Fuzzy Identity-Based Data Integrity Auditing for Reliable Cloud Storage Systems

Yannan Li, Yong Yu*, Geyong Min, Willy Susilo, Jianbing Ni and Kim-Kwang Raymond Choo

Abstract—As a core security issue in reliable cloud storage, data integrity has received much attention. Data auditing protocols enable a verifier to efficiently check the integrity of the outsourced data without downloading the data. A key research challenge associated with existing designs of data auditing protocols is the complexity in key management. In this paper, we seek to address the complex key management challenge in cloud data integrity checking by introducing fuzzy identity-based auditing-the first in such an approach, to the best of our knowledge. More specifically, we present the primitive of fuzzy identity-based data auditing, where a user's identity can be viewed as a set of descriptive attributes. We formalize the system model and the security model for this new primitive. We then present a concrete construction of fuzzy identity-based auditing protocol by utilizing biometrics as the fuzzy identity. The new protocol offers the property of error-tolerance, namely, it binds private key to one identity which can be used to verify the correctness of a response generated with another identity, if and only if both identities are sufficiently close. We prove the security of our protocol based on the computational Diffie-Hellman assumption and the discrete logarithm assumption in the selective-ID security model. Finally, we develop a prototype implementation of the protocol which demonstrates the practicality of the proposal.

Index Terms—Cloud Storage, Data Integrity, Biometric-based Identity, Threshold Secret Sharing.

1 Introduction

Big data is eliciting attention from the academia as well as the industry. Over 2.5 quintillion bytes of data are created every day in the world, so much that 90% of the data has been created in the last two years alone. The explosive growth in the volume of data captured by the machines, sensors and IoT etc. has changed our lifestyle gradually. According to a prediction by IDC, data set will grow 10-fold by the year of 2020 and there will be 5,200 GB of data for every person on earth [1]. Traditional storage model cannot meet the people's requirement due to the increasing large amount of data, which leads to the emergence of cloud storage.

As a basic service of IaaS (Infrastructure as a service) model in cloud computing [1], cloud storage enables data owners to store their files to the cloud and deletes the local copy of the data, which dramatically reduces the burden of maintenance and management of the data. Cloud storage has a number of eye-catching features [2], say global data access, independent geographical locations, on demand self-service, resource elasticity, etc. Currently, both the individuals and big companies are enjoying the benefits due to cloud storage services.

Despite the benefits offered by cloud storage, there are many inherent security risks since when data owners outsource their data to the cloud, they generally lose physical possession of their data and even have no idea where their data are actually stored or who has the permission to getting access to their data. That is to say, it is the cloud servers who control the fate of the data after the data owners uploading their files to the cloud. The cloud servers assure they will try their best to protect the security of the cloud data, but the data loss accidents are inevitable. This is not surprising. Firstly, a short-time crash of the cloud server or the breakdown of the storage medium(e.g RAM) will cause the data easily corrupted. Moreover, users' data may be lost due to deliberate deletion by cloud servers in order to make the available storage space for other files to get more profit. A survey[2] reported that 43% of the respondents had lost their outsourced data and had to resort to recovering the data from backups. Data loss incident happens frequently in reality and has been regarded as one of the key security concerns in cloud storage[3]. For example, Amazon's cloud crash disaster permanently destroyed many customers' data. [4] It was reported that "the data loss was apparently small relative to the total data stored, but anyone who runs a web site can immediately understand how terrifying a prospect any data loss is". As a consequence, the data owners require a strong integrity guarantee of their outsourced data and

- *Yannan Li and Yong Yu are with School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China. E-mail: yyucd2012@gmail.com*
- *Yong Yu is with School of Computer Science, Shaanxi Normal University, Xi'an 710062, China.*
- *Geyong Min is with the College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, United Kingdom.*
- *Willy Susilo is with School of Computer Science and Software Engineering, University of Wollongong, Australia.*
- *Jianbing Ni is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1.*
- *Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249 USA.*

Manuscript received April 19, 2005; revised August 26, 2015.

1. http://www.computerworld.com/article/2493701/data-center/by-2020–there-will-be-5-200-gb-of-data-for-every-person-on-earth.html

2. http://datainsurance.org/wp-content/uploads/2015/02/A-CIS-White-Paper-2013-4-4.pdf

3. http://blogs.idc.com/ie/?p=210

4. http://www.businessinsider.com/amazon-lost-data-2011-4

they want to make sure that the cloud servers store their data correctly. Therefore, cloud data integrity is of particular importance in secure and reliable cloud storage.

To solve the aforementioned problems, Deswarte et al. [3] proposed the concept of remote data integrity checking (RDIC, is also known as data integrity auditing), which comprises three parties, namely: cloud server, data owner and third party auditor (TPA). A publicly verifiable RDIC protocol allows the TPA or anyone to check the integrity of the stored data on the cloud without the need to retrieve the entire dataset. In 2007, Ateniese et al. [4], [5] proposed a new notion of provable data possession (PDP) and gave two efficient constructions using homomorphic verifiable tag (HVT) based on RSA [6] algorithm. A HVT aggregates response of the challenged blocks into a single value, which significantly reduces the communication costs between the server and the TPA. A year later in 2008, Shacham and Waters [7] proposed the concept of proof of retrievability (POR), as well as providing a construction based on short signature algorithm [8] and proving its security in the random oracle model. Since then, a number of remote data integrity checking protocols have been proposed catering to different real world requirements, such as dynamic operation [9], [10], [11], [12], privacy-preserving [13], [14], [15] and publicly auditing [16], [17].

In the schemes discussed above, the data owner has a pair of public/private keys (pk and sk respectively), where sk is used to generate authenticators of blocks and pk is used to verify a proof generated by the cloud server. In other words, these systems are based on public key infrastructure (PKI), which involves a certificate authority (CA) that issues and verifies digital certificates, a registration authority that validates the identity of users requesting information from the CA, a central directory, and a certificate management system. PKI based protocols have two key limitations. Firstly, generation, management and revocation of certificates is complex and computationally expensive, and consequently scalability is challenging. Secondly, the level of trust required of CA may be unrealistic, particularly in light of recent high profile incidents. For example, in May 2015, a number of unauthorized digital certificates issued by an Egyptian CA was uncovered, which can potentially facilitate malicious attacks. [5]

A popular alternative to PKI (and simplifying the complex certificate management) is identity-based (ID-based) cryptosystem [18] proposed by Shamir in 1984 which binds the secret key to the user's identity, without the need for a digital certificate. Since then, a number of ID-based schemes (including remote data auditing protocols) have been proposed. Recently in 2015, for example, several ID-based remote data auditing protocols were proposed [20], [21] and in these protocols, identity information is an arbitrary text string. The latter comprises user's name, IP address and E-mail address, which allows a user to register for a private key corresponding to his identity from the private key generation center. Although ID-based cryptosystems resolve the need for complex certificate management, there are a number of limitations inherent in such systems. For example, the user's identity may not be truly unique if the identity information is not chosen properly (e.g. using a common name such as "John Smith"). Secondly, a user needs to "prove" to the private key generator centre that he is indeed entitled to a claimed identity, such as presenting a legal document supporting the claim. However, the supporting documents are subject to forgery. Generally, both ID-based and PKI-based schemes rely on what you know (e.g. identity information) and what you have (e.g. digital certificate and password).

Recently, fuzzy identity-based constructions [22], [23], [24] are gaining popularity and the ideas of fuzzy theory are involved in many applications [25], [26], [27]. Biometrics, as a typical kind of fuzzy identity, are based on what you are. In other words, biometric-based schemes authenticate or identify a user based on one's physiological or behavioral features (e.g. fingerprint, iris and facial features) [28], [29] and deployed in a number of real-world applications (e.g. biometric passport and mobile devices such as Apple iOS and Samsung phones). This is unsurprising, due to the benefits offered by biometric-based schemes. For example, biometric-based identities are easily portable and cannot be forgotten or misplaced. Although there is potential for fuzzy identity-based schemes to be used in cloud storage services, we observe that there is no published fuzzy identity-based data integrity auditing protocols for such services. This is, perhaps, due to the challenges in establishing an optimal error-tolerance for data integrity auditing protocols.

Our Contributions. In this paper, we seek to simplify key management issue of remote data integrity auditing protocols by introducing biometric-based identities in traditional RDIC protocols. We regard our contributions to be three-fold.

1) We propose the notion of fuzzy identity-based data integrity auditing designed to simplify key management.
2) We then formalize the system model and security model to ensure the security called *soundness* of this new primitive (i.e. if a cloud server can convince a verifier that the server is storing a file, if and only if it is actually storing that file).
3) We describe a concrete construction of fuzzy identity-based data integrity auditing protocol, by borrowing the idea of fuzzy identity-based encryption due to Shacham and Waters [7]. The latter employs "set overlap" distance metric to measure the distance between two identity sets. We then prove the security of the protocol in the selective-ID security model, which relies on the Computational Diffie-Hellman and Discrete Logarithm assumptions.

Organization. The rest of the paper is organized as follows. In the next section, we review the preliminaries required in the understanding of the rest of the paper. We describe the system model and security model in Section 3. In Section 4, we present our construction of the biometric-based data integrity auditing scheme, as well as analyzing its correctness and efficiency. The security proof of the proposed scheme is given in Section 5. Finally, we conclude the paper in Section 6.

5. http://www.pcworld.com/article/2901072/google-catches-bad-digital-certificates-from-egyptian-company.html

2 PRELIMINARIES

In this section, we review some preliminaries, including bilinear maps, complexity assumptions and threshold secret sharing scheme.

2.1 Bilinear Maps

Let G and G_T denote two groups of the same prime order p. A bilinear pairing $e : G \times G \rightarrow G_T$ is a map satisfying the following properties [30]:

Bilinear: $e(u^a, v^b) = e(u, v)^{ab}$, for all $u, v \in G$ and all $a, b \in Z_p$.

Non-degenerate: $e(g, g) \neq 1$, where g is the generator of G.

Computational: $e(u, v)$ can be computed efficiently for all $u, v \in G$.

2.2 Complexity Assumption

A. Computational Diffie-Hellmam(CDH) Assumption [8]

G denotes a cycle group of a prime order p. Given the tuple of (g, g^a, g^b) by the challenger, and the adversary $\mathcal{A}$ attempts to output $g^{ab} \in G$. We say the adversary $\mathcal{A}$ has an ϵ advantage if

$$Pr[\mathcal{A}(g, g^a, g^b) = g^{ab}] \geq \epsilon.$$

Definition 1. The (t, ϵ)-CDH assumption holds in G if no t-time adversary has advantage at least ϵ in solving the problem above.

B. Discrete Logarithm(DL) Assumption [31]

The discrete logarithm is defined as follows. The adversary $\mathcal{A}$ is given the tuple (g, g^a), and attempts to output $a \in Z_p$. The adversary $\mathcal{A}$ has the advantage ϵ in solving discrete logarithm problem if

$$Pr[\mathcal{A}(g, g^a) = a] \geq \epsilon$$

Definition 2. The (t, ϵ)-DL assumption holds in G if no t-time adversary has advantage at least ϵ in solving the problem above.

2.3 Threshold Secret Sharing Scheme

Secret sharing schemes were proposed by Shamir [32] which is used for storing important and sensitive information. It divides a secret into several parts and distributes a unique part, which we called a share of the secret, to each participant in a group. The secret can be recovered by a certain number of shares combining together. Threshold denotes the number which is sufficient for reconstructing the secret. Specifically, a (t, n) threshold secret sharing scheme includes n players $P_1, \cdots, P_n$ and a dealer. Each player P_i is distributed a secret share $s_i (1 \leq i \leq n)$ by the dealer respectively and each share should be kept confidential. Any group of t or more players can reconstruct the secret s by showing their shares. However, any group of players with less than t players learns nothing about the secret. Shamir's secret sharing scheme is based on polynomial interpolation, which takes t points to define a polynomial of degree $t - 1$. Suppose a (t, n) threshold secret sharing scheme is to share a secret s, the dealer first randomly chooses $t - 1$ integers $a_1, \cdots, a_{t-1}$ and a polynomial $f(x)$ of degree $t - 1$ such that $f(0) = s$. $f(x)$ can be defined as $f(x) = s + \sum_{i=0}^{t-1} a_i x^i$. Then, the dealer picks some $x_i \in Z_p$, and distributes the secret share $s_i = f(x_i)$ to each player P_i. If a set of players $S \subset P$, where $|S| \geq t$, wants to recover the secret s, they are able to reconstruct $f(x)$ by:

$$f(x) = \sum_{P_i \in S} \Delta_{x_i, s}(x) s_i,$$

where

$$\Delta_{x_i, s}(x) = \prod_{P_i \in S, j \neq i} \frac{x - x_j}{x_i - x_j}$$

denotes the Lagrange coefficient.

2.4 Fuzzy Identity-based Signature

The notion of Fuzzy identity-based signature is the signature analog of fuzzy identity based encryption, which was first proposed in [23]. It has the property of error tolerance, which allows a user issuing a signature with the identity ω and could be verified with another identity ω' if and only if they are within a certain distance. Thus, it can provide biometric authentication. This primitive consists of 4 algorithms as follows,

- Setup($1^k, d$): This is a probabilistic algorithm which takes a security parameter 1^k and an error tolerance d as input. It generates the master key mk as well as the public parameter pp.
- Extract(msk, ID): This is a probabilistic algorithm which takes a master key mk and an identity ID as input. It generates the secret key D_{ID} as well as the public parameter pp.
- Sign(pp, M, D_{ID}): This is a probabilistic algorithm which takes the public parameter pp, a secret key D_{ID} and a message M as input. It outputs the signature σ.
- Verify (pp, ID', M, σ): This is a deterministic algorithm which takes the public parameter pp, an identity ID' such that $|ID \cap ID'| \geq d$, the message M and the signature σ as input. It returns 1 or 0 to prove the signature is valid or not .

3 SYSTEM MODEL AND SECURITY MODEL

We now describe the system model and security model for biometric-based data integrity auditing protocols.

3.1 System Model

A fuzzy identity-based data integrity auditing scheme involves four entities, namely key generation centre (KGC), cloud user, cloud server and TPA (see Fig.1). The KGC is responsible for generating user's secret key based on the user's identities, and a TPA is the trusted entity designated to verify the cloud data's integrity on behalf of the cloud user upon request. The details of the biometric-based data integrity auditing protocol are as follows.

1) A cloud user presents the fuzzy identity to the KGC.
2) The KGC measures the properties of the fuzzy identity, and then generates a private key for the cloud user.

3) Upon receiving the secret key from the KGC, the cloud user is able to preprocess the file by generating metadata of the file, upload files together with the metadata to the cloud, and delete the local copy of the data.
4) Finally, both TPA and cloud server run a challenge-response protocol for data integrity auditing to determine if the stored data are intact.

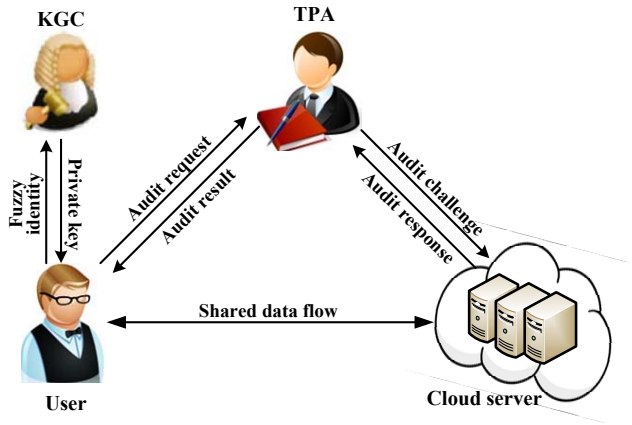

Fig. 1. The system model of fuzzy identity-based data integrity auditing protocol

3.2 System components

More formally, a biometric-based data integrity auditing protocol consists of the following six probabilistic algorithms.

- Setup($1^k, m, d$). This algorithm is a probabilistic algorithm run by the KGC. It takes a secure parameter k, a maximum number of dimensions in a vector that can describe a biometric-based identity m and an error tolerance d as input. It outputs the public parameters PP and the master secret key MK.
- Extract(PP, MK, ID). This algorithm is a probabilistic algorithm run by the KGC. It takes the public parameters PP, the master key MK and a user's identity ID as input. It outputs a private key K_w corresponding to the identity ID.
- MetadataGen(PP, K_w, F). This algorithm is a probabilistic algorithm run by the data owner. It takes the public parameters PP, the private key K_w and a file F as input. It outputs the file tag τ, and a set of block authenticators $\{\sigma_i\}$ of the file blocks $\{m_i\}$.
- Challenge(PP, τ, ID). This algorithm is a probabilistic algorithm run by the TPA. It takes the public parameters PP, the file tag τ and the data owner's biometric-based identity ID as input. It outputs a challenge $chal$.
- Response($PP, F, \{\sigma_i\}, chal$). This algorithm is a probabilistic algorithm run by the cloud server. It takes the public parameters PP, the file F, the block authenticators $\{\sigma_i\}$ and the challenge $chal$ as input. It outputs a respond $resp$ to prove the possession of the data.
- Verify($PP, ID', chal, resp$). This algorithm is a probabilistic algorithm run by the TPA. It takes the public parameters PP, the data owner's new biometric-based identity ID', the challenge $chal$ and the response $resp$ as input. It outputs an auditing result $result \in \{0, 1\}$ showing whether the cloud server keeps the file F unchanged.

3.3 Security Requirements and Security Model

A fuzzy identity-based data integrity auditing protocol needs to provide the following properties [7]

1) **Correctness.** Correctness states that a valid proof, generated by the *Response* algorithm, can pass the *Verify* algorithm with overwhelming probability.
2) **Soundness.** Soundness requires that, any cheating prover, who can generate a valid proof that can pass the *Verify* algorithm is actually storing the challenged file. In other words, there is no adversary who does not store the file can produce a valid proof of the challenge.

We provide the following selective-ID security model to make the notion of soundness more precisely. The essence of the security model is there exists an extractor $Extr(PP, ID, \tau, P')$, a polynomial time algorithm run by the challenger. It takes the public parameters PP, the identity ID, the file tag τ and the cheating prover P' as input, and can extract the file F. The game of soundness between an adversary and a challenger is described below.

Initial. The adversary declares a target identity, α, to be challenged upon.

Setup. The challenger runs the *Setup* algorithm and gets the public parameters PP and the master key MK. Then it forwards the PP to the adversary.

Queries. The adversary is allowed to make some queries including the extract queries and the metadatagen queries.

- Extract queries. The adversary can make queries of private keys for some identities γ_j, where $|\gamma_j \cap \alpha|$ should be less than d for all j.
- MetadataGen queries. The adversary can make queries on the file tag for some file F, the challenger runs the *Extract* algorithm to get the private keys and runs the *MetadataGen* algorithm to obtain the metadata of the file, and returns the metadata to the adversary.

ProofGen. For the file F on which a MetadataGen query has been made, the adversary can make an interaction with the challenger following the challenge-response protocol by specifying the identity ID and the file tag τ, where the verifier acts as the challenger. The adversary is provided with an output of *Verify* algorithm when the protocol execution completes.

Output. Finally, the adversary outputs a challenge tag τ together with the target identity α chosen at the $Initial$ stage, and the description of a prover P'.

The cheating prover P' now can interact with the verifier by executing the protocols with the input of the challenge tag τ, which is returned from file F', and the challenge identity α. If it can convincingly answer an ϵ fraction of the challenges, we say the cheating prover P' is ϵ-admissible.

Definition 1. An fuzzy identity-based data integrity auditing protocol is ϵ-sound if there exists an adversary playing the *Setup* algorithm and outputs an ϵ-admissible cheating prover P' for a file F, then there exists an extraction algorithm that can recover the file F from P' with a nonnegligible probability.

4 A Concrete Construction

In this section, we present a concrete construction of fuzzy identity-based data integrity auditing protocol using a biometric-based identity , e.g. fingerprint. We firstly present the basic idea of our construction, and then describe the concrete protocols in detail.

4.1 Basic idea

As shown in Fig.2, our biometric-based data integrity auditing protocol has three procedures, namely **Enroll**, **Store** and **Audit**.

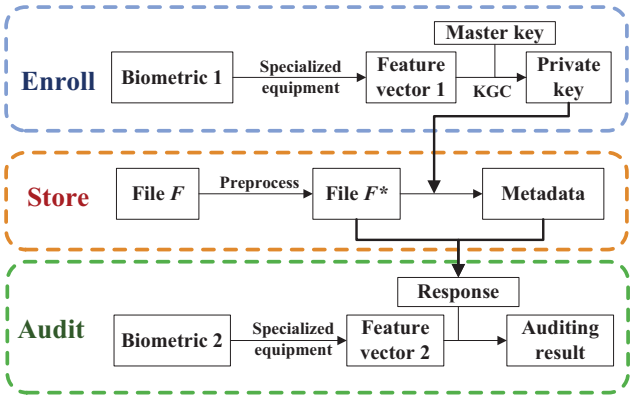

Fig. 2. Top view of biometric-based data integrity auditing protocols

- **Enroll.** It consists of the *Setup* and *Extract* algorithms. Firstly, the KGC sets up the public parameters PP and the master private key MK, and the cloud server establishes the data storage service. The cloud user, who is willing to enjoy the storage service, use the biometric collection equipments, e.g., fingerprint scanner, to obtain the biometric data α, which is a feature vector, and sends them to the KGC. The KGC extracts the private key K_w from the biometric data α for the user using its master key MK.
- **Store.** The *MetadataGen* algorithm is executed in this phase, in which the cloud user precesses the file F to generate some metadata, including the file tag τ and the block authenticators $\{\sigma_i^{(k)}\}$. Specifically, the cloud user firstly computes a fuzzy identity-based signature on the file name $name$ and some parameters $n, u_1, \cdots, u_s$ to generate the file tag τ. Then, the user splits the file F into n blocks $\{m_i\}_{1\le i\le n}$ and generates a block authenticator $\{\sigma_i^{(k)}\}_{1\le i\le n, k\in\omega}$ for each file block, which has the desirable homomorphism and allows the TPA to detect the corruption of the file F in cloud without heavy communication overhead.
- **Audit.** It is composed of the *Challenge*, *Response* and *Verify* algorithms. In *Challenge* algorithm, the TPA samples on the blocks of the file M to generate a challenge $chal$ and sends $chal$ to the cloud server. According to the challenge, the server generates proof $resp$ by aggregating the challenged blocks and the corresponding authenticators in the *Response* algorithm. Finally, the TPA verifies the response $resp$ to determine whether the file F is intact on the cloud.

4.2 Construction Description

The details of the protocol are as follows: Let G and G_T be two multiplicative cyclic groups of the same order p. g is a generator of G. $e: G\times G\to G_T$ denotes a bilinear map. We define Lagrange coefficient $\Delta_{i,S}$ for $i\in Z_p$ and a set, S, of element in Z_p:

$$\Delta_{i,S}=\prod_{j\in S, j\neq i}\frac{x-j}{i-j}.$$

Setup(m,d). First choose $g_1=g^y, g_2\in G$. Next, uniformly choose $t_1,\cdots,t_{m+1}\in G$ at random. Let M be the set $\{1,2,\cdots,m+1\}$ where m is the maximum number of attributes to describe a biometric-based identity. And we define a function T as:

$$T(x)=g_2{}^{x^m}\prod_{i=1}^{m+1}t_i{}^{\Delta_{i,M}(x)}.$$

Select a random integer $z'\in Z_p$, and compute $v'=g^{z'}$. The public parameters are $PP=\{g_1,g_2,t_1,\cdots,t_{m+1},v',A=e(g_1,g_2)\}$, and the master key is $MK=y$.

Extract(PP,MK,ω). To generate a private key for an identity ω, where $|\omega|=m$, the KGC first chooses a random $d-1$ degree polynomial q such that $q(0)=y$. Then the KGC calculates the private key corresponding to the identity ω as $K_w=(\{D_k\}_{k\in\omega},\{d_k\}_{k\in\omega})$ where:

$$D_k=g_2^{q(k)}T(k)^{r_k},$$
$$d_k=g^{-r_k}.$$

where each $r_k(k\in\omega)$ is a random number in Z_p.

MetadataGen(K_w,F). Given a file F, the data owner firstly applies the erasure code to F and obtains F'; then splits F' into n blocks, each s sectors long: $F^*=\{m_{ij}\}_{1\le i\le n,1\le j\le s}$. Choose a $name$ for a file from Z_p, s random elements $u_1,\cdots,u_s\in G$. Let $\tau_0=name\|n\|u_1\|\cdots\|u_s$. Then the file tag τ is τ_0 together with a fuzzy identity-based signature [23] on τ_0: $\tau=\tau_0\|Sign(\tau_0)$. For block $i(1\le i\le n)$, choose a random $s_k\in Z_p$ for each $k\in\omega$, and generate a block authenticator for the i-th block as follows:

$$\sigma_{1i}^{(k)}=\left\{D_k\cdot\left(H(name\|i)\cdot v'\cdot\prod_{j=1}^{s}u_j{}^{m_{ij}}\right)^{s_k}\right\}_{k\in\omega}$$
$$\sigma_{2i}^{(k)}=\{g^{-s_k}\}_{k\in\omega}$$

$$\sigma_{3i}^{(k)} = \{g^{-r_k}\}_{k\in\omega}$$

The data owner stores the file F', and the corresponding **metadata** of the file including file tag together with the block authenticators $(\tau, \{\sigma_{1i}^{(k)}, \sigma_{2i}^{(k)}, \sigma_{3i}^{(k)}\}_{1\le i\le n})$ on the cloud.

Challenge(PP, τ, w'). Upon receiving the auditing request from the cloud user with identity w', the TPA checks whether $|\omega \cap \omega'| \geq d$ holds. If so, the TPA picks an l-element subset I of the set $[1, n]$, and for each $i \in I$, chooses a random $v_i \in Z_p$. Let C be the set $\{(i, v_i)\}_{i\in I}$, the TPA forwards the challenge C to the cloud server.

Response$(F, \tau, C, \{\sigma_{1i}^{(k)}, \sigma_{2i}^{(k)}, \sigma_{3i}^{(k)}\}_{1\le i\le n})$. Upon receiving the challenge $C = \{(i, v_i)\}_{i\in I}$ from the TPA, the cloud server generates a response in the following way,

$$\mu_j = \sum_{(i,v_i)\in C} v_i m_{ij},$$

$$\sigma_1^{(k)} = \{\prod_{(i,v_i)\in C} \sigma_{1i}^{(k)^{v_i}}\}_{k\in\omega},$$

$$\sigma_2^{(k)} = \{\sigma_{2i}^{(k)}\}_{i\in C, k\in\omega},$$

$$\sigma_3^{(k)} = \{\sigma_{3i}^{(k)}\}_{i\in C, k\in\omega}.$$

and returns $resp = (\mu_1, \cdots, \mu_s, \sigma_1^{(k)}, \sigma_2^{(k)}, \sigma_3^{(k)})$ to the TPA.

Verify$(resp, chal, \omega')$. Upon receiving the proof from the server, the TPA chooses an arbitrary d-element subset S of $\omega \cap \omega'$, and verifies whether

$$\prod_{(i,v_i)\in C} A^{v_i} \stackrel{?}{=} \prod_{k\in S}\left(e(\sigma_1^{(k)}, g)\cdot \prod_{(i,v_i)\in C} e\Big(T(k), {\sigma_3^{(k)}}^{v_i}\Big)\cdot e\Big(\Big(H(name\|i)v'\Big)^{v_i}\prod_{j=1}^{s} {u_j}^{\mu_j}, \sigma_2^{(k)}\Big)\right)^{\Delta_{k,S}(0)}$$

holds. If the equation holds, return 1, otherwise return 0.

Correctness. The correctness of the verification is straightforward by the bilinear property of bilinear maps.

$$\begin{aligned}\sigma_1^{(k)} &= \prod_{(i,v_i)\in C} {\sigma_i}^{v_i}\\ &= \prod_{(i,v_i)\in C}\left(D_k\Big(H(name\|i)v'\prod_{j=1}^{s} {u_j}^{m_{ij}}\Big)^{s_k}\right)^{v_i}\end{aligned}$$

so we have

$$\begin{aligned}\prod_{k\in S} e(\sigma_1^{(k)}, g) &= \prod_{k\in S}\prod_{(i,v_i)\in C} e\Big({D_k}^{v_i}, g\Big)\cdot e\Big(\Big(H(name\|i)v'\Big)^{v_i}\prod_{j=1}^{s} {u_j}^{\mu_j}, g^{s_k}\Big)\Big)\\ &= \prod_{k\in S}\prod_{(i,v_i)\in C}\left(e\Big({g_2}^{q(k)}T(k)^{r_k}\Big)^{v_i}, g\right)\cdot e\Big(\Big(H(name\|i)v'\Big)^{v_i}\prod_{j=1}^{s} {u_j}^{\mu_j}, g^{s_k}\Big)\Big)\\ &= \prod_{k\in S}\prod_{(i,v_i)\in C} e\Big({g_2}^{q(k)}, g^{v_i}\Big)e\Big(T(k)^{r_k}, g^{v_i}\Big)\cdot e\Big(\Big(H(name\|i)v'\Big)^{v_i}\prod_{j=1}^{s} {u_j}^{\mu_j}, g^{s_k}\Big)\Big),\end{aligned}$$

which means that

$$\begin{aligned}&\prod_{k\in S}\left(e(\sigma_1^{(k)}, g)\cdot \prod_{(i,v_i)\in C} e\Big(T(k), {\sigma_3^{(k)}}^{v_i}\Big)\cdot e\Big(\Big(H(name\|i)v'\Big)^{v_i}\prod_{j=1}^{s} {u_j}^{\mu_j}, \sigma_2^{(k)}\Big)\right)^{\Delta_{k,S}(0)}\\ =\ &\prod_{k\in S}\prod_{(i,v_i)\in C}\left(e\big({g_2}^{q(k)}, g\big)^{v_i}\right)^{\Delta_{k,S}(0)}\\ =\ &\prod_{(i,v_i)\in C} e\big(g_2, g^y\big)^{v_i} = \prod_{(i,v_i)\in C} A^{v_i}\end{aligned}$$

so the equation in *Verify* algorithm holds.

5 Security Proofs

In this section, we prove that the biometric-based data integrity auditing protocol achieves the property of soundness. In other words, even if a cloud server exhibits arbitrary dishonest behavior, the server must store the file with a high probability provided that it generates a valid response for a challenge of the file. The soundness of our construction is concluded in the following theorem.

Theorem. If the fuzzy identity-based digital signature algorithm employed for file tags is existentially unforgeable and the CDH assumption is hard in bilinear groups, in the random oracle model, except with negligible probability no adversary against the soundness of our fuzzy identity-based remote data integrity checking protocol can cause the verifier to accept a response of a challenge instance, except by generating values $\{\mu_j\}$ and $\{\sigma_1^{(k)}, \sigma_2^{(k)}, \sigma_3^{(k)}\}$ correctly, i.e., as they computed in the *Response* algorithm in our protocol.

We prove this theorem in a series of games.

Game 0. The first game, Game 0, is simply the challenge game with a change for a publicly verifiable response defined in Section 3.

Game 1. Game 1 is the same as Game 0, with one difference. The challenger keeps a list of file tags issued as the metadata of the outsourced data. If the adversary is able to generate such a file tag t that (1) is valid under the *Sign* algorithm but (2) is not a tag generated by the challenger, the challenger aborts.

It is clear that the difference in the adversary's success probability between Game 0 and Game 1 is the probability that the adversary can break the fuzzy identity-based digital signature scheme used for generating tags of a file.

Game 2. Game 2 is the same as Game 1, with one difference. The challenger keeps a list of responses to extraction queries from the adversary. Now the challenger observes each instance of the protocol, including key extraction, tag generation, challenge-response and verification. If in any of these instances the adversary is successful, that is, the *Verify* algorithm outputs 1, but the private key for the identity w in this instance is not generated by the extraction algorithm, the challenger declares failure and aborts.

Clearly, the difference in the adversary's success probability between Game 2 and Game 1 is the probability that the adversary can forge a valid private key for an identity w. With this in mind, we now demonstrate that if there is a nonnegligible difference in the adversary's success probability between Game 2 and Game 1, we can construct a simulator that can solve the Computational Diffie-Hellman problem.

The simulator is given g, g^a, g^b as input, and is supposed to output g^{ab}. The simulation runs as follows:

The adversary selects a random identity α to be challenged upon.

The simulator sets the public parameters as $g_1 = g^a, g_2 = g^b$. It then chooses a random polynomial of m degree $f(x)$ and another m degree polynomial such that

$$u(x)\begin{cases} = -x^m, & x \in \alpha \\ \neq -x^m, & others \end{cases}$$

the simulator sets $t_i = {g_2}^{u(i)} g^{f(i)}$, for i from 1 to m. Then we can compute T_i in the same way in construction: $T(i) = {g_2}^{i^m} \prod_{i=1}^{m} {t_i}^{\Delta_{j,M(i)}} = {g_2}^{i^m} \prod_{i=1}^{m} {g_2}^{u(i)} {g^{f(i)}}^{\Delta_{j,M(i)}} = {g_2}^{i^m+u(i)} g^{f(i)}$. So we equivalently set $T(i)$ as follows:

$$T(i) = \begin{cases} g^{f(i)}, & x \in \alpha \\ {g_2}^{i^n+u(i)} g^{f(i)}, & others \end{cases}$$

Then the public parameters $PP = (g, g_1, g_2, t_1, \cdots, t_{m+1}, A = e(g_1, g_2))$, and the master secret key is a, which is unknown to the simulator.

To answer the private key queries on identity γ, where $|\gamma \cap \alpha|$ is less than d, the simulator acts as follows. We first define three sets, as shown in Fig. 1, Γ, Γ', S, where $\Gamma = \gamma \cap \alpha$, Γ' is the set satisfying $\Gamma \subseteq \Gamma' \subseteq \gamma$ and $|\Gamma'| = d - 1$, $S = \Gamma' \cup 0$ Then we set the private keys $\{D_i, d_i\}_{i \in \gamma}$ as follows:

1) For $i \in \Gamma'$, the private keys are set as:

$$D_i = {g_2}^{\lambda_i} T(i)^{r_i}, d_i = g^{-r_i}$$

where λ_i, r_i are randomly chosen in Z_p.

2) For $i \in \gamma - \Gamma'$, the private keys are computed as

$$D_i = \left(\prod_{j \in \Gamma'} {g_2}^{\lambda_j \Delta_{j,S(i)}}\right) \left({g_1}^{\frac{-f(m)}{i^m+u(i)}} \left({g_2}^{i^m+u(i)} g^{f(i)}\right)^{r_i'}\right)$$

$$d_i = \left({g_1}^{\frac{1}{i^m+u(i)}} g^{-r_i'}\right)^{\Delta_{i,S(0)}}$$

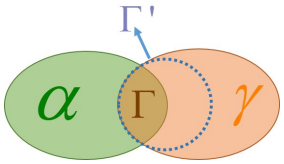

Fig. 3. Query sets

When $i \in \gamma - \Gamma'$, which indicates $i \notin \alpha$, $u(i) \neq -i^m$, so $i^m + u(x)$ will be non-zero. We claim the assignment is identical to the original scheme from the adversaries view. To observe this, we set $r_i = (r_i' - \frac{a}{i^m+u(i)})\Delta_{0,S}(i)$, then we have

$$\begin{aligned} D_i &= \left(\prod_{j \in \Gamma'} {g_2}^{\lambda_j \Delta_{j,S(i)}}\right) \left({g_1}^{\frac{-f(m)}{i^m+u(i)}} \left({g_2}^{i^m+u(i)} g^{f(i)}\right)^{r_i'}\right)^{\Delta_{0,S}(i)} \\ &= \left(\prod_{j \in \Gamma'} {g_2}^{\lambda_j \Delta_{j,S(i)}}\right) \left(g^{\frac{-af(m)}{i^m+u(i)}} \left({g_2}^{i^m+u(i)} g^{f(i)}\right)^{r_i'}\right)^{\Delta_{0,S}(i)} \\ &= \left(\prod_{j \in \Gamma'} {g_2}^{\lambda_j \Delta_{j,S(i)}}\right). \\ &\left({g_2}^{a} \left({g_2}^{i^m+u(i)} g^{f(i)}\right)^{\frac{-af(m)}{i^m+u(i)}} \left({g_2}^{i^m+u(i)} g^{f(i)}\right)^{r_i'}\right)^{\Delta_{0.S}(i)} \\ &= \left(\prod_{j \in \Gamma'} {g_2}^{\lambda_j \Delta_{j,S(i)}}\right) \left({g_2}^{a} \left({g_2}^{i^m+u(i)} g^{f(i)}\right)^{r_i' - \frac{-af(n)}{i^m+u(i)}}\right)^{\Delta_{0.S}(i)} \\ &= \left(\prod_{j \in \Gamma'} {g_2}^{\lambda_j \Delta_{j,S(i)}}\right) {g_2}^{a\Delta_{0,S}(i)} (T(i))^{r_i} \\ &= {g_2}^{q(i)} T(i)^{r_i} \end{aligned}$$

$$\begin{aligned} d_i &= \left({g_1}^{\frac{1}{i^m+u(i)}} g^{-r_i'}\right)^{\Delta_{i,S}(0)} = \left(g^{-(r_i' - \frac{a}{i^m+u(i)})}\right)^{\Delta_{i,S}(0)} \\ &= g^{-r_i} \end{aligned}$$

Eventually, the adversary outputs a valid forgery of the private keys $k_\alpha = (\{{D_i}^*\}, \{{d_i}^*\})_{i \in \alpha}$ for the identity α. The simulator then can solve the CDH problem using the forgery from the adversary. First, the simulator selects a random set $\alpha^* \subseteq \alpha$, where $|\alpha^*| = d$, and computes as follows:

$$D^* = \prod_{i \in \alpha^*} \{{D_i}^*\}^{\Delta_{i,\alpha^*}(i)}$$

$$d^* = \prod_{i \in \alpha^*} \{{d_i}^*\}^{\Delta_{i,\alpha^*}(i) f(i)}$$

Finally. the simulator can give the solution to the instance of the CDH problem as

$$g^{ab} = D^* d^*$$

Therefore if there is a nonnegligible difference in the adversary's success probability between Game 2 and Game 1, we can construct a simulator that can solve the Computational Diffie-Hellman problem, as required.

Game 3. Game 3 is the same as Game 2, with one difference. The challenger keeps a list of responses to metadata queries from the adversary. Now the challenger observes each instance of the protocol, including key extraction,

metadata generation, challenge-response and verification. If in any of these instances the adversary is successful, that is, the *Verify* algorithm outputs 1, but the adversary's aggregate authenticator is not equal to

$$\sigma_1^{(k)} = \{\prod_{(i,v_i)\in C} {\sigma_{1i}^{(k)}}^{v_i}\}_{k\in\omega},$$

$$\sigma_2^{(k)} = \{\prod_{(i,v_i)\in C} g^{-s_i}\}_{k\in\omega},$$

$$\sigma_3^{(k)} = \{\prod_{(i,v_i)\in C} g^{-r_i}\}_{k\in\omega},$$

the challenger declares failure and aborts.

We analyze the difference in success probabilities between Game 3 and Game 2. Suppose the file which leads to abort has n blocks, with name $name$, has generated exponents $\{\mu_j\}$ and contains sectors $\{m_{ij}\}$, and the block authenticators issued by **Metadata** generation are $\{\sigma_{1i}^{(k)}, \sigma_{2i}^{(k)}, \sigma_{3i}^{(k)}\}$. Suppose $C = \{(i, v_i)\}$ is the query that causes the challenger's failure, and the adversary's response to that query was $\mu_1', \mu_2', \cdots, \mu_s'$ together with ${\sigma_1^{(k)}}', {\sigma_2^{(k)}}', {\sigma_3^{(k)}}'$. The difference in success probabilities between Game 3 and Game 2 is forging a valid aggregate authenticator for challenge C. With this in mind, we show that if there is a nonnegligible difference in the adversary's success probability between Game 3 and Game 2, there is another algorithm that can solve the Computational Diffie-Hellman problem.

The simulator $\mathcal{C}$ is given as inputs an instance of the Computational Diffie-Hellman problem (g, g^a, g^b), and its goal is to output the value of g^{ab}. Assume the adversary makes at most $l \ll p$ metadata queries. The simulation between the simulator $\mathcal{C}$ and the adversary $\mathcal{A}$ is as follows.

Setup: $\mathcal{C}$ picks a target identity α^* and sets $g_1 = g^a$ and $g_2 = g^b$. Then, $\mathcal{C}$ sets up the simulation as follows.

1) Select a random $k \in \{0, 1, \cdots, n\}$.
2) Choose random values $x', x_1, \cdots, x_s$ from $\{0, 1, \cdots, 2l-1\}$.
3) Choose random $z', z_1, \cdots, z_s$ from z_p.
4) Find a random m degree polynomial $f(x)$.
5) Define an m degree polynomial $u(x)$ such that $u(x) = -x^m$ if and only if $x \in \alpha^*$.
6) For $1 \le k \le n+1$, set $t_k = g_2^{u(k)} g^{f(k)}$

Under this assignment, we have

$$T(k) = g_2^{k^n} \prod_{j=1}^{n+1} (g_2^{u(j)} g^{f(j)})^{\delta_{j,N}(k)} = g_2^{k^n + u(k)} g^{f(k)}.$$

$\mathcal{C}$ publishes the public parameters of the system as

$$\Big(g, g_1, g_2, t_1, \cdots, t_{n+1}, v' = g_2^{x'-2kl} g^{z'},$$

$$\{u_j = g_2^{x_j} g^{z_j}\}_{1\le j\le s}, A = e(g_1, g_2)\Big)$$

The simulator keeps a list of hash table. For each $i (1 \le i \le n)$, the simulator picks a random $\rho_i \in Z_p$ and sets the random oracle at i as

$$H(name\|i) = g^{\rho_i}$$

.

To respond a query on identity α^* of a file $M = \{m_{ij}\}_{1\le i\le n, 1\le j\le s}$, for the L-th block $\{m_{Lj}\}_{1\le j\le s}$, we define

$$F_L = -2lk + x' + \sum_{j=1}^{s} x_j m_{Lj}$$

and

$$J_L = z' + \sum_{j=1}^{s} z_j m_{Lj}.$$

If $F_L = 0 \pmod p$, $\mathcal{C}$ declares failure and aborts. Otherwise, $\mathcal{C}$ chooses a set $\Theta \subset \alpha^*$, where $|\Theta| = d-1$ and for $k \in \Theta$, defines $g^{q'(k)} = g^{\lambda_k'}$ in which λ_k' are random elements in Z_p. For $k \in \alpha^* \setminus \Theta$, $\mathcal{C}$ computes $g^{q'(k)} = (\prod_{j=1}^{d-1} g^{\lambda_j' \Delta_{j,\alpha^*}(k)}) g^{a\Delta_{0,\alpha^*}(k)}$. For $k \in \alpha^*$, $\mathcal{C}$ picks random $r_k^L, s_k^L \in Z_p$ and computes

$$\sigma_{1L}^{(k)} = \{(g^{q'(k)})^{-\frac{J_L+\rho_L}{F_L}} g^{\rho_L s_k^L + f(k) r_k^L} (g^{J_L} g_2^{F_L})^{s_k^L}\}_{k\in\alpha^*},$$

$$\sigma_{2L}^{(k)} = \{g^{-r_k^L}\}_{k\in\alpha^*},$$

$$\sigma_{3L}^{(k)} = \{(g^{q'(k)})^{1/F_L} g^{-s_k^L}\}_{k\in\alpha^*}.$$

It is straightforward to show that $\{\sigma_{1L}^{(k)}, \sigma_{2L}^{(k)}, \sigma_{3L}^{(k)}\}_{k\in\alpha^*}$ is a valid authenticator for the block L where the random value $\hat{s}_k = s_k - q'(k)/F_L$. In this way, the simulator $\mathcal{C}$ is able to simulate the authenticators for all the blocks.

Eventually, the adversary $\mathcal{A}$ outputs a valid authenticator forgery $S^* = \{\sigma_{1i}^{(k)^*}, \sigma_{2i}^{(k)^*}, \sigma_{3i}^{(k)^*}\}_{k\in\alpha^*, 1\le i\le n}$ on file F^* for the identity α^*.

Assume ${F_L}^* = -2lk + x' + \sum_{j=1}^{s} x_j m_{Lj}^*$ and ${J_L}^* = z' + \sum_{j=1}^{s} z_j m_{Lj}^*$. If $|\gamma \cap \alpha^*| \ge d$ or if $F^* \ne 0 \pmod p$, $\mathcal{C}$ aborts. Otherwise, the L-th block authenticator forgery is $\{\sigma_{1L}^{(k)^*}, \sigma_{2L}^{(k)^*}, \sigma_{3L}^{(k)^*}\}_{k\in\alpha^*}$

Now the challenger $\mathcal{C}$ selects a random set $\Theta^* \subset \alpha^*$ and $|\Theta^*| = d$, and computes as follows:

$$\sigma_1^{(k)^*} = \prod_{k\in\Theta^*} \left({S_{1L}}^*\right)^{\Delta_{k,\alpha}(0)},$$

$$\sigma_2^{(k)^*} = \prod_{k\in\Theta^*} \left({S_{2L}}^*\right)^{\Delta_{k,\alpha}(0) f(k)},$$

$$\sigma_3^{(k)^*} = \prod_{k\in\Theta^*} \left({S_{3L}}^*\right)^{\Delta_{i,\alpha}(0)}.$$

Then $\mathcal{C}$ can solve the CDH problem problem by computing $\sigma_1^{(k)^*} \cdot \sigma_2^{(k)^*} \cdot \left(\sigma_3^{(k)^*}\right)^{{\rho_L}^* {J}^*} = g^{ab}$

Game 4. Game 4 is the same as Game 3, with one difference. As before, the challenger keeps a list of responses to metadata queries from the adversary. Now the challenger observes each instance of the protocol, including key extraction, metadata generation, challenge-response and verification. If in any of these instances the adversary is successful, that is, the *Verify* algorithm outputs 1, but at least one of the aggregate messages μ_j is not equal to $\sum_{(i,v_i)\in C} v_i m_{ij}$, where C is the challenge issued by the verifier, the challenger declares failure and aborts.

Again, we analyze the difference in success probabilities between Game 3 and Game 2. Suppose the file

which leads to abort has n blocks, with name $name$, has generated exponents $\{\mu_j\}$ and contains sectors $\{m_{ij}\}$, and the block authenticators issued by **Metadata** generation are $\{\sigma_{1i}^{(k)}, \sigma_{2i}^{(k)}, \sigma_{3i}^{(k)}\}$. Suppose $C = \{(i, v_i)\}$ is the query that causes the challenger's failure, and the adversary's response to that query was $\mu_1', \cdots, \mu_s'$ together with ${\sigma_1^{(k)}}', {\sigma_2^{(k)}}', {\sigma_3^{(k)}}'$. Let the expected response be $\mu_1, \cdots, \mu_s$ and $\{\sigma_1^{(k)}, \sigma_2^{(k)}, \sigma_3^{(k)}\}$, in which

$$\mu_j = \sum_{(i,v_i)\in C} v_i m_{ij},$$

$$\sigma_1^{(k)} = \left\{ \prod_{(i,v_i)\in C} {\sigma_{1i}^{(k)}}^{v_i} \right\}_{k\in\omega},$$

$$\sigma_2^{(k)} = \{\sigma_{2i}^{(k)}\}_{i\in C, k\in\omega},$$

$$\sigma_3^{(k)} = \{\sigma_{3i}^{(k)}\}_{i\in C, k\in\omega}.$$

Game 3 already guarantees that the authenticators of all the blocks are equal, and it is only the values μ_j' and μ_j that can be different. Define $\Delta\mu_j = \mu_j' - \mu_j$ for $1 \le j \le s$, there is at least one $\Delta\mu_j$ whose value is not zero since at least one of the aggregate messages μ_j is not equal to the expected value.

We show that if there is a nonnegligible difference in the adversary's success probability between Game 4 and Game 3, there is another algorithm that can solve the the discrete logarithm problem. The simulator $\mathcal{S}$ is given $g, h \in G$, and its goal is to output x such that $h = g^x$. $\mathcal{S}$ behaves like the Game 3 challenger, with the following differences.

1) When being asked to generate the metadata of the file with n blocks $\{m_{ij}\}$, $\mathcal{S}$ picks two random values β_j, γ_j and sets $u_j = g^{\beta_j} h^{\gamma_j}$.
2) $\mathcal{S}$ continues interacting with the adversary until the specified condition of Game 3 occurs.

Since the authenticators of all the blocks are equal and both the responses are valid, we have

$$\begin{aligned}\prod_S e(\sigma_1^{(k)}, g) &= \prod_S \prod_{(i,v_i)\in C} e\Big(g_2{}^{q(i)}, g^{v_i}\Big) e\Big(T(i)^{r_i}, g^{v_i}\Big) \cdot \\ &\quad e\Big(\Big(H(name\|i)v'\Big)^{v_i}, g^{s_i}\Big) e\Big(\prod_{j=1}^{s} {u_j}^{\mu_j}, g^{s_k}\Big)\end{aligned}$$

and

$$\begin{aligned}\prod_S e(\sigma_1^{(k)}, g) &= \prod_S \prod_{(i,v_i)\in C} e\Big(g_2{}^{q(i)}, g^{v_i}\Big) e\Big(T(i)^{r_i}, g^{v_i}\Big) \cdot \\ &\quad e\Big(\Big(H(name\|i)v'\Big)^{v_i}, g^{s_i}\Big) e\Big(\prod_{j=1}^{s} {u_j}^{\mu_j'}, g^{s_k}\Big)\end{aligned}$$

Therefore, we can get

$$\prod_{j=1}^{s} {u_j}^{\mu_j} = \prod_{j=1}^{s} {u_j}^{\mu_j'}.$$

Since $u_j = g^{\beta_j} h^{\gamma_j}$, we have

$$g^{\sum_{j=1}^{s} \beta_j \Delta\mu_j} h^{\sum_{j=1}^{s} \gamma_j \Delta\mu_j} = 1.$$

Because there is at least one of the $\{\Delta\mu_j\}$ is nonzero, finally, we get the solution to the given instance of discrete logarithm as follows,

$$h = g^{-\frac{\sum_{j=1}^{s} \beta_j \Delta\mu_j}{\sum_{j=1}^{s} \gamma_j \Delta\mu_j}}.$$

Wrapping up. In Game 4, the adversary is constrained from generating a response to a challenge other than those that would have been generated by the *Response* algorithm specified in Section 4. Yet we have shown that, assuming the employed fuzzy identity-based signature algorithm is unforgeable, say the scheme in [23], and computation Diffie-Hellman problem and discrete logarithm problem are intractable in bilinear groups, there is only a negligible difference in the success probability of the adversary in Game 4 compared with Game 1, where the adversary is not constrained in this manner. This completes the proof of this **Theorem**.

6 IMPLEMENTATION

All the algorithm in this implementation was conducted on a 64-bit PC with Intel Core (TM) i5 - 4300 CPU @ 2.13GHz and 8.0 GB RAM. All the projects were written in the C++ language using the Visual Studio 2010 compiler with the assistance of the MIRACL library [33], which is widely regarded as the gold standard open source for elliptic curve cryptography. To make the implementation of our protocol more general, we implement it with an asymmetric bilinear map, $e : G_1 \times G_2 \to G_T$, which is more efficient than the symmetric ones in practice. For the asymmetric map in MIRACL library, the ate pairing is always used. We fix the security level of our protocol as $\lambda = 80$, and the Cock-Pinch curve [34] as $y^2 = x^3 - 3x + B \pmod p$ is chosen, where p is a 2λ-bit prime, i.e. 160 bits. To capture the biometric-based identity(e.g. fingerprints) as an input, we use an optical fingerprint sensor to collect the information. We set the acquisition rate to be 24 frames per second which is to ensure the real-time performance of human-computer interaction. We will show the implementation results from the following four aspects.

In the first part, we show the time cost of the **Setup** and **Extract** algorithms. We can learn from Fig. 4 that the time consumption of **Setup** grows linearly with m, the maximum number of attributes to describe an identity in the protocol. This is reasonable because the function T needs to perform m multiplications when generating the public parameters. Fig. 5 shows that the time cost in **Extract** algorithm grows linearly with the increment of the number of attributes. This is consistent with the empirical analysis since a user's private key is calculated for each attribute of an identity.

In the second part, we would like to see the time cost to generate metadata for a file of fixed size with the increasing of block size. We fix the file as 1 MB, and select m, the maximum number of attributes, as 10 and use 3 attributes to describe the identity of a user. We increase the block size from 1 KB to 100 KB with the increment of 10 KB for each step. The result is reported in Fig. 6. As one shall see, it takes about 157 seconds in the on-line phase to generate metadata for the file, however, the time cost of the off-line phase grows almost linearly with the increasing of the block size.

This is in accordance with the analysis of the **MetadataGen** algorithm because the sector has a constant size (160 bits), and thus, for a file of fixed size, the total number of sectors is a constant. This leads to the computation of $\prod_{j=1}^{s} {u_j}^{m_{ij}}$ is constant in the on-line line phase regardless of the block size.

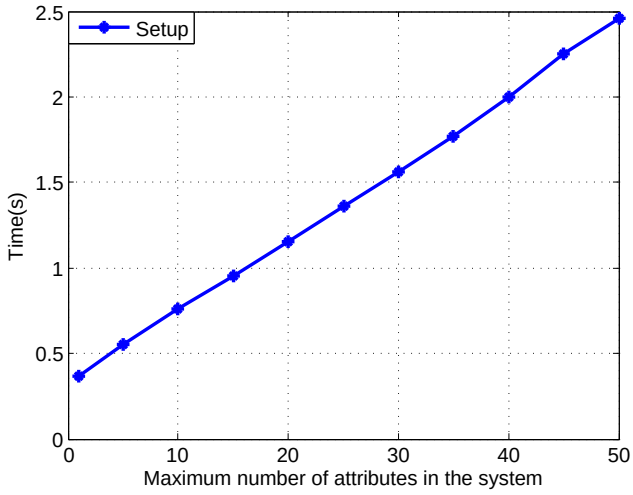

Fig. 4. Time consumption for **Setup** algorithm

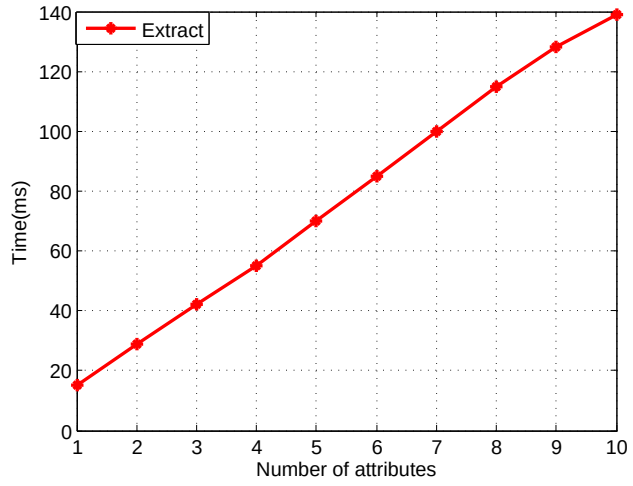

Fig. 5. Time consumption for **Extract** algorithm

In the third part, we try to determine the optimal block size by analyzing the tradeoff of the time cost between **MetadataGen** and **Audit** algorithm. With the increase of block size, the number of sectors increases as well (sector size is a constant). This will result in the increase of the computational cost of $\prod_{j=1}^{s} {u_j}^{m_{ij}}$ in the **MetadataGen** algorithm. However, the number of the blocks decreases when the block size increases, which leads to less time consumption since fewer $\sigma_1^{(k)}, \sigma_2^{(k)}$ and $\sigma_3^{(k)}$ need to be calculated in **Response** and verified in **Verify** (Note that, we challenge all the blocks in the example). The tradeoff in Fig. 7 indicates that the protocol has the best performance when the block size is between 8 KB and 10 KB. We choose 10 KB as the

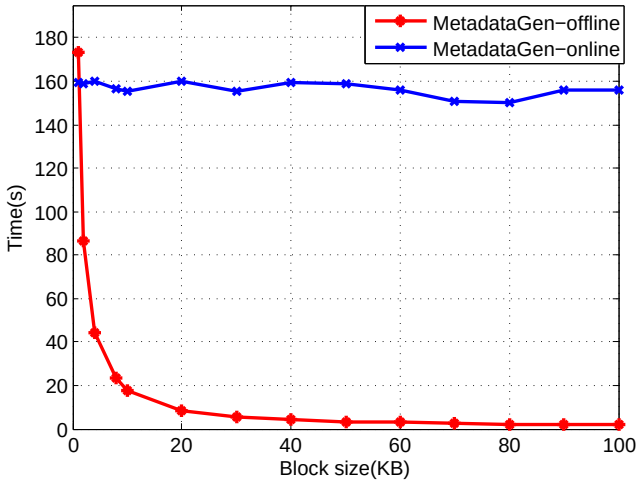

Fig. 6. Time consumption for **MetadataGen** algorithm of 1MB file

optimal block size to minimum the user's computational cost. When the block size is of 10 KB, the time cost of **Setup**, **Extract** and **MetadataGen** is 0.703 second, 0.041 second and 173 seconds respectively.

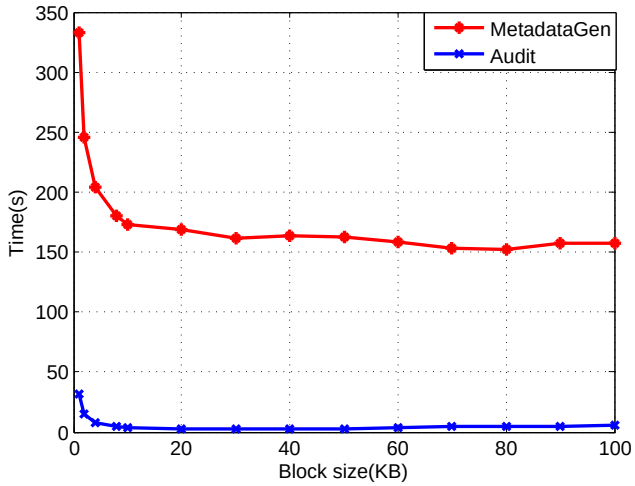

Fig. 7. Tradeoff between **MetadataGen** and **Audit** for 1MB file

In the fourth part, we choose a file of $10,000$ blocks with each block size 10 KB to test the performance of the server and the TPA by increasing the number of challenged blocks from 100 to 800, with the increment of 100 for each step. We can see from Fig. 8 that both the server and the TPA's cost grows with the increasing of the number of challenged blocks linearly. According to the observation due to Ateniese et al. [4], if 1% of the entire blocks have been corrupted, 300 and 460 blocks out of $10,000$ blocks can be challenged by the TPA to detect the misbehavior of the server with the probability of 95% and 99%, respectively. We can see that it costs the TPA 997.3 milliseconds and the server 9.071 seconds when challenging 300 blocks, while when challenging 460 blocks, it costs the TPA 1.324 seconds and the server 12.938 seconds. The implementation results show

that our protocol is very efficient for both the server and the TPA.

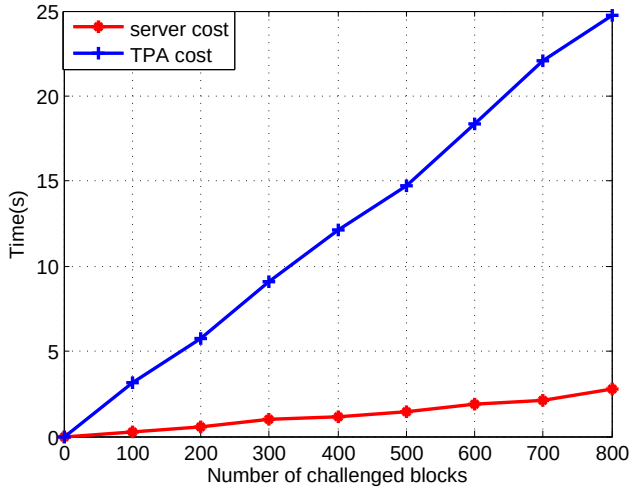

Fig. 8. Challenged blocks for a file of 10000 blocks

7 Conclusion

Cloud storage services have become an increasingly important part of the information technology industry in recent years. With more users getting involved in cloud storage, ensuring the integrity of data outsourced to the cloud is of paramount importance. In this paper, we presented the first fuzzy identity-based data integrity auditing protocol. The proposed protocol revolutionizes key management in traditional remote data integrity checking protocols. We also presented the the system and security models for this primitive, and a concrete fuzzy identity-based data integrity auditing protocol using the biometric-based identity as an input. We then demonstrated the security of the protocol in the selective-ID model. The prototype implementation of the protocol demonstrates the practicality of the proposal.

Acknowledgments

The authors would like to thank the anonymous referees sincerely for their very valuable comments. This work is supported by the National Natural Science Foundation of China (61501333,61300213,61272436,61472083), the Fundamental Research Funds for the Central Universities under Grant ZYGX2015J059.

References

[1] M. Hogan, F. Liu, A. Sokol and J. Tong, "NIST Cloud Computing Standards Roadmap," NIST Cloud Computing Standards Roadmap Working Group, SP 500-291-v1.0, NIST, Jul, 2011.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing, University of California, Berkeley, Tech. Rep.

[3] Y. Deswarte, J. J. Quisquater and A. Saidane. "Remote integrity checking". *Integrity and Internal Control in Information Systems VI*. Springer US, pp.1-11, 2004.

[4] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson and D. X. Song, "Provable data possession at untrusted stores," in *Proc. of ACM Conference on Computer and Communications Security*, pp.598-609, 2007.

[5] G. Ateniese, S. Kamara and J. Katz. "Proofs of storage from homomorphic identification protocols". *Proc. of ASIACRYPT*, pp.319-333, 2009.

[6] R. L. Rivest, A. Shamir and L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems". *Communications of the ACM*, 21(2), pp.120-126, 1978.

[7] H. Shacham and B. Waters, "Compact proofs of retrievability," *Proc. of Cryptology-ASIACRYPT*, 5350, pp.90-107, 2008.

[8] D. Boneh , B. Lynn, and H. Shacham, "Short signatures from the weil pairing", In *Proc. of Asiacrypt 2001*, pp.514-532, 2001.

[9] C. C. Erway, A. Kupcu and C. Papamanthou. "Dynamic provable data possession". *ACM Transactions on Information and System Security (TISSEC)*, 17(4), 15, 2015.

[10] Q. Wang, C. Wang, J. Li, K. Ren and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing". *Proc. of ESORICS2009*, LNCS 5789, pp.355-370, 2009.

[11] Y. Yu, J.B. Ni, M. H. Au, H.Y. Liu, H. Wang and C.X. Xu. "Improved security of a dynamic remote data possession checking protocol for cloud storage". *Expert Syst. Appl*. 41(17), pp.7789-7796, 2014.

[12] Y. Zhu, G. J. Ahn, H. Hu, S. S. Yau, H. G. An and C. J. Hu, "Dynamic audit services for outsourced storages in Clouds". *IEEE Trans. Services Computing*, 6(2), pp. 227-238, 2013.

[13] Y. Yu, M.H. Au, Y. Mu, S.H. Tang, J. Ren, W. Susilo and L.J. Dong, "Enhanced privacy of a remote data integrity-checking protocol for secure cloud storage"*International Journal of Information Sececurity*. 14(4), pp.307-318, 2015.

[14] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing". *Proc. of IEEE INFOCOM*, pp.525-533, 2010.

[15] C. Wang, S. S.Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage". *IEEE Transactions on Computers*, 62, pp.362-375, 2013.

[16] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services", *IEEE Network*, 24, pp.19-24, 2010.

[17] Y. Yu, J.B. Ni, M. H. Au, Y. Mu, B.Y. Wang and H. Li. "Comments on a Public Auditing Mechanism for Shared Cloud Data Service". *IEEE Transantions on Services Computing*, 8(6),pp.998-999, 2015.

[18] A. Shamir. "Identity-based cryptosystems and signature schemes". *Advances in cryptology*. pp.47-53, 1985.

[19] J. N. Zhao, C. X. Xu, F. G. Li, and W. Z. Zhang, "Identity-Based Public Verification with Privacy-Preserving for Data Storage Security in Cloud Computing". *IEICE Transactions*, 96-A(12), pp.2709-2716, 2013.

[20] Y. Yu, Y. F. Zhang, Y. Mu, W. Susilo and H. Y. Liu, "Provably Secure Identity Based Provable Data Possession". *Provable Security*, pp.310-325, 2015.

[21] H. Q. Wang. "Identity-Based Distributed Provable Data Possession in Multicloud Storage". *IEEE Transactions on Services Computing*, 8(2), pp.328-340, 2015.

[22] A.Sahai and B. Waters. "Fuzzy identity-based encryption". *Advances in Cryptology-EUROCRYPT*, pp.457-473, 2005.

[23] P. Yang, Z. Cao and X. Dong. "Fuzzy identity based signature with applications to biometric authentication". *Computers and Electrical Engineering*, 37(4), pp.532-540, 2011.

[24] F. C. Guo, W. Susilo and Y. Mu. "Distance-Based Encryption: How to Embed Fuzziness in Biometric-Based Encryption". *IEEE Transactions on Information Forensics and Security*, 11(2), pp.247-257, 2016.

[25] C. Esposito, M. Ficco, F. Palmieri and A. Castiglione. "Smart cloud storage service selection based on fuzzy logic". *theory of evidence and game theory*, 65(8), pp.2348-2362, 2015.

[26] X. Li, J. Li and F. Huang. "A secure cloud storage system supporting privacy-preserving fuzzy deduplication". *Soft Computing*, 20(4), pp. 1437-1448, 2016.

[27] H. Fang, L. Xu and X. Huang. "Self-adaptive trust management based on game theory in fuzzy large-scale networks". *Soft Computing*, pp.1-15, 2015.

[28] P. Salil, S. Pankanti and A. K. Jain. "Biometric recognition: Security and privacy concerns." *IEEE Security and Privacy*, 2, pp.33-42, 2003.

[29] A. K. Jain, A. Ross and S. Prabhakar. "An introduction to biometric recognition". *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1), pp.4-20, 2004.

[30] D. Boneh and M. Franklin. "Identity-based encryption from the weil pairing", *Proc. of CRYPTO 2001*, LNCS 2139, pp.213-229, 2001.
[31] J. Katz. "Digital Signatures". *Springer Science and Business Media*, 2010.
[32] A. Shamir. "How to share a secret". *Communications of the ACM*, 22(11), pp.612-613, 1979.
[33] https://certivox.org/display/EXT/MIRACL.
[34] D. Freeman, M. Scott and E. Teske, "A taxonomy of pairing-friendly elliptic curves". *J. Cryptol*, 23(2), pp.224-280 (2010).

Yannan Li received her bachelor degree from University of Electronic Science and Technology of China in 2014. She is currently a master candidate at School of Computer Science and Engineering, University of Electronic Science and Technology of China. Her research interests are digital signature and secure cloud storage.

Yong Yu received his Ph.D. degree in cryptography from Xidian University in 2008. He is currently an associate professor of University of Electronic Science and Technology of China. His research interests are cryptography and its applications, especially public encryption, digital signature and secure cloud computing. He has published over 40 referred journal and conference papers.

Geyong Min is a Professor of High Performance Computing and Networking in the Department of Mathematics and Computer Science within the College of Engineering, Mathematics and Physical Sciences at the University of Exeter, United Kingdom. He received the PhD degree in Computing Science from the University of Glasgow, United Kingdom, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. His research interests include Future Internet, Computer Networks, Wireless Communications, Multimedia Systems, Information Security, High Performance Computing, Ubiquitous Computing, Modelling and Performance Engineering.

Willy Susilo received the Ph.D. degree in computer science from the University of Wollongong, Australia. He is currently a Professor and the Head of the School of Computing and Information Technology with the University of Wollongong, Australia. He is also the Director of the Centre for Computer and Information Security Research, University of Wollongong. His main research interests include cloud security, cryptography, and information security. He has received the prestigious ARC Future Fellowship from the Australian Research Council. He has served as a Program Committee Member in major international conferences.

Jianbing Ni received his master degree from University of Electronic Science and Technology of China in 2014. He is currently a Ph.D candidate at Department of Electrical and Computer Engineering, University of Waterloo. His research interests are digital signature and secure cloud storage.

Kim-Kwang Raymond Choo received the Ph.D. in Information Security from Queensland University of Technology, Australia. He is an associate professor at the University of South Australia, and a visiting expert at INTERPOL Global Complex for Innovation. He was named one of 10 Emerging Leaders in the Innovation category of The Weekend Australian Magazine / Microsoft's Next 100 series in 2009, and is the recipient of various awards including ESORICS 2015 Best Research Paper Award, Highly Commended Award from Australia New Zealand Policing Advisory Agency, British Computer Society's Wilkes Award, Fulbright Scholarship, and 2008 Australia Day Achievement Medallion.