该文件存储的信息和标签生成者的身份。在添加或修改数据块后，用户应该添加或更新日志文件中相应的和的标签生成者的身份。用户上传这些块和标签到CSP。CSP可以通过下列等式检查标签的验证情况：

（1）

Challenge算法： 验证者随机选择挑战块的数量和两个随机值，其中分别是PRP和PRF的种子。验证者发送给CSP作为挑战信息。

ProofGen算法：当CSP收到挑战消息，CSP计算集合，其中，对于。根据每个被挑战块的标签生成者，CSP把集合分割成个子集，其中子集是由用户生成的标签的集合。令表示中元素的数量。我们可以得到 ，且对于，。对于每一个子集，CSP计算出，通过和 。CSP返回作为最终的证据，这里的。

Verify。当收到完整性证据，验证者计算，并按照CSP那样根据标签生成者对挑战集合进行划分。通过在日志文件中搜索，验证者获得了所有被挑战文件块的。最后，验证者检查等式（2）是否成立：

(2)

如果等式（2）成立，验证者输出1，否则输出0。如果CSP和验证者诚实的运行该协议，我们可以通过以下等式来检查协议的正确性：

**3.2 支持用户撤销**

如果任何用户 l离开该群组，则应声明的密钥和公钥无效。因此，应该从生成的标签中删除的密钥。否则，将无法验证这些标记的有效性，也无法检查文件的完整性。重新生成被撤销用户的标签的传统方法是从CSP下载块，重新生成标签，然后再次上传新的标签。这不可避免地增加了用户的通信成本和计算成本。因此，为了减轻这些潜在的开销，我们的方案设计了将标签更新工作外包给CSP，并降低了通信成本。我们使用ReTagGen算法来更新由被撤销的用户生成的标签。ReTagGen的构造如下所示，其中是被撤销的用户，是组中的另一个有效用户。值得注意的是，是组的创建者和管理者，我们认为不应该被撤销。

ReTagGen: 该算法包含、和CSP之间的三个交互。我们假设、和CSP之间不存在勾结，并且在交互过程中使用了安全通道。此外，在撤销程序中，要求、和CSP同时在线。

1. CSP P随机选择一个值 并将 通过安全信道发送给 。
2. 计算并发送 给 。
3. 计算并发送 给 CSP。
4. 当收到 ，CSP 计算 。使用等式（1）来检查由生成的所有标签-块对 。然后CSP将块的标签转换为：

，其中 是由生成的的有效标签。

**4 安全证明**

通过以下三个定理证明了该方案是安全的。

**4.1 安全证明**

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 的概率输出 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。 维护一个秘密值询问的列表 。首先检查 是否在 中存在。

1. 如果 不存在于 随机选择一个值 并抛掷一个硬币 。假设 的概率是 ，则 的概率是 。如果 计算 并添加元组 到中。然后 输出“失败” 并终止。 如果 计算 并添加 , ) 到中。然后 返回 给。
2. 如果 存在于 检查对应的值 。如果 输出“失败” 并终止。否则， 检索 并把它返回给 （是是一定在 中存在的）。

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， 直接返回 给 。
2. 如果 中不包含元组 , 随机选择一个值 并抛掷一个硬币 。假设 的概率是 ，则 的概率是 。如果 计算。 如果 设置 并计算 添加一个新的元组 到 中并返回 给。

-哈希询问： 自适应地对发起 -哈希询问。 为-哈希询问维护一个列表 。如果 中包含了，检索元组 并返回 给。 否则，选择一个随机值 并返回 给。然后并添加 到中。

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 是由身份对块 伪造的标签。

分析： 如果赢得游戏II， 可以根据等式（1）得到 。然后 在中检索元组(ID', ) 。如果 ， 终止并输出“失败”。否则， 在 中得到 ，从中得到 以及从中得到。根据上面提到的等式， 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、公钥询问、哈希询问都可以被无意外地完美执行。 的终止只发生在秘密值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 的概率输出 正确的值。相应的时间代价为 。□

定理3：如果离散对数假设成立， 敌手 只能以可忽略的概率赢得游戏 III 。

证明： 令挑战信息为 。如果 输出了完整性证据 并以不可忽略的概率赢得游戏 III ，我们可以得到以下的验证等式：

其中 表示挑战中的群组子集的数量。假设挑战信息 chal 的真正证据是 ，我们仍可以得到验证等式：, 因为 赢得了游戏 III，一定存在 而。根据上面的两个等式，有 。定义 对于每一个，我们知道 。基于这个结论，离散对数问题可以被如下解决。给定两个元素 其中 ，我们将计算出 。令 ，其中和 是从 中随机选择的。 我们可以得到下面的等式

然后我们可以得到。因为，只有有一个 不为0。 是一个 中的随机值， 所以 的概率只有。因此，我们可以以不可忽略的概率输出 正确的值。

**4.2 错误检测概率**

假设CSP上个块的被篡改。CSP随机选择 中的个块来生成证据。为了检测数据损坏，必须满足 。因此，错误检测的概率就等于 的概率。 令 表示错误检测的概率，我们有 。它表示: 。从上面的式子中我们可以看到，更多的挑战块会导致更高的错误检测概率。通过对文献 [5]的分析，对于有1%篡改块的文件，300个挑战块将使 而460 个挑战块会使 。因此，我们的方案可以实现较高的错误检测概率。