The ISC Int'l Journal of Information Security
ISeCure

November 2022, Volume 14, Number 3 (pp. 43–49)
http://www.isecure-journal.org

PRESENTED AT THE ISCISC'2022 IN RASHT, IRAN.

Attribute-Based Remote Data Auditing and User Authentication for Cloud Storage Systems [☆]

Mohammad Ali [1,*]
[1] *Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran.*

ARTICLE INFO.

Keywords:
Cloud Computing, Data Integrity, Attribute-Based Authentication, Data Auditing, Attribute-Based Cryptography

Type:
Research Article

doi:
10.22042/isecure.2022.14.3.5

dor:
20.1001.1.20082045.2022.14.3.5.9

ABSTRACT

Remote data auditing (RDA) protocols enable a cloud server to persuade an auditor that it is storing a data file honestly. Unlike digital signature (DS) schemes, in RDA protocols, the auditor can carry out the auditing procedure without having the entire data file. Therefore, RDA protocols seem to be attractive alternatives to DSs as they can effectively reduce bandwidth consumption. However, existing RDA protocols do not provide adequately powerful tools for user authentication. In this paper, we put forward a novel attribute-based remote data auditing and user authentication scheme. In our proposed scheme, without having a data file outsourced to a cloud server, an auditor can check its integrity and the issuer's authenticity. Indeed, through a challenge-response protocol, the auditor can check whether 1) the cloud server has changed the content of the data file or not; 2) the data owner possesses specific attributes or not. We show that our scheme is secure under the hardness assumption of the bilinear Diffie-Hellman (BDH) problem.

1 Introduction

With the widespread cloud computing and storage applications, the ways of providing services are increasingly changing. Cloud computing technology provides users with distributed computing, convenient access, increased operational efficiencies, storage and computation resources, etc. [1]. These attractions have encouraged many companies to manage data, projects, contacts, etc., through this technology. A 2016 RightScale report has demonstrated that 95 percent of enterprises are running their applications by employing this technology. Also, surveys indicated that more than half of all organizations consider cloud computing as a necessary tool in their business models [2].

However, as a huge amount of personal data is outsourced to cloud servers, concern over data integrity and authentication arises. Digital signatures [3] are common tools to provide data integrity and user authentication services. However, the use of DS imposes high communication costs on the system as in these approaches the entirety of the data that is to be verified is required in the auditing procedure. To alleviate this problem, remote data auditing (RDA) protocols can be helpful. Employing an RDA protocol, an external auditor can verify the integrity of

[*] Corresponding author.

[☆] The ISCISC'2022 program committee effort is highly acknowledged for reviewing this paper.

Email address: mali71@aut.ac.ir

ISSN: 2008-2045 © 2022 ISC. All rights reserved.

data files outsourced to a cloud service provider without needing to have the data [4]. Generally, public key cryptosystems can be divided into three categories public key infrastructure (PKI)-based, identity-based, and attribute-based cryptosystems [5]. PKI-based schemes are considered the most basic approach in public key cryptography [5]. In PKI-based cryptosystems considering a trusted third-party, called certificate authority (CA), to validate and issue required certificates on users' public keys is a vital factor for running these systems. However, the use of certificate authority increases management costs significantly [5]. Identity-based cryptosystems can resolve this problem effectively [6]. In these systems, the identity of a user is considered as his/her public key. Therefore, users' public keys no longer need to be validated and certified. However, identity-based schemes have many inherent issues. For example, in these schemes, users' identities are revealed to other parties. It might threaten the users' privacy. Or as another example, in some cases, the selection of a unique identifier may not be friendly for users. To address these issues, attribute-based cryptosystems [7, 8] have been proposed. In these schemes, users' public keys correspond to a set of descriptive attributes [9, 10].

In the category of DSs, we see that there are several PKI-based, identity-based, and attribute-based signatures. However, in the RDA category, we have observed only PKI-based and identity-based RDA schemes. In this work, to address this issue, we design the first RDA scheme that supports the attribute-based user authentication service. Our main contributions are given below:

- **Anonymity**: In our designed scheme, data owners can be described concerning a set of descriptive attributes, and no information about the data owners' identities is leaked.
- **Remote data auditing**: In our proposed system, an auditor can verify the integrity of data files outsourced to a cloud server without needing to access the data.
- **Attribute-based authentication**: Our proposed approach is the first RDA scheme supporting attribute-based authentication. Indeed, our scheme is a suitable alternative for attribute-based signatures in cloud computing environments. The remaining of this work is organized as follows: In Section 2, we introduce some related work. We briefly describe the required preliminaries in Section 3. In Section 4, we describe the system model, threat model, and security requirements. The detailed construction is given in Section 5. In Section 6, we analyze our proposed scheme in terms of correctness and security. Finally, we conclude this work in Section 7.

2 Related Work

To verify the integrity of data outsourced to a cloud server, Deswarte *et al.* [11] put forward the concept of RDA protocols. In their proposed scheme, to prove data integrity, the cloud server has to hash the entirety of data files. However, it incurs prohibitive costs when the size of data files is large. Juels *et al.* [12] put forward the concept of Proof of Retrievability (PoR). However, their designed approach supports only limited times of integrity auditing. Ateniese *et al.* [13] independently introduced a similar concept named Provable Data Possession (PDP). Their scheme supports unlimited times of integrity verification and public auditing as well.

However, all of these schemes are based on the PKI which imposes considerable costs on the server side. To address this issue, Wang [14] designed an ID-based RDA scheme for multi-cloud storage environments. Yu *et al.* proposed a privacy-preserving ID-based RDA scheme for cloud storage [15]. Wang *et al.* [16] put forward an ID-based proxy oriented RDIC system.

However, the mentioned ID-based schemes do not provide anonymity, and identities are disclosed to the public. To address this issue, attribute-based cryptosystems seem to be promising solutions. The first formal definition of attribute-based signature (ABS) schemes was put forward by Maji *et al.* [17]. Li *et al.* [18] proposed a provable secure ABS scheme in the standard model. Herranz *et al.* [19] designed two ABS schemes with constant-size signatures. Chen *et al.* [20] designed the concept of outsourced ABS to lighten the computational burden. Sun *et al.* [21] proposed an outsourced decentralized ABS scheme for IoT networks.

However, none of the existing attribute-based authentication approaches support remote data auditing. To address this issue, we design the first attribute-based authentication scheme. It should be notified that the work presented in [22] does not support attribute-based authentication. Indeed, the scheme is designed to provide attribute-based access control in data auditing processes. Therefore, our work can be considered the first remote data auditing scheme supporting attribute-based authentication.

3 Preliminaries

Assume that $O \leftarrow \mathcal{A}(I)$ denotes execution of an algorithm $\mathcal{A}$ on input I and assign the output to O. Also, for an arbitrary set X, $x \leftarrow X$ means that an element $x \in X$ is selected uniformly at random from X. In the following, we briefly introduce some preliminaries required for studying the rest of this work.

Table 1. Notations

Notation	Description
λ	Security parameter of the system
$\mathbb{U}$	Universal attribute set
MPK	Master public key
MSK	Master secret key
Att_O	Attribute set of a DO
$P_m(Att)$	Threshold predicate associated with an attribute set Att and an integer m
M	A message
id_M	Identifier of a message M
SK_O	Secret key of a DO
(Tag_O, Tag'_O)	Tags generated by a DO
C	A challenge generated by an auditor
R	A response generated by the CS
Att_O	Set of a DO's attributes

3.1 Bilinear Pairings

Bilinear map: Consider two cyclic groups G_1 and G_2 of a prime order q. We say that a function $\hat{e} : G_1 \times G_1 \to G_2$ is a bilinear map if the following conditions hold [13]:

- **Bilinearity**: $\hat{e}(g^a, h^b) = \hat{e}(g^b, h^a) = \hat{e}(g, h)^{ab}$, for each $a, b \in \mathbb{Z}_q$ and $g, h \in G_1$.
- **Non-degeneracy**: There exists at least one $g \in G_1$ such that $\hat{e}(g, g) \neq 1$.
- **Computability**: There exists a polynomial-time algorithm which is able to compute $\hat{e}(g, h)$, for any $g, h \in G_1$.

Let $\mathcal{G}$ be a probabilistic polynomial-time (PPT) algorithm such that for a security parameter λ, $(\lambda, q, G_1, G_2, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$, where q, G_1, G_2, and $\hat{e}$ are the same as before. In this paper, we consider the following hardness assumption on $\mathcal{G}$.

3.2 Complexity Assumptions

Bilinear Diffie-Hellman (BDH) Assumption [13]: Consider a tuple $(\lambda, q, G_1, G_2, \hat{e}, g^\alpha, g^\beta, g^\gamma)$, where $(\lambda, q, G_1, G_2, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$, $g \leftarrow G_1$ and $\alpha, \beta, \gamma \leftarrow \mathbb{Z}_q$, this assumption says that the advantage of all PPT adversaries in calculation of $\hat{e}(g, g)^{\alpha\beta\gamma}$ is a negligible function in λ.

Computational Diffie-Hellman (CDH) Assumption [13]: Given a tuple $(\lambda, q, G_1, G_2, \hat{e}, g^\alpha, g^\beta)$, where $(\lambda, q, G_1, G_2, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$, $g \leftarrow G_1$ and $\alpha, \beta \leftarrow \mathbb{Z}_q$, this assumption states that there is no PPT adversary that can compute $g^{\alpha\beta}$ with a non-negligible advantage in λ.

4 Security Model

As depicted in Figure 1, four generic entities: central authority (CA), cloud server (CS), data owners (DO), and auditor participate in our designed system. In the following, we describe their main responsibilities.

- **CA**: This entity is the central processing unit of our system. It is responsible for initializing the parameters of the system and generating secret keys of DOs as well.
- **CS**: It possesses abundant computational and storage resources. Its primary responsibilities are to store the data received from DOs and prove their integrity to the auditor.
- **DOs**: Each DO is associated with a set of descriptive attributes. They obtain their secret keys according to their attributes from the CA, and to provide data integrity and prove their authenticity, they tag their data to be outsourced to the CS by using their obtained secret keys.
- **Auditor**: It is a third party that is responsible for verifying the integrity of outsourced data and the authenticity of DOs.

In the following, we present an overview of our designed system. It consists of four phases System initialization, Authorization, Data outsourcing, and Verification described below:

(1) **Initialization**: The CA executes this phase. In this phase, public parameters and the master secret key of the system are generated. Public parameters are published to the other entities and the master secret key is kept confidential by the CA.

(2) **Authorization**: In this phase, DOs request the CA to generate their attribute-based secret keys. The CA first verifies the attributes of DOs. If their correctness is confirmed, it gives their secret keys through a secure channel.

(3) **Data outsourcing**: This phase is executed by DOs. When a DO wants to outsource a data file to the CS, he/she first tags the file using his/her attribute-based secret key obtained in the previous phase. Then, he/she sends the data file, along with the generated tag, to the CS.

(4) **Verification**: In this phase, by using the tags assigned to the outsourced data files, the auditor checks the integrity and authenticity of their associated DOs through a challenge-response protocol with the CS.

4.1 Adversary Model

In our designed system, the CA is trustworthy. It does not generate improper secret keys for DOs and

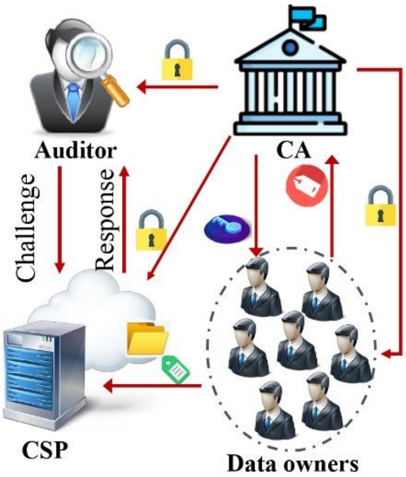

Figure 1. Workflow of our designed cryptosystem

does not collide with other entities. DOs are assumed to be malicious. They may collude with each other to forge a tag such that none of them can generate it independently. The CS is assumed to be untrustworthy in terms of both reliability and security. Indeed, it is possible that it erases or changes the outsourced data maliciously or accidentally. It may collude with DOs to forge integrity proofs.

In this work, we assume that the communication channels between the CA and DOs are secure such that the data transmitted through these channels neither might be tampered with nor eavesdropped on by other parties. Moreover, there is a tamper-resistant communication channel between any two other parties (i.e., the data transmitted through these channels may eavesdrop, but it is not definitely tampered with).

4.2 Security Requirements

In designing our data auditing approach, we consider the following security requirements:

- **Security against the server**: This requirement is to capture that if the CS has changed a data file, then it cannot forge an integrity proof for the data.
- **Anonymity**: This requirement states that the CS cannot learn any information about the attributes of DOs from the tags associated with the outsourced data.
- **Collusion resistance**: This requirement says that, for any predicate $P_m(Att)$ and any group of DOs colluding with each other, it is not feasible for the group to generate a valid tag associated with an attribute set Att', satisfying $P_m(Att)$, if none of the DOs has all attributes in Att'.

5 Our Construction

In the following, we present our proposed scheme in detail. Table 1 describes important notations used in this section. As mentioned before, our scheme consists of four phases described in detail as follows:

5.1 Initialization

The CA first selects a universal attribute set $\mathbb{U}$ and a security parameter λ. Then, it executes the algorithm $(MPK, MSK) \leftarrow$ **Setup**$(1^\lambda, \mathbb{U})$ to generate the master public parameters MPK and the master secret key MSK. It gives MPK to the other entities and keeps MSK confidential.

Setup$(1^\lambda, \mathbb{U})$: It first executes $(q, G_1, G_2, \hat{e}) \leftarrow \mathcal{G}(1^\lambda)$ and selects a random generator $g \leftarrow G_1$. Then, for each attribute $k \in \mathbb{U}$, it selects $s_k \leftarrow \mathbb{Z}_q$ and computes $PK_k = g^{s_k}$. Moreover, for a polynomial ℓ, it considers $N = \ell(\lambda)$ and a random N-degree polynomial Q such that $Q(0) = 0$. Also, using the polynomial interpolation technique, it constructs a $(|\mathbb{U}|-1)$-degree polynomial Q' such that $Q'(k) = sk_k^{-1}Q(k)$. Afterward, it chooses an integer n as the number of blocks in each message. Also, it considers secure hash functions $H : \{0,1\}^* \rightarrow G_1$ and $H^* : \{0,1\}^* \rightarrow \mathbb{Z}_q$. Finally, it returns $MPK = (\lambda, \mathbb{U}, n, N, q, G_1, G_2, \hat{e}, g, \{PK_k\}_{k\in\mathbb{U}}, H, H^*, Q')$ and $MSK = (\{s_k\}_{k\in\mathbb{U}}, Q)$.

5.2 Authorization

In this phase, any DO with an attribute set Att_O can request the CA to generate his/her attribute-based secret key. When the CA receives the request, it first checks whether DOs have the attributes or not. If so, the CA runs $SK_O \leftarrow$ **KeyGen**(MPK, MSK, Att_O) and gives the secret key SK_O to the DO. In the following, we describe the running process of the algorithm.

KeyGen(MPK, MSK, Att_O) : Given the master secret key $MSK = (\{s_k\}_{k\in\mathbb{U}}, Q)$ and an attribute set Att_O, this algorithm selects $U_O \leftarrow G_1$ and calculates $SK_{O,k} = U_O^{s_k^{-1}}$, for each $k \in Att_O$. This algorithm outputs a secret key $SK_O = (U_O, \{SK_{O,k}\}_{k\in Att_O})$.

5.3 Data Outsourcing

When a DO with an attribute set Att_O wants to outsource a data file M to the CS, he/she first considers a unique identifier id_M and a predicate $P_m(Att)$, where m is a natural number, and Att is an attribute set such that $|Att_O \cap Att| \geq m$ and $|Att \backslash Att_O| \geq N$. Then, he/she runs $(Tag_O, {Tag'}_O) \leftarrow$ **TagGen**$(MPK, P_m(Att), M, id_M, SK_O)$ to generate tags $(Tag_O, {Tag'}_O)$. Finally, $(M, Tag_O, {Tag'}_O)$

is given to the CS.

TagGen$(MPK, P_m(Att), M, id_M, SK_O)$: Given a predicate $P_m(Att)$, a message $M = (M^{(1)}, \ldots, M^{(n)}) \in \mathbb{Z}_q^n$, an identifier id_M, a secret key $SK_O = (U_O, \{SK_{O,k}\}_{k \in Att_O})$, this algorithm first selects $t_O, u_O \leftarrow \mathbb{Z}_q$ and calculates $Tag_{\mathbf{O}}^{(i)} = U_O^{M^{(i)}} \cdot H(id_M||i)^{t_O}$, for each $i = 1, \ldots, n$, $\omega_O = g^{t_O}$, $U'_O = U_O^{u_O}$, and $U''_O = g^{u_O^{-1}}$. Then, this algorithm considers a polynomial $Q''(x) = H^*(id_M) + a_1 x + \cdots + a_{k-1}x^{m-1}$, and computes $X_k = SK_{O,k}^{\frac{Q''(k)}{l_{Att \setminus Att_O,k}}}$, for each $k \in Att_O \cap Att$. Also, for any $k \in Att \setminus Att_O$, it computes $X_k = U_O^{\frac{Q'(k)}{l_{Att_O,k}}}$. Finally, it outputs (Tag_O, Tag'_O), where $Tag_O = \{Tag_O^{(i)}\}_{i=1}^n$ and $Tag'_O = (id_M, P_m(Att), \{(k, X_k)\}_{k \in Att}, \omega_O, U'_O, U''_O)$.

5.4 Verification

When the auditor wants to verify the integrity a data file M and the authenticity of the data owner, he/she first sends the associated identifier id_M to the CS and retrieve $Tag'_O = (id_M, P_m(Att), \omega_O, U'_O, U''_O, \{(k, X_k)\}_{k \in Att})$ corresponding to the data. Then, by running $b =$ **PreCheck**(MPK, Tag'_O) the auditor verifies whether attributes of the owner satisfies the predicate $P_m(Att)$ or not. If not, then the auditor aborts. Otherwise, he/she verifies the integrity of M by executing a challenge-response protocol. To this end, it first runs $(C, r) \leftarrow$ **Challenge**(MPK, Tag'_O) to obtain a challenge C and its corresponding secret r. Then, he/she sends C to the CS and keeps r confidential. Upon receiving C, the CS runs $(R_1, R_2) \leftarrow$ **Response**$(MPK, C, Tag_O, Tag'_O, M)$ to provide an integrity proof (R_1, R_2) as the response. Finally, the auditor can execute $b =$ **Vrfy**(MPK, C, R, Tag'_O) to check the integrity of M. If $b = 0$, he/she concludes that the data is not integrated. Otherwise, the auditor confirms the data. In the following, we describe the mentioned algorithms in detail.

PreCheck(MPK, Tag'_O): This algorithm takes as input a tag $Tag'_O = (id_M, P_m(Att), \{(k, X_k)\}_{k \in Att}, \omega_O, U'_O, U''_O)$ and returns 1 if and only if the following equation holds:

$$\prod\nolimits_{k \in Att} \hat{e}(PK_k, X_k)^{l_{Att,k}(0)} = \hat{e}(U''_O, U'_O)^{H^*(id_M)}, \tag{1}$$

where $l_{Att,k}(x) = \frac{\prod\limits_{i \neq k, i \in Att}(x-i)}{\prod\limits_{i \neq k, i \in Att}(k-i)}$ is the Lagrange polynomial.

Challenge(MPK, Tag'_O): This algorithm takes as input a tag $Tag'_O = (id_M, P_m(Att), \{(k, X_k)\}_{k \in Att}, \omega_O, U'_O, U''_O)$, it selects $r \leftarrow \mathbb{Z}_q$ and computes $c = g^r$ and $c' = {U''_O}^{-r}$. Then, it considers $\mathcal{L} \subset \{1, \ldots, n\}$ and selects $v_i \leftarrow \mathbb{Z}_q$, for each $i \in \mathcal{L}$. Finally, it returns $C = (c, c', \{v_i\}_{i \in \mathcal{L}})$ as the challenge.

Response$(MPK, C, Tag_O, Tag'_O, M)$: On input a challenge $C = (c, c', \{v_i\}_{i \in \mathcal{L}})$, a message $M = (M^{(1)}, \ldots, M^{(n)})$, two tags $Tag_O = \{Tag_O^{(i)}\}_{i=1}^n$ and $Tag'_O = (id_M, P_m(Att), \{(k, X_k)\}_{k \in Att}, \omega_O, U'_O, U''_O)$, it first computes the value $\mu = \sum_{i \in L} v_i M^{(i)}$ and returns a response $R = (R_1, R_2)$, where $R_1 = \hat{e}(\prod_{i \in L} (Tag_O^{(i)})^{v_i}, c)$ and $R_2 = \hat{e}({U'}_O^{-\mu}, c')$.

Vrfy(MPK, C, R, Tag'_O): Given $C = (c, c', \{v_i\}_{i \in \mathcal{L}})$, $Tag'_O = (id_M, P_m(Att), \{(k, X_k)\}_{k \in Att}, \omega_O, U'_O, U''_O)$, and a response $R = (R_1, R_2)$, this algorithm returns 1 if and only if the following equation holds:

$$R_1 R_2 = \prod\nolimits_{i \in L} \hat{e}(H(id_M||i)^{v_i}, \omega_O)^r. \tag{2}$$

6 Correctness and Security Analysis

In this section, we first analyze the correctness of our proposed scheme. Then, we describe the initial idea of proving its security.

6.1 Correctness Analysis

In the following, we show that the algorithms **PreCheck**(MPK, Tag'_O) and **Vrfy**(MPK, C, R, Tag'_O) work right. Let $Tag'_O = (id_M, P_m(Att), \{(k, X_k)\}_{k \in Att}, \omega_O, U'_O, U''_O)$ be an output of the **TagGen** algorithm, and Att_O satisfies $P_m(Att)$. We see that:

$$\begin{aligned}
\prod_{k \in Att} \hat{e}(PK_k, X_k)^{l_{Att,k}(0)} = & \prod_{k \in Att_O} \hat{e}(PK_k, X_k)^{l_{Att,k}(0)} \\
& \prod_{k \in Att \setminus Att_O} \hat{e}(PK_k, X_k)^{l_{Att,k}(0)} \\
= & \prod_{k \in Att_O} \hat{e}(PK_k, SK_{O,k}^{\frac{Q''(k)}{l_{Att \setminus Att_O,k}(0)}})^{l_{Att,k}(0)} \\
& \prod_{k \in Att \setminus Att_O} \hat{e}(PK_k, U_O^{\frac{Q'(k)}{l_{Att_O,k}(0)}})^{l_{Att,k}(0)} \\
= & \prod_{k \in Att_O} \hat{e}(g^{sk_k}, U_O^{\frac{sk_k^{-1} Q''(k)}{l_{Att \setminus Att_O,k}(0)}})^{l_{Att,k}(0)} \\
& \prod_{k \in Att \setminus Att_O} \hat{e}(g^{sk_k}, U_O^{\frac{sk_k^{-1} Q(k)}{l_{Att_O,k}(0)}})^{l_{Att,k}(0)} \\
= & \prod_{k \in Att_O} \hat{e}(g, U_O^{Q''(k)})^{\frac{l_{Att,k}(0)}{l_{Att \setminus Att_O,k}(0)}} \\
& \prod_{k \in Att \setminus Att_O} \hat{e}(g, U_O^{Q(k)})^{\frac{l_{Att,k}(0)}{l_{Att_O,k}(0)}}
\end{aligned}$$

$$
\begin{aligned}
&= \prod_{k\in Att_O} \hat{e}(g, U_O^{Q''(k)})^{l_{Att_O,k}(0)} \\
&\prod_{k\in Att\setminus Att_O} \hat{e}(g, U_O^{Q(k)})^{l_{Att\setminus Att_O,k}(0)} \quad = \prod_{k\in Att_O} \hat{e}(g, U_O)^{Q''(k) l_{Att_O,k}(0)} \\
&\prod_{k\in Att\setminus Att_O} \hat{e}(g, U_O)^{Q(k) l_{Att\setminus Att_O,k}(0)} \\
&= \hat{e}(g, U_O)^{\sum\limits_{k\in Att_O} Q''(k) l_{Att_O,k}(0)} \\
&\hat{e}(g, U_O)^{\sum\limits_{k\in Att\setminus Att_O} Q(k) l_{Att\setminus Att_O,k}(0)} \\
&= \hat{e}(g, U_O)^{H^*(id_M)} \hat{e}(g, U_O)^0 \\
&= \hat{e}(g, U_O)^{H^*(id_M)} \\
&= \hat{e}(g^{u_O^{-1}}, U_O^{u_0})^{H^*(id_M)} \\
&= \hat{e}(U''_O, U'_O)^{H^*(id_M)}.
\end{aligned}
$$

Therefore, Equation 1 holds and the algorithm **PreCheck**(MPK, Tag'_O) is correct. Now, we prove the correctness of **Vrfy**(MPK, C, R, Tag'_O). To this end, we prove Equation 2. Let $C = (c, c', \{v_i\}_{i\in\mathcal{L}})$ and $R = (R_1, R_2)$ be a valid challenge and its corresponding response, respectively. We observe that:

$$
\begin{aligned}
R_1 &= \hat{e}(\prod\nolimits_{i\in L} (Tag_O^{(i)})^{v_i}, c) \\
&= \hat{e}(\prod\nolimits_{i\in L} U_O^{v_i M^{(i)}} . H(id_M||i)^{t_O v_i}, c) \\
&= \prod\nolimits_{i\in L} \hat{e}(U_O^{v_i M^{(i)}}, c). \prod\nolimits_{i\in L} \hat{e}(H(id_M||i)^{t_O v_i}, c) \\
&= \hat{e}(U_O^{\mu}, g^r) \prod\nolimits_{i\in L} \hat{e}(H(id_M||i)^{t_O v_i}, g^r) \\
&= \hat{e}(U_O^{u_O \mu}, g^{r u_O^{-1}}) \prod\nolimits_{i\in L} \hat{e}(H(id_M||i)^{v_i}, g^{t_O})^r \\
&= \hat{e}(U'^{\mu}_O, U''^{r}_O) \prod\nolimits_{i\in L} \hat{e}(H(id_M||i)^{v_i}, \omega_O^{(2)})^r \\
&= \hat{e}(U'^{\mu}_O, c') \prod\nolimits_{i\in L} \hat{e}(H(id_M||i)^{v_i}, \omega_O^{(2)})^r \\
&= \hat{e}(U'^{\mu}_O, c') \prod\nolimits_{i\in L} \hat{e}(H(id_M||i)^{v_i}, \omega_O^{(2)})^r.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
R_1 R_2 &= \hat{e}(U'^{\mu}_O, c') \prod\nolimits_{i\in L} \hat{e}(H(id_M||i)^{v_i}, \omega_O^{(2)}) \hat{e}(U'^{-\mu}_O, c') \\
&= \prod\nolimits_{i\in L} \hat{e}(H(id_M||i)^{v_i}, \omega_O^{(2)})^r.
\end{aligned}
$$

6.2 Security Analysis

In this section, we briefly describe the initial idea in the security proof of our proposed scheme. We consider the following security attacks:

(1) **Attack 1**: The CS changing the outsourced data tries to forge an integrity proof of the data by colluding malicious DOs.

(2) **Attack 2**: Given a predicate$P_m(Att)$, a group of DOs, such that none of them doesn't have an attribute set satisfying$P_m(Att)$, may try to generate a tag associated with an attribute set Att_O satisfying$P_m(Att)$.

Assume that the CS can succeed in attack 1. Let $\mathcal{A}$ be a PPT adversary that aims to solve the BDH problem. Indeed, on input $(q, G_1, G_2, \hat{e}, g^{\alpha}, g^{\beta}, g^{\gamma})$, $\mathcal{A}$ tries to compute the value$\hat{e}(g, g)^{\alpha\beta\gamma}$. To this end, assuming H is modeled as a random oracle such that $H(id_M||i) = g^{x_i\gamma}$, for a known value$x_i \in \mathbb{Z}_q$,$c = g^{\beta}$, and $\omega_O = g^{\alpha}$, then we see that succeeding in **Attack 1** implies solving the BDH problem. Indeed:

$$
\begin{aligned}
R_1 R_2 &= \prod\nolimits_{i\in L} \hat{e}(H(id_M||i)^{v_i}, \omega_O^{(2)})^r \\
&= \prod\nolimits_{i\in L} \hat{e}(g^{\gamma x_i v_i}, g^{\alpha})^{\beta} \\
&= \hat{e}(g, g)^{\alpha\beta\gamma \sum_{i\in L} x_i v_i}.
\end{aligned}
$$

Now, as $\mathcal{A}$ knows the value $\sum_{i\in L} x_i v_i$, it can compute $\hat{e}(g, g)^{\alpha\beta\gamma}$.

Also, if a group of DOs can succeed in **Attack 2** (i.e., they can forge Tag'_O such that $1 =$ **PreCheck**(MPK, Tag'_O)), then we see that the DH problem can be solved. Indeed, assuming that $U'_O = g^{\alpha}$ and $U''_O = g^{\beta}$, then in this case we have

$$
\begin{aligned}
&\prod\nolimits_{k\in Att} \hat{e}(PK_k, X_k)^{l_{Att,k}(0)} = \hat{e}(U''_O, U'_O)^{H^*(id_M)} \\
&\Rightarrow \hat{e}(g, g^{xy}) = \hat{e}(g^x, g^y) \\
&= \hat{e}(U''_O, U'_O) \\
&= \prod\nolimits_{k\in Att} \hat{e}(PK_k, X_k)^{l_{Att,k}(0) H^{*-1}(id_M)} \\
&= \prod\nolimits_{k\in Att} \hat{e}(g^{sk_k}, X_k^{l_{Att,k}(0) H^*(id_M)^{-1}}) \\
&= \hat{e}(g, \prod\nolimits_{k\in Att} X_k^{s_k\, l_{Att,k}(0) H^{*-1}(id_M)}) \\
&\Rightarrow g^{xy} = \prod\nolimits_{k\in Att} X_k^{s_k\, l_{Att,k}(0) H^{*-1}(id_M)}.
\end{aligned}
$$

Now, assuming that $\mathcal{A}$ has initialized our scheme and therefore knows $\{s_k\}_{k\in Att}$, we see that $\mathcal{A}$ can solve g^{xy}.

7 Conclusion

In this paper, we proposed a new primitive called attribute-based remote data auditing and user authentication for cloud storage systems. Our proposed scheme provides the following services: I) data owners can outsource their data to a cloud server anonymously; II) attributes of data owners can be efficiently authenticated; III) the integrity of outsourced data can be verified without having the data as input. Moreover, we provided correctness proof and showed that the security of the scheme can be proved under the hardness assumption of the bilinear Diffie-Hellman (BDH).

References

[1] M. Ali, J. Mohajeri, M.-R. Sadeghi, and X. Liu, "A fully distributed hierarchical attribute-based encryption scheme," Theor. Comput. Sci., vol. 815, pp. 25–46, 2020.

[2] V. Chang, M. Ramachandran, Y. Yao, Y.-H. Kuo, and C.-S. Li, "A resiliency framework for an enterprise cloud," Int. J. Inf. Manage., vol. 36, no. 1, pp. 155–166, 2016.

[3] J. Katz and Y. Lindell, Introduction to modern cryptography, 3rd ed. Boca Raton, FL: CRC Press, 2020.

[4] M. Ali and X. Liu, "Lightweight verifiable data management system for cloud-assisted wireless body area networks," Peer Peer Netw. Appl., vol. 15, no. 4, pp. 1792–1816, 2022.

[5] S. Boonkrong, Authentication and access control: Practical cryptography methods and tools, 1st ed. Berlin, Germany: APress, 2020.

[6] A. Shamir, "Identity-based cryptosystems and signature schemes," in Advances in Cryptology, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 47–53.

[7] M. Ali, J. Mohajeri, M.-R. Sadeghi, and X. Liu, "Attribute-based fine-grained access control for outscored private set intersection computation," Inf. Sci. (Ny), vol. 536, pp. 222–243, 2020.

[8] M. Ali and M.-R. Sadeghi, "Provable secure lightweight attribute-based keyword search for cloud-based Internet of Things networks: Provable secure lightweight attribute-based keyword search for cloud-based Internet of Things networks," Trans. emerg. telecommun. technol., vol. 32, no. 5, p. e3905, 2021.

[9] M. Ali, M.-R. Sadeghi, and X. Liu, "Lightweight fine-grained access control for wireless body area networks," Sensors (Basel), vol. 20, no. 4, p. 1088, 2020.

[10] M. Ali, M.-R. Sadeghi, and X. Liu, "Lightweight revocable hierarchical attribute-based encryption for internet of things," IEEE Access, vol. 8, pp. 23951–23964, 2020.

[11] Y. Deswarte, J.-J. Quisquater, and A. Saïdane, "Remote integrity checking," in Integrity and Internal Control in Information Systems VI, Boston: Kluwer Academic Publishers, 2005, pp. 1–11.

[12] A. Juels and B. S. Kaliski, "Pors." Proceedings of the 14th ACM conference on Computer and communications security - CCS , 2007, doi: 10.1145/1315245.1315317.

[13] G. Ateniese, "Provable data possession at untrusted stores." Proceedings of the 14th ACM conference on Computer and communications security - CCS , 2007, doi: 10.1145/1315245.1315318.

[14] H. Wang, "Identity-Based Distributed Provable Data Possession in Multicloud Storage." IEEE Transactions on Services Computing, vol. 8, no. 2, pp. 328-340, 2015, doi: 10.1109/tsc.2014.1.

[15] Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, G. Min, Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage, IEEE Transactions on Information Forensics and Security 12 (4) (2016) 767–778.

[16] H. Wang, D. He, and S. Tang, "Identity-Based Proxy-Oriented Data Uploading and Remote Data Integrity Checking in Public Cloud." IEEE Transactions on Information Forensics and Security, vol. 11, no. 6, pp. 1165-1176, 2016, doi: 10.1109/tifs.2016.2520886.

[17] H. K. Maji, M. Prabhakaran, M. Rosulek, Attribute-based signatures: Achieving attribute-privacy and collusion-resistance., IACR Cryptology ePrint Archive 2008 (2008) 328.

[18] J. Li, M. H. Au, W. Susilo, D. Xie, K. Ren, Attribute-based signature and its applications, in: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ACM, 2010, pp. 60–69.

[19] J. Herranz, F. Laguillaumie, B. Libert, C. Ràfols, Short attributebased signatures for threshold predicates, in: Cryptographers' Track at the RSA Conference, Vol. 7178, Springer, 2012, pp. 51–67.

[20] X. Chen, J. Li, X. Huang, J. Li, Y. Xiang, D. S. Wong, Secure outsourced attribute-based signatures, IEEE Transactions on Parallel and Distributed Systems 25 (12) (2014) 3285–3294.

[21] J. Sun, Y. Su, J. Qin, J. Hu, J. Ma, Outsourced decentralized multiauthority attribute based signature and its application in IoT, IEEE Transactions on Cloud Computing 9 (3) (2019) 1195–1209.

[22] Y. Yu, Y. Li, B. Yang, W. Susilo, G. Yang, J. Bai, Attribute-based cloud data integrity auditing for secure outsourced storage, IEEE Transactions on Emerging Topics in Computing 8 (2) (2017) 377– 390.

Mohammad Ali received the M.Sc. degree in applied mathematics, in 2016, and the Ph.D. degree in mathematics from Amirkabir University of Technology, Tehran, Iran, in 2020. He is currently an Assistant Professor at the Department of Mathematics and Computer Science, Amirkabir University of Technology. His research interests include provable security cryptography, post-quantum cryptography, cloud security, and IoT security.