Efficient Revocable Attribute-based Encryption with Verifiable Data Integrity

IEEE Publication Technology, *Staff, IEEE,*

***Abstract*—Nowadays, cloud computing and cloud storage services that can reduce the local workload are becoming increasingly popular, allowing individuals and businesses to upload data to the cloud. Since the permissions of users in the system are not immutable, the data users should have dynamic access and revocation of users who have been granted access to data is also a strong need for cloud computing systems. In addition, we should ensure the data integrity after the cloud server performs an revocation. In order to solve above issues, we propose a revocable attribute-based encryption scheme that protects the integrity of the encrypted data. The proposed scheme is more efficient compared to existing revocable attribute-based encryption with data integrity (RABE-DI) scheme. In addition, we prove the confidentiality and integrity of the scheme under the defined security model. Experimental results show that our scheme outperforms similar schemes in terms of efficiency.**

***Index Terms*—attribute-based encryption; full security; decisional linear assumption; data integrity;**

I. Introduction

CLOUD computing and cloud storage services have become increasingly popular among individuals and businesses in recent years because of their economical and efficient features. People store their huge data in the cloud or outsource heavy computing tasks to the cloud while the cloud service providers charge for it. This may seem like a great give-and-take business partnership but some problems arise. In such an environment, the confidentiality and integrity of the user's data are challenged. Data confidentiality is usually addressed by encrypting the data and data integrity requires verification of data. However, data on the cloud is usually shared by many users, which requires a one-to-many cryptographic primitive. Attribute-based encryption is a widely known cryptographic primitive that solves this problem [1]. ABE has been proven to be very suitable for access control and it is used in a wide variety of scenarios such as paid broadcasting, cloud services [2], and medical data access control [3], [4]. Since the original idea of ABE was proposed by Sahai and Waters et al. in 2005 [5], the first practical CP-ABE was proposed by Waters in 2011 [6]. ABE has been widely studied in the past decade or so and has been extended with various functions besides access control, such as hidden access structures [7], dynamic credentials [8]–[10], privacy protection [11], outsourcing computation [12]–[14], etc. In order to improve efficiency, some ABE schemes have been proposed in recent years [15]–[18]. Fast Attribute-based Message Encryption (FAME) was proposed by Agrawal et al. in 2017 [16], which is an unbounded attribute space ABE scheme with quick decryption. FAME has many advantages as one of the state-of-the-art ABE schemes. In FAME scheme, arbitrary strings can be used as attributes and decryption inherently requires only 6 pairing operations. It is very efficient compared to the ABE scheme which has a linear relationship with the number of attributes at the decryption stage. FAME is also proven to be fully secure, compared to some classical schemes [3], [19], [20] which are only selective secure.

Another issue is that the access structure determines what attribute set can satisfy it. In CP-ABE, the plaintext is encrypted into ciphertext under a particular access structure. The access structure embedded in the ciphertext is generated at the time of encryption and remains unchanged. How to revoke access rights from some users by changing the access structure is a challenging problem. The data owner, as the person who encrypts the data, should be able to control and decide who has access to that data.

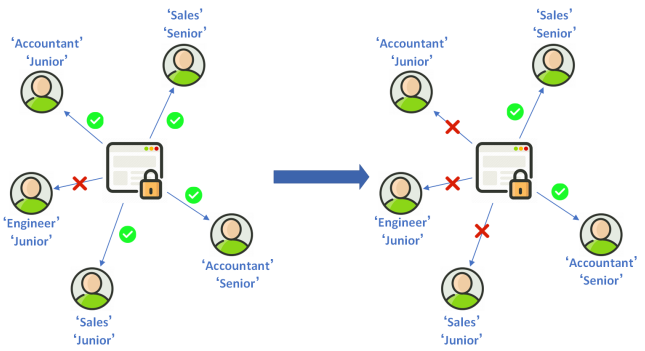

Fig. 1. System architecture of our RFAME-DI scheme

To illustrate this situation with an example, consider that in a company the annual sales data is encrypted by the access structure $\mathcal{T} = (\text{“}accountant\text{”} \vee \text{“}sales\text{”})$. The company's accountant and all sales of the marketing department have access to the report. Subsequently, the company managers, who is also the data owner, want to upgrade the access to protect the privacy and avoid malicious competition between the employees. The new regulation requires that only management-level employees of the company have access to the report, that is, those with 'senior' attribute. The access structure for the encrypted report was therefore updated to the new access structure $\mathcal{T}' = (\text{“}accountant\text{”} \vee \text{“}sales\text{”}) \wedge (\text{“}senior\text{”})$. After this change, the junior sales no longer have access to this report. This process of revocation is done by a cloud server,

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received April 19, 2021; revised August 16, 2021.

so the data integrity after revocation needs to be guaranteed. Fig.1 depicts this example. In our model, we consider that cloud servers stores the data uploaded by users correctly but may perform computational tasks inactively or incorrectly to save their computational resources, thus failing to ensure the data integrity.

A. Motivations and Contributions

Most of the existing revocable attribute based encryption (RABE) schemes cannot guarantee data integrity. Ge et al. [21] proposed a RABE scheme with data integrity, but their solutions are not efficient enough in order to achieve protection of data integrity. Therefore, our goal is to improve the efficiency of revocation and decryption while protecting data integrity. FAME [16] is suitable to be used as the underlying scheme, which is more efficient in both encryption and decryption. Our work is focused on how to implement revocation and protect the integrity of data on FAME. Our main contribution is as follows.

(1) We propose an efficient revocable ABE scheme that guarantees the data integrity. If the cloud server does not perform revocation correctly, it will be detected by the data user.
(2) The process of revocation does not require the data owner to perform any decryption and encryption. Data owner only needs to provide the cloud server with a ciphertext delegation to tell it how to revoke by giving a delegation.
(3) We prove that the security of our scheme can be reduced to FAME [16] which is fully secure. At the same time, we prove that our scheme guarantees data integrity.
(4) Experiments are done to evaluate the execution time of key generation, encryption, revocation, and decryption algorithms and compare the RABE-DI scheme in [21]. The performance of our RFAME scheme in revocation and decryption are efficient.

B. Related work

Sahai and Waters [5] proposed the idea of attribute-based encryption, the condition for decryption is that the attribute set needs to satisfy the access structure. According to whether the access structure is embedded in ciphertext or private key, attribute encryption is classified into key-policy attribute-based encryption (KP-ABE) [22], [23] and ciphertext-policy attribute-based encryption (CP-ABE) [3], [24], [25].

Revocation of attributes is an important topic in ABE research but it is not an easy task to do this, as update of a single attribute may involve a large number of users. From the way of revocation, it can be divided into direct revocation and indirect revocation. Direct revocation [26]–[28] is usually implemented in two ways, one is to add a timestamp to the key. The key generation center (KGC) broadcasts periodically to update the key and the other is to embed a list of all revoked users in the ciphertext, but faces with the problem of excessive ciphertext size. Liu et al. [26] proposed a direct revocation CP-ABE scheme. It embeds the revocation list into the ciphertext, but this leads to an increasing size of the ciphertext over time. To reduce the length of the ciphertext, it remove the expired part of the revocation list. Indirect revocation [29]–[32] splits the user's decryption privileges into two parts, the key and the key update material. The revoked user does not receive the key update material from KGC anymore and loses the decryption privileges. Cui et al. [33] proposed an indirect revocation method, which improves on the revocation method [34] by combining a binary tree data structure and fuzzy identity-based encryption [5]. The method [34] reduces the size of the key update from linear to logarithmic. Further improvement of the method in [33] allows transferring the revocation-induced computation task to an untrusted server and the data user only needs to keep a constant-sized private key locally. There are still some problems worth thinking about revocation. For example, how to revoke malicious users quickly, and how to reduce the computational cost of the revocation process. The scheme [33] reduces the computation cost on the user side with the help of auxiliary servers. Another issue is that the newly published ciphertext cannot be decrypted by the revoked user, also known as revocation with forward security [35].

Due to the cloud server may perform revocation operations dishonestly, data integrity needs to be considered during the revocation process, which means that the encrypted data in the original ciphertext and the new ciphertext generated by revocation need to be ensured the same. Some RABE schemes [36], [37] are not able to guarantee data integrity. A common solution to this problem is to verify the message being encrypted. Lai et al. [38] proposed a method to verify data integrity by giving a commitment to the message and a random message in the ciphertext. We use this verification method to ensure the data integrity of our scheme.

Proxy re-encryption is a cryptographic technique that is common in cloud services, and Attribute-based Proxy Re-encryption (ABPRE) is one of proxy encryption technologies [39]–[42]. Data owner delegates the re-encryption capability to a proxy server, which encrypts again on the ciphertext, achieving the effect that the original data is directly encrypted under the new access structure. Proxy servers reduce the computational pressure on users, but also bring credibility issues. It is necessary to verify the results after the server performs the task. Recently, Ge et al. proposed a verifiable proxy re-encryption scheme that allows users to detect and refuse to pay for services if the proxy server does not perform correctly [42]. Deng et al. proposed a technique where a proxy server re-encrypts a ABE ciphertext to a ciphertext encrypted by identity-based encryption, thus reducing the decryption cost at the user end [39].

II. PRELIMINARIES

A. Bilinear pairing

Let $(p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e, g, h)$ be a prime order bilinear group system. $\mathbb{G}$, $\mathbb{H}$ and $\mathbb{G}_T$ are multiplicative cyclic groups of prime order p. g and h are generator of the group $\mathbb{G}$ and $\mathbb{H}$, respectively. $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$ is a bilinear pairing if the following properties are satisfied.

(1) Bilinear:$e(x^u, y^v) = e(x, y)^{uv}$ for all $x \in \mathbb{G}, y \in \mathbb{H}$ and $u, v \in \mathbb{Z}_p^*$;
(2) Non-degenerate: $e(x, y) \neq 1$ whenever $x, y \neq 1_{\mathbb{G}}$;

(3) Computable: $e(x, y)$ can be efficiently computed for all $x \in \mathbb{G}, y \in \mathbb{H}$.

B. Complex assumption

Discrete Logarithm Assumption [43]. Let $(e, \mathbb{G}, \mathbb{G}_T, g, p)$ be a prime order bilinear group system. Given a tuple $(e, \mathbb{G}, \mathbb{G}_T, p, g, g^{\delta})$ where $g \in \mathbb{G}, \delta \in \mathbb{Z}_p^*$. The discrete logarithm assumption means that the advantage of a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ to find the integer δ is negligible. Formally, the advantage of a PPT adversary $\mathcal{A}$

$$Pr[\mathcal{A}(e, \mathbb{G}, \mathbb{G}_T, p, g, g^{\delta}) = \delta]$$

is negligible.

C. Access structure

The access structure sometimes called a policy, requires a attribute set to be granted permissions when its requirements are met. The formal definition is as follows.

Definition 1. *If $\mathcal{U}$ denotes the universe of attributes, then an access structure $\mathbb{A}$ is a collection of non-empty subsets of $\mathcal{U}$, i.e., $\mathbb{A} \subseteq 2^{\mathcal{U}} \setminus \{0\}$. It is called monotone if for every $B, C \subseteq \mathcal{U}$ such that $B \subseteq C, B \in A \Rightarrow C \in A$.*

From the point of comprehensibility, monotonicity suggests that a larger attribute set implies greater privileges, so adding attributes to a attribute set does not reduce its privileges but only makes it more powerful.

A Boolean formula $\mathcal{T}$ corresponds to an access structure $\mathbb{A} = (M, f)$ where M is a matrix and f is a function. The Boolean formula $\mathcal{T}$ determines the number of rows and columns of matrix M and f is a function that maps the row of matrix M to an attribute. Some simple and efficient conversion methods were proposed in some literature [36], [44]. In [45], Liu and Cao proposed an efficient algorithm to convert the Boolean formula into as small as possible LSSS matrix thus reducing the cost of communication.

The scheme in this paper does not involve the secret recovery of linear secret sharing scheme (LSSS) [46], but only the linear reconfiguration property of LSSS, defined as follows.

Let $\mathcal{S}$ be an attribute set and $I = \{i | i \in \{1, ..., n_1\}, f(i) \in \mathcal{S}\}$ be the set of rows in M that corresponding to $\mathcal{S}$. If there exists a linear combination of rows in the matrix M that yields $(1, 0, 0, ..., 0)$, then we say that the attribute set $\mathcal{S}$ satisfies the access structure (M, f). More formally, if the attribute set $\mathcal{S}$ satisfies the access structure (M, f), then there must exist a set of constant coefficients $\{\theta_i\}_{i \in I}$ such that the following equation holds $\sum_{i \in I} \theta_i (M)_i = (1, 0, 0, ..., 0)$ where $(M)_i$ is the ith row of M.

We use $(\widetilde{M}, \widetilde{f})$ to denote the delegation access structure corresponds to $\widetilde{\mathcal{T}}$. The revoked boolean formula $\mathcal{T}^{'}$ corresponding to $(M^{'}, f^{'})$ is $\mathcal{T}^{'} = (\mathcal{T} AND \widetilde{\mathcal{T}})$.

III. SYSTEM ARCHITECTURE AND DEFINITIONS

A. System architecture

Our revocable attribute-based encryption system requires four entities, which are data owner, cloud server, trusted authority center, and data user.

(1) The authority center is responsible for the initialization of the whole system and generating all public parameters according to the security parameters. The authority center also uses the master private key to generate private keys for the data users.
(2) The data owner makes the access structure and decides who can access its data accordingly. The data owner encrypts the data under the specified access structure and then uploads the ciphertext to the cloud server.
(3) The cloud server is responsible for storing the ciphertext uploaded by the data owner and performing the revocation operation.
(4) Data users can download the ciphertext from the cloud server and recover the plaintext with their own private key. Data users also can verify the correctness of the ciphertext.

B. Threat model

In this threat model [21] there are two kinds of adversaries, one adversary, which can be any entity, wants to compromise the confidentiality of the data, it does not have a valid key but wants to get the information about the plaintext from the ciphertext. The other adversary, usually a cloud server, tries to generate incorrect revoked ciphertext to corrupt the data integrity. The security models corresponding to these two security properties will be defined in detail subsequently. Threat model assumes that the cloud server does not collude with the revoked user, that is, the cloud server does not send the original ciphertext to the revoked user because the revoked user can decrypt the original ciphertext directly.

C. Syntax of Revocable ABE with Data Integrity (RABE-DI)

Our revocable attribute-based encryption scheme consists of the following seven algorithms. The process of execution is depicted in Fig.2.

$\mathrm{Setup}(1^{\lambda})$: The authority center takes the security parameters λ as input and outputs the system public parameters PP and the master private key msk.

$\mathrm{KeyGen}(\mathrm{msk}, \mathcal{S})$: The authority center takes the master private key msk and the attribute set $\mathcal{S}$ as input and outputs the private key sk corresponding to the attribute set $\mathcal{S}$ for the data user.

$\mathrm{Encrypt}(\mathrm{m}, \mathbb{A} = (\mathrm{M}, \mathrm{f}))$: The data owner encrypts the message m under the access structure (M, f) and then outputs the ciphertext CT.

$\mathrm{Delegate}(\widetilde{\mathbb{A}})$: The data owner takes the access structure $\widetilde{\mathbb{A}}$ as input, and then computes and outputs the delegation DG based on the new attributes involved in $\widetilde{\mathbb{A}}$.

$\mathrm{Revoke}(\mathrm{CT}, \widetilde{\mathbb{A}} = (\widetilde{\mathrm{M}}, \widetilde{\mathrm{f}}))$: The cloud server takes the original ciphertext CT under the access structure (M, f) and the revocation access structure $(\widetilde{M}, \widetilde{f})$ as input and outputs the revoked ciphertext $CT^{'}$ under the access structure $(M^{'}, f^{'})$. The Boolean formula corresponding to $(M^{'}, f^{'})$ here is $\mathcal{T}^{'} = (\mathcal{T} AND \widetilde{\mathcal{T}})$, where $\mathcal{T}$ and $\widetilde{\mathcal{T}}$ correspond to access structure (M, f) and $(\widetilde{M}, \widetilde{f})$, respectively.

$\mathrm{Decrypt}_{\mathrm{or}}(\mathrm{sk}, \mathrm{CT})$: The data user takes their private key sk and the original ciphertext CT downloaded from the

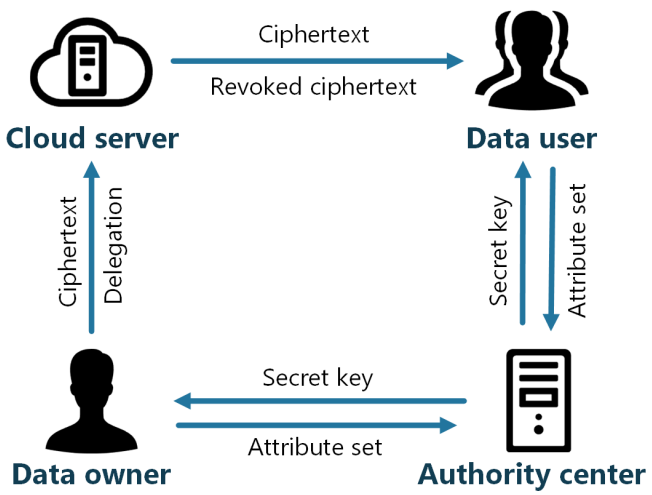

Fig. 2. System architecture of our RFAME-DI scheme

cloud server as input, where the private key corresponds to the attribute set $\mathcal{S}$ and the access structure contained in the ciphertext CT is (M, f). Then outputs the message m if the attribute set $\mathcal{S}$ satisfies the access structure (M, f), otherwise it outputs $\perp$.

$\mathrm{Decrypt}_{\mathrm{re}}(\mathrm{sk}^{'}, \mathrm{CT}_{\mathrm{csum}}, \mathrm{CT}^{'})$: The data user takes their private key $sk^{'}$ which is different from sk, a part of the original ciphertext CT_{csum} (Explicitly, it is the checksum in the original ciphertext.) and the revoked ciphertext $CT^{'}$ downloaded from the cloud server as input if the attribute set $\mathcal{S}^{'}$ about the private key $sk^{'}$ satisfies the access structure $(M^{'}, f^{'})$ in the ciphertext $CT^{'}$, then output the message m, otherwise it outputs $\perp$.

D. Security model

The two security models required for the RFAME-DI scheme are described separately below.

IND-CPA security.

Informally, if no algorithm can distinguish between m_0 and m_1 encrypted under the chosen access structure $\mathbb{A}^*$, the scheme resists the chosen plaintext attack (CPA), as long as the algorithm is not authorized with the corresponding decryption key. Such an attack occurs at any stage of the cryptographic scheme, so the choice of the attacked access structure $\mathbb{A}^*$ is influenced by the public parameters and the private key possessed by the adversary. The scheme considered in this case will obtain adaptive or full security. In a weaker model called selective security, the access structure $\mathbb{A}^*$ is selected before the system is deployed so as to prevent CPA attacks, such an idealized definition is unlikely to occur in reality.

A revocable attribute-based encryption with data integrity scheme is secure against chosen plaintext attack (CPA) if the advantage of an adversary $\mathcal{A}$ in the following game is negligible.

Setup: In this step, the challenger $\mathcal{C}$ generates public parameters and master secret key by executing the Setup algorithm. Then, $\mathcal{C}$ passes public parameters to $\mathcal{A}$.

Query: $\mathcal{A}$ makes the secret key query and gets the secret keys $sk_{\mathcal{S}_1}, ..., sk_{\mathcal{S}_{q_1}}$, where $\mathcal{S}_i$ can't satisfy the access structure (M^*, f^*) that will be challenged for $i \in \{1, 2, ..., q_1\}$. $\mathcal{A}$ also makes delegation text query and get a series of delegation texts $dt_1, dt_2, ..., dt_{q_2}$.

Challenge: Two massage (m_0, m_1) with equal length are selected by $\mathcal{A}$ and then $\mathcal{A}$ sends them to the challenger $\mathcal{C}$. $\mathcal{C}$ computes the challenge ciphertext $CT^* = Enc(m_\sigma, (M^*, f^*))$ where $\sigma \in \{0, 1\}$, and returns it to $\mathcal{A}$.

Query: This phase is the same as the previous query.

Guess: $\mathcal{A}$ outputs its guess $\sigma^{'}$. The adversary $\mathcal{A}$'s advantage to win the IND-CPA security game is defined as $Adv_{\mathcal{A}}^{IND-CPA}(\lambda) = |Pr[\sigma^{'} = \sigma] - 1/2|$.

Integrity.

The data integrity of revocable attribute-based encryption with data integrity scheme is defined by the following game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$. The scheme guarantees data integrity if the advantage of an adversary $\mathcal{A}$ is negligible.

Setup: In this step, the challenger $\mathcal{C}$ generates public parameters and master secret key by executing the $Setup$ algorithm. Then, $\mathcal{C}$ passes public parameters to $\mathcal{A}$.

Query: $\mathcal{A}$ makes the secret key query and gets the secret keys $sk_{\mathcal{S}_1}, ..., sk_{\mathcal{S}_q}$.

Challenge: $\mathcal{A}$ chooses a message m and access structure (M, f) and then sends them to the challenger $\mathcal{C}$. $\mathcal{C}$ sets $CT = Enc(m, (M, f))$ and sends it back to $\mathcal{A}$.

Query: This phase is the same as the previous query.

Output: The adversary $\mathcal{A}$ outputs a revoked ciphertext $CT^{'}$ and an attribute set $Att^{'}$. The adversary wins if $Dec_{re}(sk_{\mathcal{S}'}, CT, CT^{'}) \notin \{m, \perp\}$. The adversary $\mathcal{A}$'s advantage to win the integrity game is defined as $Pr[\mathcal{A}wins]$.

IV. CONSTRUCTION

This section first introduces a modified FAME construction by adding a commitment to the ciphertext. We then show that the modified FAME is fully secure. Furthermore, we proposed the revocable FAME scheme with data integrity.

A. FAME scheme with data integrity

The FAME scheme with data integrity contains four algorithms. The scheme is described in Fig.3.

Correctness. Here we show that if the attribute set contained in the private key sk satisfies the access structure $\mathbb{A}$ in a valid FAME ciphertext, then our decryption algorithm will recover the correct message with probability 1. The calculation process is described in Fig.4.

B. RFAME-DI scheme

The RFAME-DI scheme is a further improvement on the FAME scheme with data integrity. The Setup, KeyGen, $\mathrm{Encrypt}$, $\mathrm{Decrypt}_{\mathrm{or}}$ of our RFAME-DI scheme is the same as the scheme above. To achieve revocation, we add three algorithms $\mathrm{Delegate}$, Revoke and $\mathrm{Decrypt}_{\mathrm{re}}$ as shown in the Fig.5.

Correctness. The decryption algorithm $\mathrm{Decrypt}_{\mathrm{re}}$ is the same as the original decryption algorithm $\mathrm{Decrypt}_{\mathrm{or}}$ except

Setup(1^λ): Let $(p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e, g, h)$ be a prime order bilinear group system. Authority center selects a_1, a_2 randomly from $\mathbb{Z}_p^*$ and selects d_1, d_2, d_3 randomly from $\mathbb{Z}_p$ and computes$H_1 = h^{a_1}, H_2 = h^{a_2}, T_1 = e(g,h)^{d_1 a_1 + d_3}, T_2 = e(g,h)^{d_2 a_2 + d_3}$. Authority center also chooses two hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{G}$ and $\mathrm{H} : \mathbb{G}_T \to \mathbb{Z}_p^*$ and selects φ, ϕ randomly from $\mathbb{G}$, then authority center output public parameters $PP = (g, h, H_1, H_2, T_1, T_2, \varphi, \phi, \mathcal{H}, \mathrm{H})$. Authority center selects b_1, b_2 randomly from $\mathbb{Z}_p^*$ and output master secret key $msk = (a_1, a_2, b_1, b_2, g^{d_1}, g^{d_2}, g^{d_3})$.

KeyGen(msk, S): Authority center selects r_1, r_2 randomly from $\mathbb{Z}_p$, and uses b_1, b_2 to compute $sk_0 = (h^{b_1 r_1}, h^{b_2 r_2}, h^{r_1 + r_2})$ We use $sk_{0,1}, sk_{0,2}, sk_{0,3}$ to denote the first, second and third elements of sk_0. Authority center selects $\sigma_y, \sigma^{'}$ randomly from $\mathbb{Z}_p$, for all $y \in \mathcal{S}$ and $z = 1, 2$ computes

$$sk_{y,z} = \mathcal{H}(y1z)^{\frac{b_1 r_1}{a_z}} \cdot \mathcal{H}(y2z)^{\frac{b_2 r_2}{a_z}} \cdot \mathcal{H}(y3z)^{\frac{r_1 + r_2}{a_z}} \cdot g^{\frac{r_1 + r_2}{a_z}},$$
$$sk_z^{'} = g^{d_z} \cdot \mathcal{H}(011z)^{\frac{b_1 r_1}{a_z}} \cdot \mathcal{H}(012z)^{\frac{b_2 r_2}{a_z}} \cdot \mathcal{H}(013z)^{\frac{r_1 + r_2}{a_z}} \cdot g^{\frac{\sigma^{'}}{a_z}}.$$

Authority center let $sk_y = (sk_{y,1}, sk_{y,2}, g^{-\sigma_y})$, $sk^{'} = (sk_1^{'}, sk_2^{'}, g^{d_3 - \sigma^{'}})$ and output secret key $sk = (\mathcal{S}, sk_0, \{sk_y\}_{y \in \mathcal{S}}, sk^{'})$ for attribute set $\mathcal{S}$.

Encrypt($m, \mathbb{A} = (M, f)$): Data owner selects s_1, s_2 and $\hat{s_1}, \hat{s_2}$ from $\mathbb{Z}_p$ randomly, computes $c_0 = (H_1^{s_1}, H_2^{s_2}, h^{s_1 + s_2})$ and $d_0 = (H_1^{\hat{s_1}}, H_2^{\hat{s_2}}, h^{\hat{s_1} + \hat{s_2}})$. We use $c_{0,1}, c_{0,2}, c_{0,3}$ to denote the first, second and third elements of c_0, the same for d_0. Suppose that M is a matrix of $n_1 \times n_2$ and $(M)_{i,j}$ is the element in the i-th row and j-th column of matrix M. For $i = 1, ..., n_1$ and $l = 1, 2, 3$ computes

$$c_{i,l} = \mathcal{H}(f(i)l1)^{s_1} \cdot \mathcal{H}(f(i)l2)^{s_2} \cdot \textstyle\prod_{j=1}^{n_2} [\mathcal{H}(0jl1)^{s_1} \cdot \mathcal{H}(0jl2)^{s_2}]^{(M)_{i,j}},$$

Data owner lets $c_i = (c_{i,1}, c_{i,2}, c_{i,3})$ and computes $ct^{'} = T_1^{s_1} \cdot T_2^{s_2} \cdot m$, where message m belongs to group $\mathbb{G}_T$. Then data owner selects $m^{'}$ from $\mathbb{G}_T$ randomly and for $i = 1, ..., n_1$ and $l = 1, 2, 3$ computes

$$d_{i,l} = \mathcal{H}(f(i)l1)^{\hat{s_1}} \cdot \mathcal{H}(f(i)l2)^{\hat{s_2}} \cdot \textstyle\prod_{j=1}^{n_2} [\mathcal{H}(0jl1)^{\hat{s_1}} \cdot \mathcal{H}(0jl2)^{\hat{s_2}}]^{(M)_{i,j}},$$

Data owner lets $d_i = (d_{i,1}, d_{i,2}, d_{i,3})$ and computes $ct^{''} = T_1^{\hat{s_1}} \cdot T_2^{\hat{s_2}} \cdot m^{'}$, $csum = \varphi^{\mathrm{H}(m)} \phi^{\mathrm{H}(m')}$.

Data owner outputs ciphertext $CT = (\mathbb{A}, c_0, c_1, ..., c_{n_1}, ct^{'}, d_0, d_1, ..., d_{n_1}, ct^{''}, csum)$.

Decrypt$_{\mathrm{or}}$(sk, CT): Data user checks whether the attribute set $\mathcal{S}$ contained in the private key satisfies the access structure $\mathbb{A} = (M, f)$ in the ciphertext. If not, the algorithm outputs an error symbol $\perp$ and aborts. Otherwise, data user perform the following operations. Data owner finds the set $I \subset \{1, 2, ..., n_1\}$ and $I = \{j : f(j) \in \mathcal{S}\}$ then there exists a constant set $\{\theta_i\}_{i \in I}$ that satisfy $\sum_{i \in I} \theta_i \cdot M_i = (1, 0, ..., 0)$. Data owner computes

$$num = ct^{'} \cdot e(\textstyle\prod_{i \in I} c_{i,1}^{\theta_i}, sk_{0,1}) \cdot e(\prod_{i \in I} c_{i,2}^{\theta_i}, sk_{0,2}) \cdot e(\prod_{i \in I} c_{i,3}^{\theta_i}, sk_{0,3}),$$

$$den = e(sk_1^{'} \cdot \textstyle\prod_{i \in I} sk_{f(i),1}^{\theta_i}, ct_{0,1}) \cdot e(sk_2^{'} \cdot \prod_{i \in I} sk_{f(i),2}^{\theta_i}, ct_{0,2}) \cdot e(sk_3^{'} \cdot \prod_{i \in I} sk_{f(i),3}^{\theta_i}, ct_{0,3}),$$

From the $ct^{'}$ we know that in order to recover the message m, the essence is that we need to compute $T_1^{s_1} \cdot T_2^{s_2}$. For $l = 1, 2, 3$, using the equation $\sum_{i \in I} \theta_i (M)_i = (1, 0, 0, ..., 0)$, we can calculate $\prod_{i \in I} c_{i,l}^{\theta_i}$.

$$\begin{aligned}\prod_{i \in I} c_{i,l}^{\theta_i} &= \prod_{i \in I} (\mathcal{H}(f(i)l1)^{\theta_i s_1} \cdot \mathcal{H}(f(i)l2)^{\theta_i s_2} \cdot \prod_{j=1}^{n_2} [\mathcal{H}(0jl1)^{s_1} \cdot \mathcal{H}(0jl2)^{s_2}]^{\theta_i (M)_{i,j}}) \\ &= \prod_{j=1}^{n_2} [\mathcal{H}(0jl1)^{s_1} \cdot \mathcal{H}(0jl2)^{s_2}]^{\sum_{i \in I} \theta_i (M)_{i,j}} \cdot (\prod_{i \in I} (\mathcal{H}(f(i)l1)^{\theta_i s_1} \cdot \mathcal{H}(f(i)l2)^{\theta_i s_2}) \\ &= [\mathcal{H}(01l1)^{s_1} \cdot \mathcal{H}(01l2)^{s_2}]^{\sum_{i \in I} \theta_i (M)_{i,j}} \cdot (\prod_{i \in I} (\mathcal{H}(f(i)l1)^{\theta_i s_1} \cdot \mathcal{H}(f(i)l2)^{\theta_i s_2})\end{aligned}$$

This allows us to compute the last three pairing operations in num except for the $ct^{'}$ involving the encrypted message m. The results are as follows.

$$num/ct^{'} = \prod_{t \in \{1,2\}} [e(\mathcal{H}(011t), h)^{b_1 r_1 s_t} \cdot e(\mathcal{H}(012t), h)^{b_2 r_2 s_t} \cdot e(\mathcal{H}(013t), h)^{(r_1 + r_2) s_t} \cdot \prod_{i \in I} (e(\mathcal{H}(f(i)1t)^{\theta_i}, h)^{b_1 r_1 s_t} \cdot e(\mathcal{H}(f(i)2t)^{\theta_i}, h)^{b_2 r_2 s_t} \cdot e(\mathcal{H}(f(i)3t)^{\theta_i}, h)^{(r_1 + r_2) s_t})]$$

When the above product $num/ct^{'}$ and the related term in dem are eliminated, we get the inverse of

$$(\prod_{t \in \{1,2\}} e(g^{d_t} \cdot g^{\frac{\sigma^{'}}{a_t}} \cdot \prod_{i \in I} g^{\frac{\theta_i \sigma_{f(i)}}{a_t}}, h^{a_t s_t})) \cdot e(g^{d_3} \cdot g^{-\sigma^{'}} \cdot \prod_{i \in I} g^{-\theta_i \sigma_{f(i)}}, h^{s_1 + s_2})$$

Simply organizing the above term we get $e(g, h)^{d_1 a_1 s_1 + d_2 a_2 s_2 + d_3 (s_1 + s_2)}$, it is easy to see that this is the $T_1^{s_1} \cdot T_2^{s_2}$ we want. Hence, the message m is recovered successfully. The same reason for $m^{'}$.

Fig. 4. Correctness of scheme

for the first step, the fourth part of the original ciphertext CT_{csum} is compared with the checksum in the revoked ciphertext $CT^{'}$. Therefore, as long as $CT^{'}$ is a valid ciphertext under the access structure $(M^{'}, f^{'})$, algorithm $\mathrm{Decrypt_{re}}$ naturally satisfies the correctness. In the following, we show that the CP-ABE ciphertext $CT^{'}$ generated through Revoke algorithm is valid by lemma 1.

Lemma 1. *If the above M and $\widetilde{M}$ are valid LSSS access structures, then $M^{'}$ is a valid LSSS access structure, and vice versa.*

proof: Since (M, f) and $(\widetilde{M}, \widetilde{f})$ are valid access structures, there exist two sets of constants $\{\theta_i\}_{i \in [1, n_1]}$ and $\{\widetilde{\theta}_i\}_{i \in [1, \widetilde{n_1}]}$ which elements both belong to $\mathbb{Z}_p^*$, such that $\sum_{i \in [1, n_1]} \theta_i \cdot M_i = (1, 0, 0, ..., 0) \in \mathbb{Z}_p^{n_1}$ and $\sum_{i \in [1, \widetilde{n_1}]} \widetilde{\theta}_i \cdot \widetilde{M}_i = (1, 0, 0, ..., 0) \in \mathbb{Z}_p^{\widetilde{n_1}}$, respectively. We can then use $\{\theta_i\}_{i \in [1, n_1]}$ and $\{\widetilde{\theta}_i\}_{i \in [1, \widetilde{n_1}]}$ to construct $\{\theta_i^{'}\}_{i \in [1, n_1^{'}]}$ in the following way, where $n_1^{'} = n_1 + \widetilde{n_1}$.

$$\theta_i^{'} = \begin{cases} \theta_i & , i \in [1, n_1] \\ \widetilde{\theta}_{i - n_1} & , i \in [n_1 + 1, n_1^{'}] \end{cases}$$

It is easy to deduce from the formula that,

$$\begin{aligned}\sum_{i \in [1, n_1^{'}]} \theta_i^{'} \cdot M_i^{'} &= \sum_{i \in [1, n_1]} \theta_i \cdot M_i^{'} + \sum_{i \in [1, \widetilde{n_1}]} \widetilde{\theta}_i \cdot M_{i + n_1}^{'} \\ &= (\overbrace{1, 0, ..., 0}^{n_1}, \overbrace{-1, 0, ..., 0}^{\widetilde{n_1}}) + (\overbrace{0, ..., 0}^{n_1}, \overbrace{1, 0, ..., 0}^{\widetilde{n_1}}) \\ &= (1, 0, 0, ..., 0) \in \mathbb{Z}_p^{n_1^{'}}\end{aligned}$$

Thus, $(M^{'}, f^{'})$ is valid LSSS access structures.

Conversely, if $(M^{'}, f^{'})$ is a valid access structure, there exists a set of constants $\{\theta_i^{'}\}_{i \in [1, n_1^{'}]}$ which elements both belong to $\mathbb{Z}_p^*$, such that $\sum_{i \in [1, n_1^{'}]} \theta_i^{'} \cdot M_i^{'} = (1, 0, 0, ..., 0) \in \mathbb{Z}_p^{n_1^{'}}$. We can construct $\{\theta_i = \theta_i^{'}\}_{i \in [1, n_1]}$ and $\{\widetilde{\theta}_i = \theta_{i + n_1}\}_{i \in [1, \widetilde{n_1}]}$

$$\begin{aligned}\sum_{i \in [1, n_1^{'}]} \theta_i^{'} \cdot M_i^{'} &= (1, 0, 0, ..., 0) \\ &= (\overbrace{1, 0, ..., 0}^{n_1}, \overbrace{-1, 0, ..., 0}^{\widetilde{n_1}}) + (\overbrace{0, ..., 0}^{n_1}, \overbrace{1, 0, ..., 0}^{\widetilde{n_1}}) \\ &= \sum_{i \in [1, n_1]} \theta_i \cdot M_i^{'} + \sum_{i \in [1, \widetilde{n_1}]} \widetilde{\theta}_i \cdot M_{i + n_1}^{'}\end{aligned}$$

Delegate($\widetilde{\mathbb{A}}$): The data owner specifies an access structure $\widetilde{\mathbb{A}} = (\widetilde{M}, \widetilde{f})$, where $\widetilde{M}$ is a $\widetilde{n_1} \times \widetilde{n_2}$ matrix and $\widetilde{f}$ is a function that maps the row of matrix $\widetilde{M}$ to a attribute. Then the following calculation is done for the new attributes contained in the access structure $\widetilde{\mathbb{A}}$ and $l = 1, 2, 3$.

$$dt_l = \prod_{i=1}^{\widetilde{n_1}} \mathcal{H}(\widetilde{f}(i)l1)^{s_1} \cdot \mathcal{H}(\widetilde{f}(i)l1)^{s_2}$$

Data owner lets $dt = (dt_1, dt_2, dt_3)$ and outputs delegation $DG = (dt, \widetilde{\mathbb{A}} = (\widetilde{M}, \widetilde{f}))$ for the could server.

Revoke(CT, DG): The cloud server inputs a ciphertext $CT = (\mathbb{A}, c_0, c_1, ..., c_{n_1}, ct^{'}, e_0, e_1, ..., e_{n_1}, ct^{''}, csum)$ and a delegation $DG = (dt, \widetilde{\mathbb{A}} = (\widetilde{M}, \widetilde{f}))$. M and $\widetilde{M}$ are $n_1 \times n_2$ and $\widetilde{n_1} \times \widetilde{n_2}$ matrix, respectively. The cloud server outputs a revoked ciphertext for access structure $\mathbb{A}^{'} = (M^{'}, f^{'})$. The cloud server sets $(M^{'}, f^{'})$ as

$$M^{'} = \left(\begin{array}{c|c|c} M & -col_1 & \mathbf{0} \\ \hline \mathbf{0} & & \widetilde{M} \end{array}\right), \ f^{'} = \begin{cases} f(j) & j \le n_1 \\ \widetilde{f}(j - n_1) & j > n_1 \end{cases}$$

where col_1 is first column of M. $M^{'}$ is $n_1^{'} \times n_2^{'}$ matrix , where $n_1^{'} = n_1 + \widetilde{n_1}, n_2^{'} = n_2 + \widetilde{n_2}$. The cloud server selects $s_1^{'}, s_2^{'}$ from $\mathbb{Z}_p$ randomly and computes revoked ciphertext

$$\bar{ct^{'}} = ct^{'} \cdot T_1^{s_1^{'}} \cdot T_2^{s_2^{'}}, \ \bar{c_0} = (c_{0,1} \cdot H_1^{s_1^{'}}, c_{0,2} \cdot H_2^{s_2^{'}}, c_{0,3} \cdot h^{s_1^{'} + s_2^{'}}),$$

$$\begin{cases} \bar{c_{i,l}} = c_{i,l} \cdot \mathcal{H}(f(i)l1)^{s_1^{'}} \cdot \mathcal{H}(f(i)l2)^{s_2^{'}} \cdot \prod_{j=1}^{n_2^{'}} [\mathcal{H}(0jl1)^{s_1^{'}} \cdot \mathcal{H}(0jl2)^{s_2^{'}}]^{(M^{'})_{i,j}} & i \in [1, n_1] \\ \bar{c_{i,l}} = dt_l \cdot \mathcal{H}(f(i)l1)^{s_1^{'}} \cdot \mathcal{H}(f(i)l2)^{s_2^{'}} \cdot \prod_{j=1}^{n_2^{'}} [\mathcal{H}(0jl1)^{s_1^{'}} \cdot \mathcal{H}(0jl2)^{s_2^{'}}]^{(M^{'})_{i,j}} & i =\in [n_1 + 1, n_1^{'}] \end{cases}.$$

The cloud server lets $\bar{c_i} = (\bar{c}_{i,1}, \bar{c}_{i,2}, \bar{c}_{i,3})$ for i from 1 to $n_1^{'}$ and $\overline{csum} = csum$. The $\bar{d_0}, \bar{d_1}, ..., \bar{d}_{n_1}$ can be get in the same way as above.

The revoked ciphertext is $CT^{'} = (\mathbb{A}^{'}, \bar{c}_0, \bar{c}_1, ..., \bar{c}_{n_1^{'}}, \bar{ct^{'}}, \bar{d}_0, \bar{d}_1, ..., \bar{d}_{n_1^{'}}, \bar{ct^{''}}, \overline{csum})$.

$\mathrm{Decrypt_{re}}(sk^{'}, CT_{csum}, CT^{'})$: The data user input the private key sk' corresponding to the attribute set $\mathcal{S}'$, the checksum in original ciphertext CT_{csum} and revoked ciphertext $CT^{'}$. First, the data user verify whether $\overline{csum}$ is equal to CT_{csum}. If not, the data user outputs an error symbol $\perp$ and abort. The second step, the data user checks whether the attribute set $\mathcal{S}'$ satisfies the access structure $(M^{'}, f^{'})$. If it doesn't satisfy, the data user outputs an error symbol $\perp$ and abort. Otherwise, the data user finds a set $I^{'} \subset \{1, 2, ..., n_1^{'}\}$ where $I^{'} = \{j : f^{'}(j) \in \mathcal{S}'\}$ and a set of constants $\{\theta_i^{'}\}_{i \in I'}$ which elements belong to $\mathbb{Z}_p^*$. Such that $\sum_{i \in I'} \theta_i^{'} \cdot M_i^{'} = (1, 0, 0, ..., 0)$. Then, the data user calculates m and $m^{'}$ in the same way as in $\mathrm{Decrypt_{or}}$. At last, the data user checks whether $\overline{csum} = \varphi^{\mathrm{H}(m)} \phi^{\mathrm{H}(m')}$. If the equation holds, the data user outputs m, otherwise outputs an error symbol $\perp$.

Fig. 5. RFAME-DI scheme

$$= (\sum_{i\in[1,n_1]} \theta_i \cdot M_i, \sum_{i\in[1,n_1]} \theta_i \cdot (\overbrace{-col_{1,i}, 0, ..., 0}^{\widetilde{n_1}}))$$

$$+ (\sum_{i\in[1,\widetilde{n_1}]} \widetilde{\theta}_i \cdot \overbrace{(0, ..., 0)}^{n_1}, \sum_{i\in[1,\widetilde{n_1}]} \widetilde{\theta}_i \cdot \widetilde{M}_i)$$

From the above equation it can be inferred that $\sum_{i\in[1,n_1]} \theta_i \cdot M_i = (1, 0, 0, ..., 0) \in \mathbb{Z}_p^{n_1}$ and $\sum_{i\in[1,\widetilde{n_1}]} \widetilde{\theta}_i \cdot \widetilde{M}_i = (1, 0, 0, ..., 0) \in \mathbb{Z}_p^{\widetilde{n_1}}$ .Thus, (M, f) and $(\widetilde{M}, \widetilde{f})$ are valid LSSS access structures.

V. SECURITY PROOFS

In this section, we first prove the semantic security of the modified FAME scheme by reducing it to the original FAME scheme and then prove the semantic security of the proposed revocable FAME with data integrity scheme. Finally, we give a data integrity proof of our formal scheme by reducing to the discrete logarithm problem.

Theorem 1. *The above modified FAME scheme is fully IND-CPA secure if Shashank's FAME scheme [9] is fully IND-CPA secure.*

Proof: If there exists an adversary $\mathcal{A}$ to break the full security of our scheme, then a simulator $\mathcal{B}$ can be constructed which can break the full security of the underlying FAME scheme by interacting with the challenger $\mathcal{C}$.

• Setup. Simulator $\mathcal{B}$ calls $\mathcal{C}$ to get the basic parameters $(p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e, H_1 = h^{a_1}, H_2 = h^{a_2}, T_1 = e(g,h)^{d_1a_1+d_3}, T_2 = e(g,h)^{d_2a_2+d_3}, \mathcal{H})$. $\mathcal{B}$ randomly chooses $\phi, \varphi \in \mathbb{G}$ and hash function H. $\mathcal{B}$ returns the public parameters $PP = (H_1, H_2, T_1, T_2, \mathcal{H}, \phi, \varphi, \mathrm{H})$ to $\mathcal{A}$.

• Query. $\mathcal{A}$ issues a private key query on attribute set $\mathcal{S}$, $\mathcal{B}$ forwards this query to $\mathcal{C}$ and then forwards the result returned by $\mathcal{C}$ to $\mathcal{A}$. $\mathcal{A}$ can also initiate a delegation query to access structure $\widetilde{\mathbb{A}} = (\widetilde{M}, \widetilde{f})$. $\mathcal{B}$ randomly selects two random numbers s_a and s_b and computes $dt_l = \prod_{i=1}^{\widetilde{n_1}} \mathcal{H}(\widetilde{f}(i)l1)^{s_a} \cdot \mathcal{H}(\widetilde{f}(i)l1)^{s_b}$, set $dt = (dt_1, dt_2, dt_3)$, then $\mathcal{B}$ forwards dt to $\mathcal{A}$.

• Challenge. $\mathcal{A}$ submits two equal length plaintext m_0, m_1 and the access structure $\mathbb{A}^* = (M^*, f^*)$ intended to be challenged to $\mathcal{B}$. $\mathcal{B}$ forwards m_0, m_1 and $\mathbb{A}^*$ to challenger $\mathcal{C}$. $\mathcal{C}$ selects a random value $\sigma \in \{0, 1\}$ and returns the challenge ciphertext of m_σ encrypted under access structure $\mathbb{A}^*$ to $\mathcal{B}$, denote as $(\mathbb{A}^*, c_0, c_1, ..., c_{n_1}, ct^{'})$. The simulator $\mathcal{B}$ chooses $m^{'} \in \mathbb{G}, Q \in \mathbb{G}_T$ and $\hat{s_1}, \hat{s_2}$ from $\mathbb{Z}_p$ randomly. The simulator $\mathcal{B}$ computes $d_0 = (H_1^{\hat{s_1}}, H_2^{\hat{s_2}}, h^{\hat{s_1}+\hat{s_2}})$ and for $i = 1, ..., n_1$ and $l = 1, 2, 3$ $\mathcal{B}$ computes

$d_{i,l} = \mathcal{H}(f(i)l1)^{\hat{s_1}} \cdot \mathcal{H}(f(i)l2)^{\hat{s_2}} \cdot \prod_{j=1}^{n_2}[\mathcal{H}(0jl1)^{\hat{s_1}} \cdot \mathcal{H}(0jl2)^{\hat{s_2}}]^{(M^*)_{i,j}}$. $\mathcal{B}$ sets $d_i = (d_{i,1}, d_{i,2}, d_{i,3})$. Then $\mathcal{B}$ computes $ct^{''} = T_1^{\hat{s_1}} \cdot T_2^{\hat{s_2}} \cdot m^{'}$ and sets $csum = Q$. Eventually $\mathcal{B}$ sets ciphertext $CT = ((M^*, f^*), c_0, c_1, ..., c_{n_1}, ct^{'}, d_0, d_1, ..., d_{n_1}, ct^{''}, csum)$ and forwards it to $\mathcal{A}$ as the challenge ciphertext.

• Query. This phase is the same as the previous query.

• Output. When the adversary $\mathcal{A}$ outputs its guess $\sigma^{'}$, $\mathcal{B}$ also takes $\sigma^{'}$ as its guess.

Analysis. If $\mathcal{A}$ can break our modified scheme, $\mathcal{B}$ can break the underlying FAME scheme with the same advantage. The simulation in the above game works perfectly, except that in the challenge phase $\mathcal{B}$ returns the random value Q in the group $\mathbb{G}_T$ instead of the computed check value $csum = \varphi^{\mathrm{H}(m)}\phi^{\mathrm{H}(m^{'})}$. Since adversary $\mathcal{A}$ does not know the value of $m^{'}$, the random value Q has the same distribution as the computed check value $csum$. This means that the computed $csum$ and the random value Q look the same from $\mathcal{A}$'s perspective.

Theorem 2. *The revocable FAME with data integrity scheme is fully IND-CPA secure if the modified FAME scheme is fully IND-CPA secure.*

Proof: First, in the revocable FAME with data integrity scheme, the ciphertext generated by encryption algorithm $\mathrm{Encrypt}$ before revocation (here called the original ciphertext) is the same as the ciphertext in the modified FAME scheme. So the original ciphertext generated by encryption in RFAME-DI achieves the same full security as the modified FAME scheme. Second, we will show that the revoked ciphertext generated by the revocation algorithm Revoke and the ciphertext generated by encryption are identically distributed, i.e., these two are indistinguishable from the adversary's view. In our RFAME-DI scheme, the ciphertext generated by the Revoke is the elements in the $\mathbb{G}_T$ generated by the value $s_1 + s_1^{'}$ and $s_2 + s_2^{'}$. Where $s_1^{'}, s_2^{'}$ are chosen randomly and s_1, s_2 are in the original ciphertext. So in the adversary's view $s_1 + s_1^{'}$ and $s_2 + s_2^{'}$ are also chosen randomly. Thus the revocation ciphertext generated with random value $s_1 + s_1^{'}, s_2 + s_2^{'}$ and the ciphertext generated by directly encrypting under the access structure $(M^{'}, f^{'})$ with randomly selected value $s_1^{''}, s_2^{''}$ are identically distributed, i.e., indistinguishable from the adversary's viewpoint.

Theorem 3. *If the discrete logarithm assumption holds, the revocable FMAE scheme captures the data integrity.*

Proof: If there exists an adversary $\mathcal{A}$ to break the integrity of the RFAME-DI scheme, then a simulator $\mathcal{B}$ can be constructed to solve the discrete logarithm problem. The input of simulator $\mathcal{B}$ is an instance of the discrete logarithm $(e, \mathbb{G}, \mathbb{G}_T, p, g, g^\delta)$, and its target is to output the value δ.

• Setup. Simulator $\mathcal{B}$ sets a prime order bilinear group system $(e, \mathbb{G}, \mathbb{G}_T, g, p)$. $\mathcal{B}$ selects a_1, a_2, b_1, b_2 randomly from $\mathbb{Z}_p^*$ and select d_1, d_2, d_3, η randomly from $\mathbb{Z}_p$, then chooses two collusion resistant hash function $\mathcal{H} : \{0, 1\}^* \to \mathbb{G}_T$ and $\mathrm{H} : \mathbb{G}_T \to \mathbb{Z}_p^*$. $\mathcal{B}$ sets $\varphi = g^\delta, \phi = g^\eta$ and computes $H_1 = h^{a_1}, H_2 = h^{a_2}, T_1 = e(g,h)^{d_1a_1+d_3}, T_2 = e(g,h)^{d_2a_2+d_3}$. Then $\mathcal{B}$ returns the public parameters $PP = (H_1, H_2, T_1 = , T_2, \varphi, \phi, \mathcal{H}, \mathrm{H})$ to $\mathcal{A}$.

• Query. When $\mathcal{A}$ issues a private key query for an attribute set $\mathcal{S}$. Since $\mathcal{B}$ knows the mast secret key $msk = (g, h, a_1, a_2, b_1, b_2, g^{d_1}, g^{d_2}, g^{d_3})$, $\mathcal{B}$ can generate the private key $sk_\mathcal{S}$ and then returns it to the adversary $\mathcal{A}$.

• Challenge. $\mathcal{A}$ chooses a message m and access structure $\mathbb{A} = (M, f)$ and sends them to the challenger. Note that

here $\mathcal{B}$ is both the simulator and the challenger. The simulator $\mathcal{B}$ executes the $Encrypt(m,(M,f))$ algorithm to get $CT=(\mathbb{A},c_0,c_1,...,c_{n_1},ct^{'},d_0,d_1,...,d_{n_1},ct^{''},csum)$, where $csum=\varphi^{\mathrm{H}(m)}\phi^{\mathrm{H}(m^{'})}$ and $m^{'}$is randomly selected during the encryption process. $\mathcal{B}$ returns CT to the adversary $\mathcal{A}$.

• Query. This phase is the same as the previous query.

• Output. The adversary $\mathcal{A}$ outputs a revoked ciphertext under the revoked access structure $\overline{CT}^{'}=(\mathbb{A}^{'}=(M^{'},f^{'}),\bar{c}_0,\bar{c}_1,...,\bar{c}_{n_1^{'}},\bar{ct}^{'},\bar{d}_0,\bar{d}_1,...,\bar{d}_{n_1^{'}},\bar{ct}^{''},\overline{csum})$.

At this point, simulator $\mathcal{B}$ can choose a attribute set $\mathcal{S}^{'}$ that satisfy the revoked access structure $M^{'}$ and then use the master private key to generate the corresponding private key$sk_{\mathcal{S}'}$, which is then used to decrypt $\overline{CT}^{'}$ to obtain the encrypted message. Denote the decrypted real message as $\bar{m}$ and the random message as $\bar{m^{'}}$. If $\mathcal{A}$ can break the integrity, then it means $\bar{m}\neq m$ and $\bar{m^{'}}\neq m^{'}$ but $\overline{csum}=csum$. So that $\mathcal{B}$ can calculate δ by the following computation and return δ as its answer for discrete logarithm assumption.

$$
\begin{aligned}
csum=\overline{csum} &\Leftrightarrow \varphi^{\mathrm{H}(m)}\phi^{\mathrm{H}(m^{'})}=\varphi^{\mathrm{H}(\bar{m})}\phi^{\mathrm{H}(\bar{m^{'}})}\\
&\Leftrightarrow g^{\delta\cdot\mathrm{H}(m)+\eta\cdot\mathrm{H}(m^{'})}=g^{\delta\cdot\mathrm{H}(\bar{m})+\eta\cdot\mathrm{H}(\bar{m^{'}})}\\
&\Leftrightarrow \delta\cdot(\mathrm{H}(m)-\mathrm{H}(\bar{m}))=\eta\cdot(\mathrm{H}(\bar{m^{'}})-\mathrm{H}(m^{'}))
\end{aligned}
$$

$$\Rightarrow \delta=\frac{\eta\cdot(\mathrm{H}(\bar{m^{'}})-\mathrm{H}(m^{'}))}{\mathrm{H}(m)-\mathrm{H}(\bar{m})}$$

Analysis. The simulation of $\mathcal{B}$ in the above game is perfect, and the advantage of $\mathcal{B}$ in solving the discrete logarithmic hard problem is the same as the advantage of adversary $\mathcal{A}$ in winning the above integrity security game.

VI. PERFORMANCE AND EVALUATION

In this section, we implement our scheme and analyze its practicality in terms of computational cost, we also compare it with other RABE schemes.

To implement our scheme, we experimented on a laptop with AMD's CPU. It's Ryzon 7 5800H @3.2GHz with 4GB RAM. The operating system of the client and cloud server are both Linux Ubuntu 20.04. We use the library charm 0.5.0 [47] in python to write the code and run it with version 2.7.15 of python. It is possible to carry our scheme on type III curves, such as MNT and BN curves, but some MNT and BN curves are known to be insecure in some parameters [48], [49], so be sure to choose elliptic curves carefully when implementing cryptographic schemes. We implement our scheme on the type III curve MNT224 in PBC, which is recognized as an excellent curve in terms of security and efficiency.

Since there are AND-gates and OR-gates in the randomly generated access structure, the size of the access structure does not represent the number of attributes involved in decryption, and only some of the attributes in the access structure are needed to participate in the calculation to complete the decryption. Therefore, the access structure we generate for testing only contains AND-gates, so that the number of attributes involved in decryption is the same as the size of the access structure. We also did the same test 50 times and averaged the results to get more realistic and accurate time data. Specifically in the experiment, for the three algorithms $\mathrm{Encrypt}$, $\mathrm{Decrypt_{or}}$, and Revoke we set the size of the attribute set and access structure from 10 to 100 with a step of 10. For the Revoke algorithm, the size of the access structure is the size of the added access structure $(\widetilde{M},\widetilde{f})$. For the algorithm $\mathrm{Decrypt_{re}}$, the size of the access structure is from 20 to 200 with a step of 20. In fact, there is no difference between the two algorithms $\mathrm{Decrypt_{or}}$ and $\mathrm{Decrypt_{re}}$, but $\mathrm{Decrypt_{re}}$ does one more equation verification operation than $\mathrm{Decrypt_{or}}$, i.e., verifying whether the $csum$ item from the original ciphertext is equal to $csum$ in the revoked ciphertext, and this operation is very light and its time consumption is negligible. We can see from the Fig that the performance of the $\mathrm{Decrypt_{re}}$ algorithm is almost the same as that of the $\mathrm{Decrypt_{or}}$ algorithm in the set size range from 10 to 100.

We compare with the scheme [21]. Our scheme is inferior to the comparison scheme in the key generation phase, which takes 220ms to generate the key for the user with 50 attributes. In contrast, from Fig.6-9, we see that our scheme outperforms the the scheme [21] in all four phases: encryption, decryption, revocation, and decryption of the revoked ciphertext. Especially in decryption phases, where the execution time of our scheme for decryption is independent of the set size, which is stable at about 36ms regardless of the set size. This is because the decryption process of FAME only needs 6 pairing operations. Our scheme has acceptable disadvantages and very large advantages in practical applications. Because the key generation is performed by the key generation center, which does not lack computational power, and the algorithm is usually executed only once for one user. The encryption and decryption are performed by the client device which requires the computational task as lighter as possible and since ABE is a one-to-many cryptographic primitive, decryption is performed more frequently than encryption in practical scenarios, and our scheme happens to exactly has a non-negligible advantage in decryption.

The above experimental results show that our revocation scheme, which can guarantee data integrity, is efficient and practical.

VII. CONCLUSION

Some revocable attribute-based encryption schemes consider the problem of data integrity. Nevertheless, these schemes have some drawbacks in algorithmic efficiency. In this work, we propose revocable attribute-based encryption schemes that have efficient algorithmic execution efficiency while protecting the integrity of the data stored in the cloud. We have conducted simulation experiments and experimental analysis shows that our scheme is efficient in encryption, decryption and revocation algorithms. Thus our scheme is an efficient revocable attribute-based encryption scheme with data integrity.

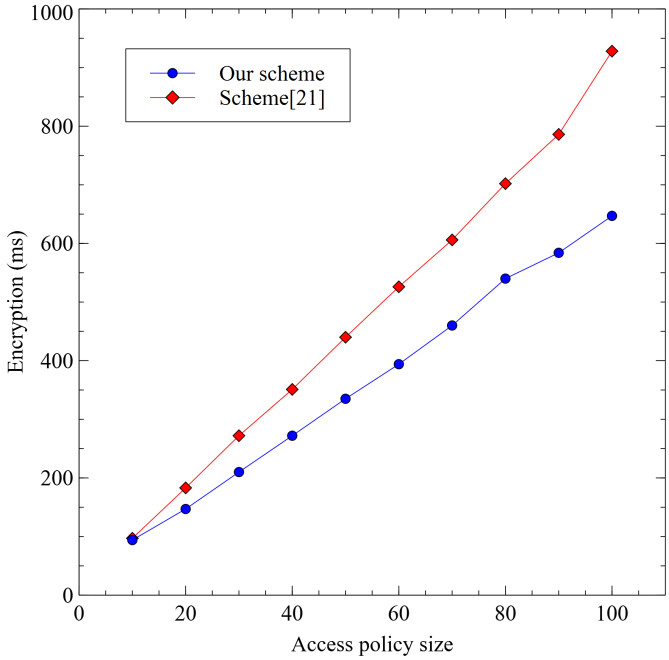

Fig. 6. Encryption time of RFAME-DI scheme and compared scheme

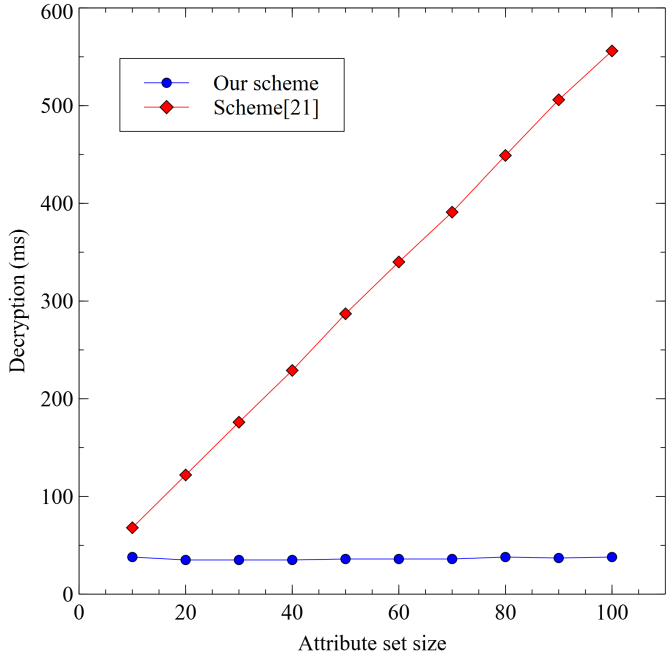

Fig. 7. Decryption time of RFAME-DI scheme and compared scheme

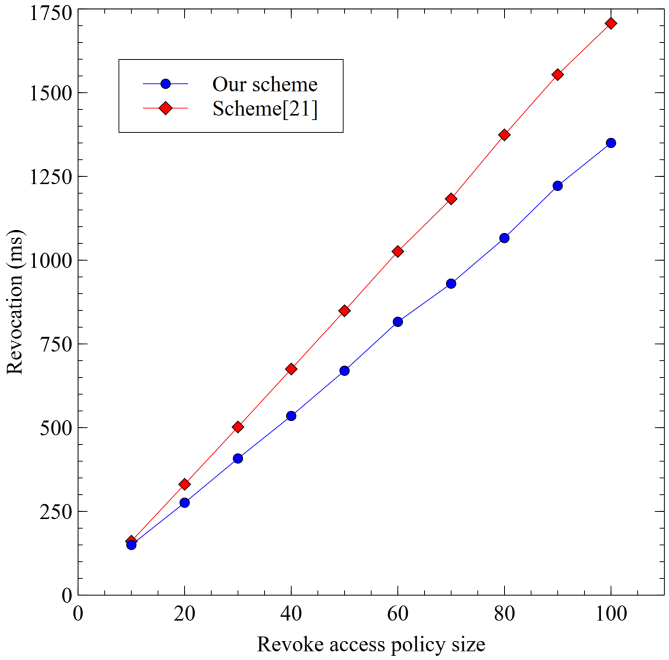

Fig. 8. Revocation time of RFAME-DI scheme and compared scheme

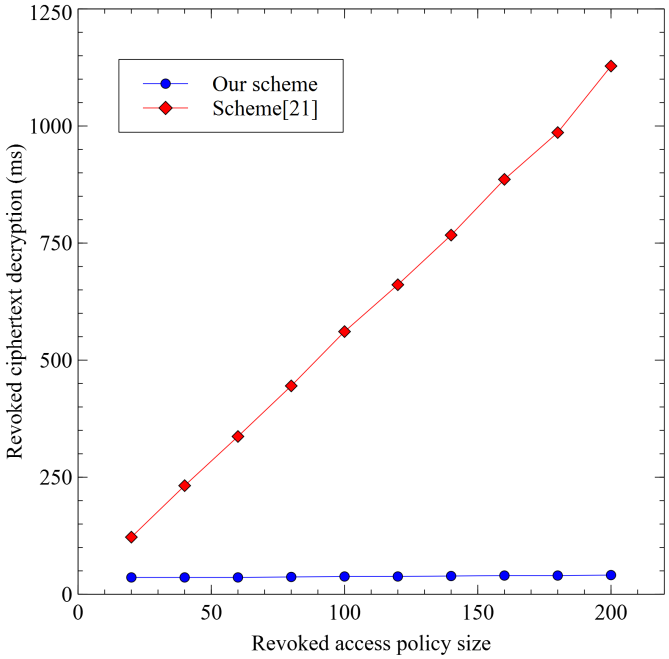

Fig. 9. Revoked ciphertext decryption time of RFAME-DI scheme and compared scheme

REFERENCES

[1] L. Zhang, H. Xiong, Q. Huang, J. Li, K. R. Choo, and J. Li, "Cryptographic solutions for cloud storage: Challenges and research opportunities," *IEEE Trans. Serv. Comput.*, vol. 15, no. 1, pp. 567–587, 2022.

[2] P. K. Premkamal, S. K. Pasupuleti, and P. J. A. Alphonse, "Attribute based encryption in cloud," *J. Netw. Comput. Appl.*, vol. 108, pp. 37–52, 2018.

[3] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Trans. Parallel Distributed Syst.*, vol. 24, no. 1, pp. 131–143, 2013.

[4] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," *Int. J. Inf. Sec.*, vol. 14, no. 6, pp. 487–497, 2015.

[5] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, vol. 3494. Springer, 2005, pp. 457–473.

[6] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in*

Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings, ser. Lecture Notes in Computer Science, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds., vol. 6571. Springer, 2011, pp. 53–70.

[7] K. B. Frikken, M. J. Atallah, and J. Li, "Attribute-based access control with hidden policies and hidden credentials," *IEEE Trans. Computers*, vol. 55, no. 10, pp. 1259–1270, 2006.

[8] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, vol. 7417. Springer, 2012, pp. 199–217.

[9] R. Zhang, J. Li, Y. Lu, J. Han, and Y. Zhang, "Key escrow-free attribute based encryption with user revocation," *Inf. Sci.*, vol. 600, pp. 59–72, 2022.

[10] J. Li, R. Zhang, Y. Lu, J. Han, Y. Zhang, W. Zhange, and X. Dong, "Multi-authority attribute-based encryption for assuring data deletion," *IEEE Systems Journal*, 2022.

[11] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang, "Attribute based encryption with privacy protection and accountability for cloudiot," *IEEE Trans. Cloud Comput.*, vol. 10, no. 2, pp. 762–773, 2022.

[12] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Trans. Parallel Distributed Syst.*, vol. 22, no. 7, pp. 1214–1221, 2011.

[13] J. Li, X. Huang, J. Li, X. Chen, and Y. Xiang, "Securely outsourcing attribute-based encryption with checkability," *IEEE Trans. Parallel Distributed Syst.*, vol. 25, no. 8, pp. 2201–2210, 2014.

[14] J. Li, Y. Wang, Y. Zhang, and J. Han, "Full verifiability for outsourced decryption in attribute based encryption," *IEEE Trans. Serv. Comput.*, vol. 13, no. 3, pp. 478–487, 2020.

[15] N. Chen, J. Li, Y. Zhang, and Y. Guo, "Efficient CP-ABE scheme with shared decryption in cloud storage," *IEEE Trans. Computers*, vol. 71, no. 1, pp. 175–184, 2022.

[16] S. Agrawal and M. Chase, "FAME: fast attribute-based message encryption," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*. ACM, 2017, pp. 665–682.

[17] D. Riepel and H. Wee, "FABEO: fast attribute-based encryption with optimal security," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, H. Yin, A. Stavrou, C. Cremers, and E. Shi, Eds. ACM, 2022, pp. 2491–2504.

[18] M. Ambrona, G. Barthe, R. Gay, and H. Wee, "Attribute-based encryption in the generic group model: Automated proofs and new constructions," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*. ACM, 2017, pp. 647–664.

[19] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*. ACM, 2007, pp. 195–203.

[20] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, vol. 5126. Springer, 2008, pp. 579–591.

[21] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "Revocable attribute-based encryption with data integrity in clouds," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 5, pp. 2864–2872, 2022.

[22] N. Attrapadung, B. Libert, and E. de Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, vol. 6571. Springer, 2011, pp. 90–108.

[23] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Trans. Parallel Distributed Syst.*, vol. 23, no. 11, pp. 2150–2162, 2012.

[24] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1767–1777, 2018.

[25] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Trans. Serv. Comput.*, vol. 10, no. 5, pp. 785–796, 2017.

[26] J. K. Liu, T. H. Yuen, P. Zhang, and K. Liang, "Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list," in *Applied Cryptography and Network Security - 16th International Conference, ACNS 2018, Leuven, Belgium, July 2-4, 2018, Proceedings*, vol. 10892. Springer, 2018, pp. 516–534.

[27] Y. Shi, Q. Zheng, J. Liu, and Z. Han, "Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation," *Inf. Sci.*, vol. 295, pp. 221–231, 2015.

[28] H. Wang, Z. Zheng, L. Wu, and P. Li, "New directly revocable attribute-based encryption scheme and its application in cloud storage environment," *Clust. Comput.*, vol. 20, no. 3, pp. 2385–2392, 2017.

[29] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, vol. 7417. Springer, 2012, pp. 199–217.

[30] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," *IEEE Trans. Parallel Distributed Syst.*, vol. 25, no. 7, pp. 1735–1744, 2014.

[31] S. Xu, G. Yang, and Y. Mu, "Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation," *Inf. Sci.*, vol. 479, pp. 116–134, 2019.

[32] Y. Yang, J. K. Liu, K. Liang, K. R. Choo, and J. Zhou, "Extended proxy-assisted approach: Achieving revocable fine-grained encryption of cloud data," in *Computer Security - ESORICS 2015 - 20th European Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceedings, Part II*, vol. 9327. Springer, 2015, pp. 146–166.

[33] H. Cui, R. H. Deng, Y. Li, and B. Qin, "Server-aided revocable attribute-based encryption," in *Computer Security - ESORICS 2016 - 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26-30, 2016, Proceedings, Part II*, vol. 9879. Springer, 2016, pp. 570–587.

[34] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008*. ACM, 2008, pp. 417–426.

[35] C. Liu, W. Hsien, C. C. Yang, and M. Hwang, "A survey of attribute-based access control with user revocation in cloud data storage," *Int. J. Netw. Secur.*, vol. 18, no. 5, pp. 900–916, 2016.

[36] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, ser. Lecture Notes in Computer Science, vol. 7417. Springer, 2012, pp. 199–217.

[37] Y. Jiang, W. Susilo, Y. Mu, and F. Guo, "Ciphertext-policy attribute-based encryption supporting access policy update and its extension with preserved attributes," *Int. J. Inf. Sec.*, vol. 17, no. 5, pp. 533–548, 2018.

[38] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 8, pp. 1343–1354, 2013.

[39] H. Deng, Z. Qin, Q. Wu, Z. Guan, and Y. Zhou, "Flexible attribute-based proxy re-encryption for efficient data sharing," *Inf. Sci.*, vol. 511, pp. 94–113, 2020.

[40] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2009, Sydney, Australia, March 10-12, 2009*. ACM, 2009, pp. 276–286.

[41] S. Luo, J. Hu, and Z. Chen, "Ciphertext policy attribute-based proxy re-encryption," in *Information and Communications Security - 12th International Conference, ICICS 2010, Barcelona, Spain, December 15-17, 2010. Proceedings*, vol. 6476. Springer, 2010, pp. 401–415.

[42] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 5, pp. 2907–2919, 2022.

[43] A. Sadeghi and M. Steiner, "Assumptions related to discrete logarithms: Why subtleties make a real difference," in *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, vol. 2045. Springer, 2001, pp. 244–261.

[44] J. A. Akinyele, M. Green, and A. D. Rubin, "Charm: A framework for rapidly prototyping cryptosystems," in *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5-8, 2012*. The Internet Society, 2012.

[45] Z. Liu and Z. Cao, "On efficiently transferring the linear secret-sharing scheme matrix in ciphertext-policy attribute-based encryption," *IACR Cryptol. ePrint Arch.*, p. 374, 2010.

[46] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*. ACM, 2013, pp. 463–474.

[47] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *J. Cryptogr. Eng.*, vol. 3, no. 2, pp. 111–128, 2013.

[48] A. Guillevic, F. Morain, and E. Thomé, "Solving discrete logarithms on a 170-bit MNT curve by pairing reduction," in *Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers*, vol. 10532. Springer, 2016, pp. 559–578.

[49] T. Kim and R. Barbulescu, "Extended tower number field sieve: A new complexity for the medium prime case," in *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, vol. 9814. Springer, 2016, pp. 543–571.