IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received June 7, 2019, accepted June 26, 2019, date of publication July 3, 2019, date of current version July 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2926627

Revocable Attribute-Based Encryption Scheme With Efficient Deduplication for Ehealth Systems

HUA MA[1], YING XIE[1], JIANFENG WANG[2], GUOHUA TIAN[1], AND ZHENHUA LIU[1]
[1]School of Mathematics and Statistics, Xidian University, Xi'an 710071, China
[2]State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xi'an 710071, China

Corresponding author: Ying Xie (chrissy525@126.com)

This work was supported in part by the National Natural Science Foundation of Shaanxi Province under Grant 2018JQ6001, in part by the Fundamental Research Funds for the Central Universities under Grant XJS17053, and in part by the China 111 Project under Grant B16037.

ABSTRACT The deduplication based on attribute-based encryption can be well used in eHealth systems to save storage space and share medical records. However, the excessive computation costs of existing schemes lead to inefficient deduplication. In addition, the frequent changes of clients' attribute weaken the forward secrecy of data, and thus, how to achieve the attribute revocation in deduplication is a problem that remains to be solved. In this paper, we propose a variant of the attribute-based encryption scheme that supports efficient deduplication and attributes revocation for eHealth systems. Specifically, an efficient deduplication protocol based on the nature of prime number is used to alleviate the computation burden on the private cloud, and attribute revocation is realized by updating the attribute agent key and the ciphertext. Moreover, outsourcing decryption is introduced to reduce the computation overhead of clients. The security analysis argues that the proposed scheme can reach the desired security requirements, and the visual experiment result indicates the excellent performance of the proposed scheme while realizing deduplication and attribute revocation.

INDEX TERMS Secure deduplication, attribute-based encryption, attribute revocation, eHealth systems.

I. INTRODUCTION

Cloud computing has brought about a tremendous revolution in various fields of society. More and more enterprises and individuals prefer to enjoy high-quality services through cloud computing, such as data storage, data sharing, and outsourcing computation. Due to the requirements for storage and continuous availability, cloud computing has been widely applied in the field of healthcare. The patient-centered eHealth systems [5] were proposed to maintain the clinical information on requirement basis, where each patient stores their medical records on the cloud server. Then they can selectively share their health information with someone who possesses the access privileges, which means that the patient may obtain many healthcare support. eHealth systems can play an important role in improving patient safety and health care quality. However, due to the ever-increasing volume of medical records and the data redundancy, the storage capacity and computation power of medical cloud are suffering a severe challenge. For example, the two patients are diagnosed with hypertension and stable angina pectoris, respectively, and then they may need the medicines ‘‘Metoprolol Tartrate tablets'', ‘‘Aspirin Enteric-ciated tablets'' and ‘‘Nifedipine sustained-release tablets'' with the same usage and dosage. Thus, many research on how to realize data sharing while saving storage space in eHealth systems are going on.

A promising countermeasure for saving storage space is to adopt deduplication in eHealth systems, which can identify the redundant physical copy of the same file according to data similarity, and delete the redundant copy. Then, all valid clients can access the single physical copy stored in the cloud via a link. To guarantee the privacy of medical records, patients encrypt them and upload the encrypted data to the medical cloud. Nevertheless, traditional encryption primitives are incompatible with deduplication since different patients may encrypt the identical data to generate distinct encrypted data with different keys. Bellare *et al.* [13] proposed a novel encryption method,

The associate editor coordinating the review of this manuscript and approving it for publication was Mansoor Ahmed.

named message-locked encryption (MLE), to realize deduplication over the encrypted data. MLE-based schemes require clients to encrypt their data with the hash value of object data, which ensures that different clients can obtain identical encrypted data of the same plaintext data.

The existing MLE-based deduplication schemes are not perfectly compatible with data sharing in eHealth systems. As one of the useful cryptographic primitives in multiple-party communication, attribute-based encryption (ABE) [22] can meet the service requirements of eHealth systems due to the fine-grained access control. In ABE, each client whose attribute set meets the access policy is allowed to download and decrypt the encrypted data. Cui *et al.* [4] designed a promising ABE-based deduplication scheme, which incorporates deduplication and data sharing perfectly. However, applying Cui *et al.*'s scheme to the medical system will create a security issue. Suppose a doctor gets promoted or leaves the system, his attributes should be changed correspondingly. If the doctor's access rights have not been revoked, he can access the encrypted data normally. Thus, how to realize attribute revocation in eHealth systems is worth to research.

Furthermore, most existing deduplication schemes use one-by-one retrieval to determine whether the uploaded data is duplicate, resulting in an inefficient physical copy search, as well as a tremendous waste of resources. Jiang *et al.* [19] proposed an efficient physical copy search method, named deduplication decision tree, which reduces the time complexity from the linear-level to the logarithm-level. How to further improve the efficiency of physical copy search in deduplication is attracting wide attention.

In response to the problems mentioned above, we propose a revocable ABE-based deduplication scheme for eHealth systems. The detailed contributions are enumerated as below:

1) We present a novel deduplication protocol by adopting an efficient physical copy search method, in which the private cloud can determine whether the outsourcing ciphertext is duplicate by executing only a division operation.
2) We propose an attribute-based encryption scheme supporting ciphertext deduplication and attribute revocation. Specifically, the private cloud re-encrypts the original ciphertext with a trapdoor key if the uploaded file is duplicate, and updates the attribute agent key and ciphertext after the client's attribute is revoked. Meanwhile, outsourcing decryption is introduced to reduce the task of the client.
3) The security analysis argues that our scheme is secure in the corresponding security model, and the visual simulation experiment demonstrates the excellent performance of the proposed scheme.

II. RELATED WORK

A. MLE-BASED DEDUPLICATION

With the explosive growth of data, encrypted data deduplication has attracted wide attention. Douceur *et al.* [9] introduced convergent encryption (CE) to implement the deduplication over encrypted data, in which the hash value of outsourced data is used as the encryption key. Thus, CE ensures that different clients can encrypt the same file to get the same ciphertext. Based on their work, Bellare *et al.* [13] presented the notion of message-locked encryption (MLE) and defined the corresponding security model. Then Abadi *et al.* [12] designed a randomized scheme to avoid the ciphertext derived from the message. In 2015, Bellare and Keelveedhi [14] also extended MLE and provided semantic security for messages, called interactive MLE (iMLE).

However, existing deduplication schemes suffer from one-by-one physical copy search or high-cost search algorithms, resulting in the high-latency and resource waste for deduplication. Jiang *et al.* [19] firstly proposed an efficient method for physical copy search, named deduplication decision tree, which associates the node of the decision tree with outsourced data, and the server can search the object data with logarithm-level computation cost rather than linear-level. Zhang *et al.* [26] firstly introduced a selective deduplication technique in eHealth systems, in which all similar object data are stored in the same department to improve the efficiency of copy search. Yang *et al.* [23] presented a cross-domain deduplication scheme that implements an efficient physical copy search by employing a B+ Tree.

B. ABE-BASED DEDUPLICAITON

Despite the compelling performances, MLE-based deduplication schemes cannot satisfy service demands since they are not compatible with data sharing.

For flexible data sharing, Sahai and Waters [22] presented a fuzzy identity-based encryption (FIBE) scheme that realizes a threshold access control. Besides, they also gave the definition of attribute-based encryption (ABE). Following the core idea of their work, Goyal *et al.* [20] constructed the first key-policy attribute-based encryption (KP-ABE), in which the private key of the client implies the information of the access structure, and the descriptive attributes set is inserted into ciphertext. To improve the flexibility of access control, Bethencourt *et al.* [6] presented another construction, named ciphertext-policy attribute-based encryption (CP-ABE), which inserts the access structure into the ciphertext, and the client's private key implies the information of the descriptive attributes set. Clients can acquire the plaintext only if their attribute set meets the access structure.

In recent years, an enormous amount of research has been done with respect to the practical application of ABE. Narayan *et al.* [15] adopted CP-ABE into health medical record system to achieve high-quality medical resource sharing and cross-regional optimization. Zhang *et al.* [31] introduced a privacy-aware s-health access control system, in which a large universe CP-ABE with access policies partially hidden is proposed. However, one issue that ABE schemes exist is that both the ciphertext size and time cost for decryption grow with the size of the access policy. To circumvent the obstacle, Green *et al.* [32] proposed a new method for ABE that largely eliminates the overhead of clients.

Zhang *et al.* [25] proposed an efficient and privacy-aware attribute-based encryption scheme that supports offline key generation and offline encryption to reduce the resource-limited clients' computation burden. To implement optimized data storage in ABE scheme, Cui *et al.* [4] constructed the first ABE-based deduplication scheme, which realizes flexible data sharing that is compatible with deduplication. Specifically, when the object data is uploaded again, the cloud acts as an agent to re-encrypt the original ciphertext under the new access policy. Zhou *et al.* [27] put forward a similarity-aware deduplication scheme with flexible access control to resist the brute-force attack while guaranteeing efficient deduplication.

C. REVOCATION

Boldyreva *et al.* [1] defined a primitive, named revocable identify-based encryption (RIBE), and constructed more efficient revocation scheme by integrating the fuzzy IBE (FIBE) scheme of [22] with the complete subtree (CS) scheme of [37]. The size of key updates is reduced from linear to logarithmic in Boldyreva *et al.*'s scheme. In 2014, Seo and Emura [33] showed that the previous revocation methods in identity-based encryption (IBE) are vulnerable to decryption key exposure. Then they improved their scheme in terms of security models, and constructed the first scalable RIBE scheme that can resist decryption key exposure. The main design principle of efficient RIBE schemes employed the CS scheme, and then Park *et al.* [34] put forward a new method for RIBE, in which the private key element is a constant number.

Before deploying ABE into any practical scenarios, one has to solve for the revocation problem. Based on the first practical RIBE [1] introduced in 2008, several followup revocable attribute-based encryption (RABE) schemes have been proposed. Ibraimi *et al.* [10] proposed attribute revocation by introducing a semi-trusted agent, but the third party agent must be honest and online. In Yu *et al.*'s scheme [17], a semi-trusted third party assists in completing attribute revocation, but their scheme only supports the AND threshold. In 2011, Hur and Noh [7] provided an efficient attribute revocation scheme through the KEK tree technique. However, their scheme is vulnerable to client collude, and key maintenance costs are considerable. Yang *et al.* [8] proposed an efficient attribute revocation solution to solve the dynamic changes of the client's access privileges in large-scale systems. In view of the key problem, Xie *et al.* [21] presented an optimized version of Hur *et al.*'s scheme, which minimizes the length of the key and ciphertext. In their scheme, each client has an individual key and a group key. In 2017, Li *et al.* [11] improved Hur *et al.*'s scheme in term of the client collusion problem. In their scheme, the client's private key and KEK are not irrelevant. The client owns the attribute private key and corresponding KEK, then they can restore the plaintext data, which can resist client collusion. Li *et al.* [24] introduced a novel multi-authority ciphertext-policy ABE scheme with attribute-level client revocation and outsourcing decryption. In 2018, Wang *et al.* [29] put forward an efficient revocable attribute-based scheme that grants most of the revocation tasks to the cloud. Recently, Liu *et al.* [35] introduced an efficient and revocable ABE scheme based on the direct revocation approach, by embedding the revocation list into ciphertext. Due to the fact that the existing RABE schemes are vulnerable to decryption key exposure attack, Xu *et al.* [18] proposed a new method, called re-randomizable ABE, to realize re-randomizable key generation and ciphertext delegation. In addition, they refined the security model for RABE to resist decryption key exposure, and put forward a general construction of RABE.

III. PRELIMINARIES AND NOTATION

A. BILINEAR MAPS

Suppose that $\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative cyclic groups whose order is a prime number p, and g is a generator of $\mathbb{G}$. Besides, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map, which must meet the following properties [3]:

1) Bilinear: If $\phi, \varphi \in \mathbb{G}$, and $c, d \in \mathbb{Z}_p$, the equation $e(\phi^c, \varphi^d) = e(\phi, \varphi)^{cd}$ holds on.
2) Non-degeneracy: $e(g, g) \neq 1$.
3) Computability: Given $m, n \in \mathbb{G}$, there exists an efficient algorithm that can calculate $e(m, n)$.

B. ACCESS STRUCTIONS

Denote $\{Q_1, Q_2, \ldots, Q_n\}$ as a set of attributes. For $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$, we can get $C \in \mathbb{A}$ and then define that $\mathbb{A} \subseteq 2^{\{Q_1, Q_2, \ldots, Q_n\}}$ is monotone. An access structure [28] is a collection $\mathbb{A}$, which contains non-empty subsets of $\{Q_1, Q_2, \ldots, Q_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{Q_1, Q_2, \ldots, Q_n\}} \backslash \{\varnothing\}$. The sets in $\mathbb{A}$ and not in $\mathbb{A}$ are called the authorized sets and the unauthorized sets, respectively.

C. LINEAR SECRET SHARING SCHEMES (LSSS)

LSSS [2] can achieve any access policy $\mathbb{A}$ by (M, ρ), in which M is a matrix with ℓ rows and n columns, and the function ρ can map each row of $M_{\ell \times n}$ to a corresponding attribute.

- Consider a n-dimensional column vector $\boldsymbol{v} = (s, y_2, \ldots, y_n)$, where $y_2, \ldots, y_n \in \mathbb{Z}_p$ are selected at random, and s is a secret value to be shared. Then compute the *i-th* share of s as $\lambda_i = M_i \cdot \boldsymbol{v}$, where λ_i belongs to the attribute $\rho(i)$.
- Let $S \in \mathbb{A}$ be an authorized set, and $I = \{i : \rho(i) \in S\} \subseteq \{1, 2, \ldots, l\}$. Then there exist the constants $\{c_i | i \in I\}$ that can meet $\sum_{i \in I} c_i M_i = (1, 0, \ldots, 0)$. Thus $\sum_{i \in I} c_i \lambda_i = s$.

D. DECISIONAL Q-PARALLEL BDHE PROBLEM

Given a group $\mathbb{G}$ with prime order p and a generator g of $\mathbb{G}$. The vector $\boldsymbol{y}$ provided for the adversary consists of

$$
\begin{aligned}
&g, g^s, \ldots, g^{(a^q)}, , g^{(a^{q+2})}, \ldots, g^{(a^{2q})},\\
&\quad \forall 1 \leq j \leq q, \;\; g^{s \cdot b_j}, g^{a/b_j}, \ldots, g^{(a^q/b_j)}, , g^{(a^{q+2}/b_j)}, \ldots, g^{(a^{2q/b_j})},\\
&\quad \forall 1 \leq j, k \leq q, k \neq j, \;\; g^{asb_k/b_j}, \ldots, g^{(a^q sb_k/b_j)},
\end{aligned}
$$

where $a, s, b_1, \ldots, b_q \in \mathbb{Z}_p$ are picked randomly. It is difficult to distinguish $e(g, g)^{a^{q+1}s} \in \mathbb{G}_T$ from a random element in $\mathbb{G}_T$. An algorithm $\mathscr{B}$ that output $\beta \in \{0, 1\}$ has the advantage δ with respect to addressing the decisional q-parallel BDHE in $\mathbb{G}$ if

$$\left|\Pr[\mathscr{B}(\mathbf{y}, e(g, g)^{a^{q+1}s}) = 0] - \Pr[\mathscr{B}(\mathbf{y}, R) = 0]\right| \geq \delta.$$

Definition 1: If any probabilistic polynomial time (PPT) algorithm cannot address the decisional q-parallel BDHE problem with a non-negligible advantage, the decisional q-parallel BDHE hardness assumption [28] holds.

E. PROOF OF OWNERSHIP

Halevi *et al.* [16] proposed a tunable proof of ownership (PoW) scheme, which is an interactive ownership authorization protocol by executing between a prover and a verifier in client-side deduplication, such as Fig. 1. Later on, Blasco *et al.* [30] put forward a clever idea, named bloom filter proof of ownership (BF-PoW), to give a flexible, scalable, and provably secure method.

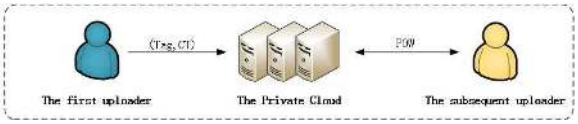


FIGURE 1. Client-side deduplication.

In their scheme, a bloom filter (BF) is initialized by the verifier, and its elements are set to 0. Then the verifier divides the file into chunks of identical length. Suppose that the number of chunks is ω. For $i \in [0, \omega]$, the verifier computes chunk tag $t = \mathcal{H}(F[i])$ and $e = PRF(t, i)$, where $\mathcal{H} : \{0, 1\}^* \to \{0, 1\}^{n_1}$ is a secure hash function and n_1 is a positive integer. Finally, the value of e will be inserted into BF. In the challenge phase, the verifier requires the prover to upload some chunk tags to prove that he indeed owns the file. Only if all chunk tags belong to BF, PoW verification is successful. In our scheme, the cloud needs to send partial ciphertext to the data uploader when PoW is performed, so that the data uploader can restore the random key by utilizing the key derived from the file.

IV. PROBLEM FORMULATION

A. SYSTEM MODEL

Fig. 2 describes the system construction of the proposed scheme, including four entities:

- ***Attribute authority (AA)***: AA is a completely trustworthy server that takes charge of generating the attribute agent key and private key, as well as distributing the prime number. Besides, when a doctor's attribute changes, AA generates the update key and sends it to the private cloud.
- ***Cloud servers***: The cloud servers consist of an untrustworthy public cloud and a semi-trustworthy private cloud. The former mainly provides data storage services, and the latter is mainly in charge of deduplication interaction with patients in the process of the data upload, updating the attribute agent key and ciphertext after the doctor's attribute dynamically changed, and partially decrypting the ciphertext when the doctor accesses the medical records.
- ***Patients***: The entities own the medical records, define access policies and upload the encrypted medical records to the cloud server for data sharing.
- ***Doctors***: The doctors as clients will be assigned the private key based on their attributes, and they can restore the patient's medical records only if their attribute set meets the access policy.

B. NOTATIONS

Then, we define some necessary notations in Table 1 that will be used in the following sections.

TABLE 1. Notations.

Notations	Descriptions
U	The attribute universe in the system
$\mathcal{F}$	Message space
Enc/Dec	Symmetric encryption/decryption algorithm
Φ	Mapping from file tag to prime number
p_i	An element in the prime number set
p	The product of prime numbers based on the file tag
Exp	Time cost of an exponential operation
P	Time cost of a pairing operation
k	Number of the doctor's attributes
l	Number of attributes in the access policy
y	Number of attributes in the new access policy
m	Number of revoked attributes
h	Number of attributes involved in decryption
n_x	Number of nodes in minimum cover

C. FRAMEWORK

The detailed construction of the proposed scheme consists of nine algorithms:

- ***AA.Setup***$(1^\lambda, U) \to (PP, MK, \{VK_x, PK_x\}_{x \in U})$ Given a security parameter λ and an attribute universe U, AA outputs corresponding public parameters PP, a master secret key MK, the version key $\{VK_x\}_{x \in U}$ and the public attribute key $\{PK_x\}_{x \in U}$.
- ***AA.KeyGen***$(PP, MK, A, \{VK_x\}_{x \in A}) \to (SK_1, SK_2)$ Taking as input the public parameters PP, the master secret key MK, an attribute set A, and the version key $\{VK_x\}_{x \in A}$, AA outputs an attribute agent key SK_1 and a private key SK_2 of the doctor.
- ***Encrypt***$(PP, \{PK_x\}_{x \in U}, K_F, (M, \rho)) \to (CT, TK)$ Input the public parameters PP, the public attribute key $\{PK_x\}_{x \in U}$, a random key K_F, and an access policy (M, ρ). The patient outputs a ciphertext CT and a trapdoor key TK.
- ***Re-encrypt***$(PP, \{PK_x\}_{x \in U}, CT, (M', \rho'), TK) \to CT'$ If the uploaded file is duplicate, the private cloud takes the public parameters PP, the public attribute key

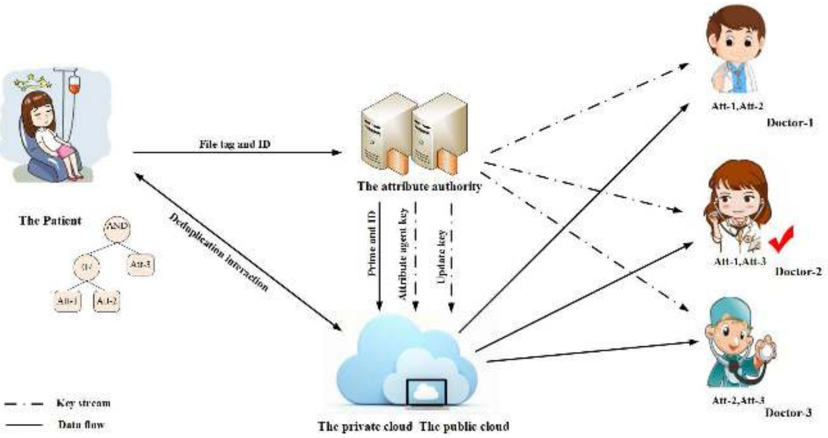


FIGURE 2. System construction of the proposed scheme.

$\{PK_x\}_{x \in U}$, the original ciphertext CT, the new access policy (M', ρ'), and the trapdoor key TK as input, and obtains a new ciphertext CT'.

- ***Pre.Decrypt***$(CT, SK_1) \rightarrow CT''$ Given the ciphertext CT and the attribute agent key SK_1, the private cloud returns CT''.
- ***D.Decrypt***$(CT'', SK_2) \rightarrow F$ Decrypting the downloaded ciphertext CT'' by utilizing the private key SK_2, the doctor restores the medical records F.
- ***UKeyGen***$(j, VK_j) \rightarrow \overline{UK_j}$ Taking the revoked attribute j and the version key VK_j as input, AA returns an update key $\overline{UK_j}$.
- ***SKUpdate***$(SK_1, \overline{UK_j}) \rightarrow \overline{SK_1}$ Taking the attribute agent key SK_1 and the update key $\overline{UK_j}$ as input, the private cloud returns a new attribute agent key $\overline{SK_1}$.
- ***CTUpdate***$(CT, \overline{UK_j}) \rightarrow \overline{CT}$ Taking the ciphertext CT and the update key $\overline{UK_j}$ as input, the private cloud returns a new ciphertext $\overline{CT}$.

D. SECURITY MODEL

In the proposed scheme, we take account of two types of adversaries. The type 1 adversaries are permitted to issue any attribute agent key and update key queries, in addition to the ability to decrypt the challenge ciphertext. The type 2 adversaries are given the trapdoor key in the challenge phase.

We give a security model against the type 1 adversaries, which is carried out by a challenger $\mathscr{B}$ and an adversary $\mathscr{A}_1$.

IND-sCP-CPA game:

- ***Init.*** $\mathscr{A}_1$ formulates an access policy (M^*, ρ^*) that will be challenged.
- ***Setup.*** $\mathscr{B}$ obtains PP, $\{PK_x\}_{x \in U}$ as the public parameters and public attribute key, respectively, then sending them to $\mathscr{A}_1$.
- ***Phase 1.*** $\mathscr{A}_1$ makes a series of Oracle queries as below:
  - $O_{SK_1}(A)$: Make attribute agent key queries about the attribute set A that can not meet the access policy (M^*, ρ^*).
  - $O_{UK}(j)$: Make update key queries about the revoked attribute j.
- ***Challenge.*** $\mathscr{A}_1$ submits the messages F_0, F_1 with identical length to $\mathscr{B}$. On receiving the messages, $\mathscr{B}$ picks a random exponent $\beta \in \{0, 1\}$ and generates the ciphertext CT^*_β of F_β for $\mathscr{A}_1$.
- ***Phase 2.*** The key queries in this phase are consistent with *Phase 1*.
- ***Guess.*** $\mathscr{A}_1$ returns a guess β' of β. If $\beta' = \beta$ then he could achieve victory with the advantage

$$Adv_{\mathscr{A}_1}^{IND-sCP-CPA}(\lambda) = \left| \Pr(\beta' = \beta) - \frac{1}{2} \right|.$$

Definition 2: The proposed scheme is IND-sCP-CPA secure if any PPT adversary cannot win the above game with a non-negligible advantage.

Then we give a security model against the type 2 adversaries for the proposed scheme, which is executed by a challenger $\mathscr{B}$ and an adversary $\mathscr{A}_2$.

PRV-CDA game:

- ***Setup.*** $\mathscr{B}$ generates the public parameters PP and public attribute key $\{PK_x\}_{x \in U}$ for $\mathscr{A}_2$ in this phase.
- ***Challenge.*** $\mathscr{A}_2$ chooses the messages F_0, F_1 with equal length and an access policy (M^*, ρ^*) that is going to be challenged, and sends them to $\mathscr{B}$. $\mathscr{B}$ randomly chooses a bit $\theta \in \{0, 1\}$, gets a challenge ciphertext CT^*_θ about F_θ under (M^*, ρ^*), and then transmits CT^*_θ to $\mathscr{A}_2$.
- ***Guess.*** $\mathscr{A}_2$ returns a guess bit θ'. If $\theta' = \theta$ then he wins the game with the advantage

$$Adv_{\mathscr{A}_2}^{PRV-CDA}(\lambda) = \left| \Pr(\theta' = \theta) - \frac{1}{2} \right|.$$

Definition 3: The proposed scheme is PRV-CDA secure if any PPT adversary cannot win the above game with non-negligible advantage.

V. OUR CONSTRUCTION

Inspired by Wang *et al.*'s scheme [29] and Cui *et al.*'s scheme [4], a revocable attribute-based encryption scheme with efficient deduplication is introduced into eHealth systems. The detailed construction of the proposed scheme includes the following six subsections:

A. SYSTEM INITIALIZATION

Attribute authority (AA) implements system initialization by calling the following *AA.Setup* algorithm.

AA.Setup$(1^{\lambda}, U) \rightarrow (PP, MK, \{VK_x, PK_x\}_{x\in U})$ AA takes a security parameter λ and an attribute universe U as input, then does the following steps as

1) Select a bilinear map $e : (\mathbb{G}, \mathbb{G}) \rightarrow \mathbb{G}_T$, where $\mathbb{G}$ and $\mathbb{G}_T$ are multiplicative cyclic groups of prime order p. Choose a generator g of $\mathbb{G}$, and pick four secure hash functions: $H : \{0,1\}^* \rightarrow \mathbb{G}$, $H_0 : \mathcal{F} \rightarrow \mathbb{Z}_p$, $H_1 : \mathbb{G}_T \rightarrow \mathbb{Z}_p$, $H_2 : \{0,1\}^* \rightarrow \{0,1\}^*$.
2) Randomly select $\alpha, a \in \mathbb{Z}_p$, $h \in \mathbb{G}$, and compute g^a, $e(g,g)^{\alpha}$. The master secret key MK and the public parameters PP are obtained as

$$MK = \alpha,$$
$$PP = (g, h, H, H_0, H_1, H_2, g^a, e(g,g)^{\alpha}).$$

3) For each attribute $x \in U$, pick $v_x \in \mathbb{Z}_p$ as the version key VK_x at random, and compute $PK_x = g^{v_x}$ as the public attribute key.
4) Make PP and $\{PK_x\}_{x\in U}$ as public, while keeping MK and $\{VK_x\}_{x\in U}$ as secret.

B. KEY GENERATION

When a doctor joins medical systems, AA generates an attribute agent key SK_1 and a private key SK_2 corresponding to the attribute by calling the *AA.KeyGen* algorithm. Then AA distributes SK_1 to the private cloud and SK_2 to the doctor.

AA.KeyGen$(PP, MK, A, \{VK_x\}_{x\in A}) \rightarrow (SK_1, SK_2)$ AA firstly inputs the public parameters PP, the master secret key MK, an attribute set $A = \{A_1, \ldots, A_{|A|}\}$ and the version key $\{VK_x\}_{x\in A}$. Then it selects $t, z \in \mathbb{Z}_p$ at random, and obtains the attribute agent key SK_1 and private key SK_2 as

$$SK_1 = \{K = g^{\frac{\alpha}{z}} g^{\frac{at}{z}}, E = g^{\frac{t}{z}}, \{K_x = H(x)^{\frac{t}{zv_x}}\}_{x\in A}\},$$
$$SK_2 = z.$$

C. ENCRYPTION

When the patient prepares to outsource the medical records $F \in \mathcal{F}$ to cloud server, he performs the symmetric encryption algorithms $C = Enc(K_F, F)$ and $C_0 = Enc(K_0, K_F)$, where $K_F \in_R \mathbb{G}_T$ and $K_0 = H_0(F)$. Note that the medical record involves medicines, and their usage and dosage. Then the patient constructs an access policy (M, ρ), and runs the following *Encrypt* algorithm.

Encrypt$(PP, \{PK_x\}_{x\in U}, K_F, (M, \rho)) \rightarrow (CT, TK)$ The patient inputs the public parameters PP, the public attribute key $\{PK_x\}_{x\in U}$, the random key K_F, and the access policy (M, ρ). Let M be an $l \times n$ matrix. Then the patient performs the following steps as

1) Choose $s, y_2, \ldots, y_n \in \mathbb{Z}_p$ at random, generate a vector $\boldsymbol{v} = (s, y_2, \ldots, y_n) \in \mathbb{Z}_p^n$, then compute $\lambda_j = M_j \cdot \boldsymbol{v}$, where s is a secret value and $j \in [1, l]$.
2) Select $r_j \in \mathbb{Z}_p$ at random, then output the following components as

$$B = g^s, L = g^{H_0(F)} h^{H_1(K_F)},$$
$$D = K_F \cdot e(g,g)^{\alpha s}, TK = (g^a)^s,$$
$$C_j = g^{a\lambda_j} H(\rho(j))^{r_j}, D_j = g^{v_{\rho(j)} r_j}.$$

3) Calculate $T = H_2(K_0)$ as the file tag.

Finally, the patient could obtain a ciphertext $CT = \{L, (M, \rho), B, C, D, \{C_j, D_j\}_{j\in[1,l]}\}$, a trapdoor key TK and a file tag T. The trapdoor key TK is utilized as a re-encrypt key for the private cloud in deduplication phase.

D. DEDUPLICATION PROTOCOL

The patient sends a deduplication request (ID, T) to AA. Then AA generates a prime number p_i through a mapping Φ, and transmits (p_i, ID) to the private cloud, where ID denotes the patient's identity. It is noted that the mapping Φ satisfies the following properties:

- For each file tag, generate a unique prime number.
- Take the same (different) file tag as input, output the same (different) prime number.

After receiving p_i, the private cloud executes a division operation $p_i | p$ (as shown in Algorithm 1).

Algorithm 1 Deduplication Protocol (The Private Cloud)

```
Input: The prime number $p_i$ corresponding to file tag $T$,
    the product of existing prime numbers $p$
Output: The outcome of deduplication
  if $p_i | p$ then
    Duplicate file has been found, sends $C_0$ to the patient
    and requests for PoW
    if PoW verification passed then
      Re-encrypts the original ciphertext under the union of
      two access policies $(M', \rho')$
    else
      Rejects the patient
    end if
  else if $p_i \nmid p$ then
    Duplicate file has not been found, and computes $p = p_i \cdot p$
  end if
```

If $p_i \nmid p$, we call the patient as the first uploader. Then the private cloud computes $p = p_i \cdot p$, and requests the patient for the ciphertext. Once received the ciphertext CT, the private cloud divides C into chunks of equal length, then generates a tag for each chunk, and inserts a pseudo random function (PRF) of each chunk tag into BF. Finally, the private cloud stores $((T, p_i, L), BF, C_0)$, and sends the ciphertext CT to the public cloud.

Otherwise, the private cloud regards the patient as the subsequent uploader and triggers PoW verification.

Specifically, the private cloud randomly picks some chunk indexes array J and sends (J, C_0) to the patient. The patient decrypts C_0 to obtain the random key K_F, and encrypts the local medical records F' to obtain the ciphertext $\widetilde{C}$. Then, divides $\widetilde{C}$ into chunks, computes chunk tag of array J, and sends them to the private cloud. The chunk tag is processed with PRF, and private cloud checks whether the generated bitstring is a member of BF. If all chunk tag belongs to BF, the patient passes the PoW verification. The private cloud sends an access link to him, requests the access policy, and then re-encrypts the original ciphertext under the union of two access policies (M', ρ') through the following *Re-encrypt* algorithm; otherwise, rejects the patient.

After completing the process of the deduplication, the patient deletes the local medical records while storing the key K_0 derived from the medical records.

Re-encrypt$(PP, \{PK_x\}_{x \in U}, CT, (M', \rho'), TK) \rightarrow CT'$ The private cloud inputs the public parameters PP, the public attribute key $\{PK_x\}_{x \in U}$, the original ciphertext CT, the new access policy $(M'_{l' \times n'}, \rho')$, and the trapdoor key TK. Then the private cloud executes the following steps:

1) Pick a vector $\bar{\boldsymbol{v}} = (\bar{s}, y'_{2'}, \ldots, y'_{n'})$ at random and denote $\boldsymbol{v}' = (s', y'_{2'}, \ldots, y'_{n'})$, where $s' = s + \bar{s}$ and $y'_{2'}, \ldots, y'_{n'} \in_R \mathbb{Z}_p$.
2) Calculate the partial ciphertext components as

$$C' = C, B' = B \cdot g^{\bar{s}},$$
$$L' = L, D' = D \cdot e(g, g)^{\alpha \bar{s}}.$$

3) For $j' \in [1, l']$, select $r'_{j'} \in \mathbb{Z}_p$ at random, and calculate the ciphertext for the attribute components as

$$C'_{j'} = (g^a)^{\boldsymbol{v}' \cdot M'_{j'}} H(\rho'(j'))^{r'_{j'}},$$
$$D'_{j'} = g^{v_{\rho'(j')} r'_{j'}}.$$

Finally, the private cloud sends a new ciphertext

$$CT' = \{L', \left(M', \rho'\right), C', B', D', \{C'_{j'}, D'_{j'}\}_{j' \in [1, l']}\}$$

to the public cloud.

Remark 1: The value of $C'_{j'}$ could be computed by the private cloud even when the private cloud does not hold s.

$$\begin{aligned} C'_{j'} &= (g^a)^{\boldsymbol{v}' \cdot M'_{j'}} H(\rho'(j'))^{r'_{j'}} \\ &= (g^a)^{s' m'_{j'1} + \ldots + y'_{n'} m'_{j'n'}} H(\rho'(j'))^{r'_{j'}} \\ &= (g^a)^{s m'_{j'1}} (g^a)^{\bar{s} m'_{j'1} + \ldots + y'_{n'} m'_{j'n'}} H(\rho'(j'))^{r'_{j'}}. \end{aligned}$$

Remark 2: In existing deduplication schemes, the efficiency of physical copy search was elevated through the deduplication decision tree technology, which can reduce the time complexity from linear-level to logarithm-level. However, our scheme could search the duplicate file by only executing a division operation, which is efficient in terms of the private cloud.

E. DATA DECRYPTION

If the patient wants to access the medical records or checks them whether are tampered, he will request the ciphertext according to the link. Upon receiving the ciphertext, the patient computes as

$$K_F = Dec(K_0, C_0), F = Dec(K_F, C).$$

Then he matches whether $H_0(F)$ is consistent with K_0.

If a doctor intends to access the medical records, the private cloud partially decrypts the ciphertext through the *Pre.Decrypt* algorithm, then the doctor restores the final data through the *D.Decrypt* algorithm. Especially, the data consistency will be tested after decrypting the ciphertext.

Pre.Decrypt$(CT, SK_1) \rightarrow CT''$ The private cloud inputs a ciphertext CT and the attribute agent key SK_1 for an attribute set A. Denote $I = \{j : \rho(j) \in A\}$ and A is an authorized set. The constants $c_j \in \mathbb{Z}_p$ could be computed such that they can be satisfied $\sum_{j \in I} c_j A_j = (1, 0, .., 0)$. The calculation process is given as follows:

$$\begin{aligned} CT'' &= e(B, K) \cdot \prod\nolimits_{j \in I} \left(\tfrac{e(D_j, K_{\rho(j)})}{e(C_j, E)} \right)^{c_j} \\ &= e(g, g)^{\frac{\alpha s}{z}}. \end{aligned}$$

D.Decrypt$(CT'', SK_2) \rightarrow F$ The doctor inputs CT'' and the private key SK_2, and computes $K_F = \frac{D}{(CT'')^{SK_2}}$. Finally, the medical records F is restored by applying a symmetric decryption algorithm $F = Dec(K_F, C)$. If $g^{H_0(F)} h^{H_1(K_F)} = L$, the doctor obtains F; otherwise, the doctor rejects the ciphertext.

F. ATTRIBUTE REVOCATION

When a patient's attributes change dynamically, the attribute revocation is occurred to protect the data privacy. AA runs the *UKeyGen* algorithm to generate an update key. Then the private cloud updates the attribute agent key and ciphertext through the *SKUpdate* algorithm and the *CTUpdate* algorithm, respectively.

- ***UKeyGen***$(j, VK_j) \rightarrow \overline{UK_j}$ When a doctor's attribute is revoked, AA inputs the version key VK_j corresponding to the revoked attribute j.
    1) Choose a new version key $\overline{VK_j} = \overline{v_j}$ at random.
    2) Calculate the update key $\overline{UK_j} = \frac{\overline{v_j}}{v_j}$, and send $\overline{UK_j}$ to the private cloud.

  Then, for the revoked attribute j, AA updates the public attribute key PK_j as

$$\overline{PK_j} = (PK_j)^{\overline{UK_j}} = (g^{v_j})^{\frac{\overline{v_j}}{v_j}} = g^{\overline{v_j}}.$$

- ***SKUpdate***$(SK_1, \overline{UK_j}) \rightarrow \overline{SK_1}$ The private cloud updates the non-revoked doctors' attribute agent key SK_1 by utilizing the update key $\overline{UK_j}$, and obtains a new attribute agent key $\overline{SK_1}$ as

$$\overline{SK_1} = \begin{cases} \overline{K} = K, \overline{E} = E \\ x \in A \backslash \{j\} : \overline{K_x} = K_x \\ x = j : \overline{K_j} = H(j)^{\frac{t}{z v_j} \cdot (\overline{UK_j})^{-1}} = H(j)^{\frac{t}{z \overline{v_j}}} \end{cases}$$

- ***CTUpdate***$(CT, \overline{UK_j}) \rightarrow \overline{CT}$ When receiving $\overline{UK_j}$, the private cloud computes the new ciphertext as

$$\overline{CT} = \begin{cases} \overline{C} = C, \overline{B} = B, \overline{L} = L, \overline{D} = D, \overline{C_i} = C_i \\ \rho(i) \neq j : \overline{D_i} = D_i \\ \rho(i) = j : \overline{D_j} = (g^{v_{\rho(j)} r_j})^{\overline{UK_j}} = g^{\overline{v_{\rho(j)}} r_j} \end{cases}$$

VI. SECURITY ANALYSIS

A. CORRECTNESS

The correctness of decryption process in the proposed scheme is demonstrated as follows:

$$\begin{aligned} CT'' &= e(B, K) \cdot \prod_{j \in I} \left(\frac{e(D_j, K_{\rho(j)})}{e(C_j, E)} \right)^{c_j} \\ &= e(g^s, g^{\frac{\alpha}{z}} g^{\frac{at}{z}}) \cdot \prod_{j \in I} \left(\frac{e(g^{v_{\rho(j)} r_j}, H(\rho(j))^{\frac{t}{z v_{\rho(j)}}})}{e(g^{a\lambda_j} H(\rho(j))^{r_j}, g^{\frac{t}{z}})} \right)^{c_j} \\ &= e(g^s, g^{\frac{\alpha}{z}}) e(g^s, g^{\frac{at}{z}}) \cdot \prod_{j \in I} \left(\frac{e(g, H(\rho(j))^{r_j \frac{t}{z}}}{e(g^{a\lambda_j} H(\rho(j))^{r_j}, g^{\frac{t}{z}})} \right)^{c_j} \\ &= \frac{e(g^s, g^{\frac{\alpha}{z}}) e\left(g^s, g^{\frac{at}{z}}\right)}{\prod_{j \in I} e\left(g^{a\lambda_j}, g^{\frac{t}{z}}\right)^{c_j}} \\ &= e(g, g)^{\frac{\alpha s}{z}}. \\ K_F &= \frac{D}{(CT'')^{SK_2}} = \frac{K_F \cdot e(g, g)^{\alpha s}}{e(g, g)^{\frac{\alpha s}{z} \cdot z}}. \end{aligned}$$

B. SECURITY

The ciphertext is re-encrypted under the union of access policy when the medical record is duplicate. However, due to the fact that the distribution of the new ciphertext is consistent with the original ciphertext, we only analysis the sematic security of the original ciphertext.

Theorem 1: Suppose that the decisional q-parallel BDHE assumption holds in $\mathbb{G}$, and the symmetric encryption scheme is secure. The proposed scheme satisfies the IND-sCP-CPA security, which is similar to Waters *et al.*'s [28] proof.

Proof: Suppose that there exists an adversary $\mathscr{A}_1$ who can break the proposed scheme with the non-negligible advantage $\delta = Adv_{\mathscr{A}_1}$, the challenger $\mathscr{B}$ could address the decisional q-parallel BDHE problem with the advantage $\frac{\delta}{2}$.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. Denote g as a generator of $\mathbb{G}$ and e as a bilinear map, $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. $\mathscr{B}$ is given a decisional q-parallel BDHE problem $(\boldsymbol{y}, Z)$.

Init. $\mathscr{A}_1$ formulates an access policy (M^*, ρ^*) that he is forthcoming to challenge, where M^* is an $l^* \times n^*$ matrix and $n^* < q$.

Setup. $\mathscr{B}$ has to provide $\mathscr{A}_1$ for the public parameters PP and the public attribute key $\{PK_x = g^{v_x}\}_{x \in U}$.

1) Randomly choose $\alpha' \in \mathbb{Z}_p$, where the value of α can be computed as $\alpha = \alpha' + a^{q+1}$.
2) Select $h \in \mathbb{G}$ and four secure hash functions H, H_0, H_1, H_2 at random.
3) Perform the random oracle using a hash list L_H. When $H(x)$ is already presented in L_H, $\mathscr{B}$ returns the identical answer as in the list; otherwise, $\mathscr{B}$ chooses $z_x \in \mathbb{Z}_p$ at random, and simulates $H(x)$ as

$$H(x) = g^{z_x} \prod_{i \in Y} g^{\frac{aM^*_{i,1}}{b_i}} \cdot g^{\frac{a^2 M^*_{i,2}}{b_i}} \cdots g^{\frac{a^{n^*} M^*_{i,n^*}}{b_i}},$$

where Y denotes a set and i as its element can satisfy $\rho^*(i) = x$.If the set $Y = \emptyset$, then $H(x) = g^{z_x}$
4) Select the version key $v_x \in \mathbb{Z}_p$ at random, and compute $PK_x = g^{v_x}$, where $x \in U$.

Finally, the public parameters and the public key are published as $PP = (g, h, H, H_0, H_1, H_2, g^a, e(g, g)^\alpha)$ and $\{PK_x = g^{v_x}\}_{x \in U}$, respectively.

Phase 1. $\mathscr{B}$ answers the attribute agent key and the update key queries for an attribute set A in this phase, where A does not meet (M^*, ρ^*). $\mathscr{A}_1$ can issue the following Oracle queries in the polynomial time.

$O_{SK_1}(A)$: $\mathscr{B}$ first randomly chooses $r, z \in \mathbb{Z}_p$.

1) For $\rho^*(i) \in A$, find a vector $\boldsymbol{w} = (c_1, \ldots, c_{n^*}) \in \mathbb{Z}_p^{n^*}$ satisfying $c_1 = -1$ and $\boldsymbol{w} \cdot M_i^* = 0$, and define t by picking $r \in \mathbb{Z}_p$ as

$$t = r + c_1 a^q + c_2\, a^{q-1} + \cdots + c_{n^*} a^{q-n^*+1}.$$

2) E is computed as

$$E = g^{\frac{r}{z}} \prod_{i=1,\ldots,n^*} \left(g^{a^{q+1-i}}\right)^{\frac{c_i}{z}} = g^{\frac{t}{z}}.$$

3) According to the definition of t, even if g^{at} comprises $g^{-a^{q+1}}$, $g^{-a^{q+1}}$ is going to be cancelled out by the component in g^α. Then, K could be computed as:

$$K = g^{\frac{\alpha'}{z}} g^{\frac{ar}{z}} \prod_{i=2,\ldots,n^*} \left(g^{a^{q+2-i}}\right)^{\frac{c_i}{z}}.$$

4) If $x \in A$ and the index i that satisfies $\rho^*(i) = x$ cannot be found, then $\mathscr{B}$ computes $K_x = E^{\frac{z_x v_x}{z}}$; otherwise, it is ensured that there is no item in the form of g^{a^{q+1}/b_i} in K_x. As a result of $M_i^* \cdot \boldsymbol{w} = 0$, $\mathscr{B}$ calculates K_x as

$$K_x = E^{\frac{z_x}{z v_x}} \prod_{i \in Y} \prod_{j=1,\ldots,n^*} \left(\left(g^{\frac{a^j}{b_i}}\right)^r\right)^{\frac{M^*_{i,j}}{z v_x}} \cdot \prod_{i \in Y} \prod_{j=1,\ldots,n^*} \prod_{\substack{k=1,\ldots,n^* \\ k \neq j}} \left(\left(g^{\frac{a^{q+1+j-k}}{b_i}}\right)^{c_k}\right)^{\frac{M^*_{i,j}}{z v_x}}.$$

$O_{UK}(j)$. When the attribute j has been revoked, $\mathscr{B}$ chooses the random value $v_j^* \in \mathbb{Z}_p$ as the new version key, and returns the update key $\widetilde{UK_j} = \frac{v_j^*}{v_j}$ to $\mathscr{A}_1$.

Challenge. $\mathscr{A}_1$ submits two messages F_0 and F_1, where the length of messages are indistinguishable. $\mathscr{B}$ selects $\beta \in \{0, 1\}$, $K_F^* \in \mathbb{G}_T$ at random, and calculates $C^* = Enc(K_F^*, F_\beta)$, $D^* = K_F^* \cdot Z \cdot e(g^s, g^{\alpha'})$ and $B^* = g^s$.

The value of C_i contains the term that we cannot simulate, which leads to simulate C_i is challenge part. However, one effective way to cancel out these is to utilize the linear secret sharing scheme. $\mathscr{B}$ chooses $y'_2, \ldots, y'_{n^*}$ at random, and splits s through utilizing

$$\boldsymbol{v} = (s, sa + y'_2, sa + y'_3, \ldots, sa^{n-1} + y'_{n^*}) \in \mathbb{Z}_p^{n^*}.$$

In meanwhile, $\mathscr{B}$ randomly chooses values $r'_1, \ldots, r'_{l^*}$, defines $T_i = \{i : \rho^*(i) = \rho^*(k), k \neq i\}_{i=1,\ldots,n^*}$, and denotes $r_i = -r'_i - sb_i$. The remaining parts of the challenge ciphertext are calculated as

$$\begin{aligned}
L^* &= g^{H_0(F_\beta)} h^{H_1(K_F^*)}, \\
D_i^* &= g^{-r'_i v_{\rho^*(i)}} g^{-sb_i v_{\rho^*(i)}}, \\
C_i^* &= H\left(\rho^*(i)\right)^{-r'_i} \left(\prod_{j=2,\ldots,n^*} (g^a)^{M^*_{i,j} y'_j}\right) \left(g^{b_i s}\right)^{-z_{\rho^*(i)}} \\
&\quad \cdot \prod_{k \in T_i} \prod_{j=1,\ldots,n^*} \left(g^{a^j \cdot s \cdot (\frac{b_i}{b_k})}\right)^{-M^*_{k,j}}.
\end{aligned}$$

When $CT_\beta^* = \{L^*, (M^*, \rho^*), B^*, C^*, D^*, \{C_i^*, D_i^*\}_{i \in [1,l^*]}\}$ is simulated, the challenger $\mathscr{B}$ submits CT_β^* to $\mathscr{A}_1$.

Phase 2. The key queries in this phase are consistent with *Phase 1*.

Guess. $\mathscr{A}_1$ will return a guess β' of β. If $\beta' = \beta$, then $\mathscr{B}$ returns $\zeta = 0$ and gives a guess value $Z = e(g, g)^{a^{q+1}s}$; otherwise, $\mathscr{B}$ outputs $\zeta = 1$, and views Z as a random element in $\mathbb{G}_T$.

When $\beta' = \beta$, $\mathscr{B}$ will guess $\zeta = 0$, where $\Pr[\zeta' = \zeta | \zeta = 0] = \frac{1}{2} + Adv_{\mathscr{A}_1}$, and in this case $\mathscr{A}_1$ learns nothing of β, where $\Pr[\beta' = \beta | \zeta = 1] = \frac{1}{2}$. When $\beta' \neq \beta$ then $\mathscr{B}$ will guess $\zeta' = 1$, we have $\Pr[\zeta' = \zeta | \zeta = 1] = \frac{1}{2}$. Thus, the advantage of $\mathscr{B}$ with respect to address the decisional q-parallel BDHE problem is

$$\begin{aligned}
\Pr[\zeta' = \zeta] - \frac{1}{2} &= \left| \frac{1}{2} \Pr[\zeta' = \zeta | \zeta = 1] - \frac{1}{2} \right| \\
&= \left| \frac{1}{2}(\frac{1}{2} + \delta) + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} \right| \\
&= \frac{\delta}{2}.
\end{aligned}$$

Theorem 2: If the decisional BDH assumption holds, and the symmetric encryption scheme is secure, then the proposed scheme is PRV-CDA [4] secure. We give the proof as follows:

Proof: Assuming that an adversary $\mathscr{A}_2$ could break PRV-CDA security, the challenger $\mathscr{B}$ has the ability to address the decisional BDH problem by exploiting the advantage of $\mathscr{A}_2$. $\mathscr{B}$ is given $(g, g^{t_1}, g^{t_2}, g^{t_3}, W)$, aiming to differentiate whether $W = e(g, g)^{t_1 t_2 t_3}$ or W is a random element in $\mathbb{G}_T$.

Setup. The challenger $\mathscr{B}$ selects $a, \{v_x\}_{x \in U} \in \mathbb{Z}_p$ and $h \in \mathbb{G}$ at random. Then he computes g^a and the public attribute key $\{PK_x = g^{v_x}\}_{x \in U}$. He also randomly selects four secure hash functions H, H_0, H_1, H_2, then announces the public parameters $PP = (g, h, H, H_0, H_1, H_2, g^a, e(g^{t_1}, g^{t_2}))$ and the public attribute key $\{PK_x\}_{x \in U}$. It's worth noting that $\mathscr{B}$ does not hold the value of $\alpha = t_1 t_2$.

Challenge. $\mathscr{A}_2$ submits the messages F_0, F_1 of same length and an access policy (M^*, ρ^*) to $\mathscr{B}$, where M^* is an $l^* \times n^*$ matrix. Then $\mathscr{B}$ randomly selects F_θ ($\theta \in \{0, 1\}$), picks $t_3, y_{2^*}, \ldots, y_{n^*} \in \mathbb{Z}_p$, $K_F^* \in \mathbb{G}_T$ at random, and lets $\boldsymbol{v} = (t_3, y_{2^*}, \ldots, y_{n^*})$, $\widetilde{\boldsymbol{v}} = (\overline{t_3}, y_{2^*}, \ldots, y_{n^*})$. For the attribute $j \in [1, l^*]$, $\mathscr{B}$ selects $r_j^* \in \mathbb{Z}_p$ at random, and then returns the ciphertext tuple CT_θ^* and the trapdoor key TK^* as follows:

$$\begin{aligned}
B^* &= g^{t_3}, L^* = g^{H_0(F_\theta)} h^{H_1(K_F^*)}, \widetilde{B}^* = g^{\overline{t_3}}, \\
C^* &= Enc(K_F^*, F_\theta), D^* = K_F^* \cdot W, TK^* = (g^{t_3})^a, \\
C_j^* &= g^{a\lambda_j^*} H(\rho^*(j))^{r_j^*}, D_j^* = g^{v_{\rho^*(j)} r_j^*},
\end{aligned}$$

where the challenger $\mathscr{B}$ can calculate the value of C_j^* as

$$\begin{aligned}
C_j^* &= (g^a)^{\boldsymbol{v} \cdot M_j^*} H(\rho^*(j))^{r_j^*} \\
&= (g^a)^{(t_3 + \overline{t_3}) m_{j1}^* + \ldots + y_{n^*} m_{jn^*}^*} H(\rho^*(j))^{r_j^*} \\
&= (g^{t_3})^{a m_{j1}^*} (g^a)^{\overline{t_3} m_{j1}^* + \ldots + y_{n^*} m_{jn^*}^*} H(\rho^*(j))^{r_j^*}.
\end{aligned}$$

Since $W = e(g, g)^{t_1 t_2 t_3}$, it is obvious that the distribution of the new ciphertext

$$CT_\theta^* = \left\{ L^*, (M^*, \rho^*), B^*, C^*, D^*, \left\{ C_j^*, D_j^* \right\}_{j \in [1,l^*]} \right\}$$

and the trapdoor key TK^* are same as the input as *Re-encrypt* algorithm in the view of $\mathscr{A}_2$.

If $\mathscr{A}_2$ returns a correct guess $\theta' = \theta$, then $\mathscr{B}$ returns 1 that denotes $W = e(g, g)^{t_1 t_2 t_3}$; otherwise, he returns 0 that denotes W is randomly chosen in $\mathbb{G}_T$.

Thus, if the adversary $\mathscr{A}_2$ breaks the PRV-CDA security, the challenger $\mathscr{B}$ could address the decisional BDH problem. Thus the proposed scheme is PRV-CDA secure.

VII. PERFORMANCE ANALYSIS

In this section, we compare the proposed scheme with some existing schemes, and assess their performance in terms of functionality and computation cost.

A. FUNCTIONALITY COMPARISONS

Table 2 describes the result of functionality comparison between the proposed scheme and some existing schemes [4], [8], [11]. The scheme [11] uses access tree as access control, while our scheme and schemes [4], [8] employ LSSS as access policy to achieve more flexible access control. Besides, only our scheme and the scheme [4] can realize the ciphertext deduplication, which saves plenty of storage space. To protect the revoked clients from accessing the sensitive data, the proposed scheme and schemes [8], [11] implement attribute revocation by updating the key and ciphertext. Moreover, only our scheme supports outsourcing decryption, reducing the computation burden on the clients. Obviously, the proposed scheme enjoys more comprehensive functionalities compared with other schemes.

TABLE 2. The comparison of functionality.

Schemes	Access control	Ciphertext deduplication	Attribute revocation	Key update	Ciphertext update	Outsourcing decryption
Cui et al. [4]	LSSS	√	×	×	×	×
Yang et al. [8]	LSSS	×	√	√	√	×
Li et al. [11]	Access tree	×	√	√	√	×
Our scheme	LSSS	√	√	√	√	√

TABLE 3. The comparison of computation cost.

	Computation cost					
Schemes	*KeyGen*	*Encrypt*	*Re-encrypt*	*D.Decrypt*	*SKUpdate*	*CTUpdate*
Cui et al. [4]	$(3k+4)Exp$	$(5l+6)Exp$	$(5y+2)Exp$	$(h+2)Exp+(3h+1)P$	-	-
Yang et al. [8]	$(k+4)Exp$	$(4l+4)Exp$	-	$hExp+(2h+1)P$	$mExp$	$2mExp$
Li et al. [11]	$(5k+2)Exp$	$(2l+2)Exp$	$(y+yn_x)Exp$	$hExp+(3h+1)P$	$3mExp$	$(ln_x+2m+3)Exp$
Ours	$(k+3)Exp$	$(3l+5)Exp$	$(3y+2)Exp$	Exp	$mExp$	$mExp$

B. THEORETICAL ANALYSIS

Table 3 displays the computation cost of all the comparison schemes. Obviously, the computation cost of our scheme is less than schemes [4], [8], [11]. Specifically, in the *Encrypt* algorithm, $3l + 5$ exponential operations are required in the proposed scheme while more exponential operations in schemes [4] and [8]. In the *Re-encrypt* algorithm, our scheme consumes the least exponential operations comparing with schemes [4] and [11]. Besides, one exponential operation is conducted in the *Decrypt* algorithm to restore the medical records, which is more efficient for the clients in comparison with other schemes. Furthermore, our scheme consumes $2m$ exponential operations to achieve the attribute revocation additionally, reducing by m exponential operations compared with the scheme [8].

C. IMPLEMENTATION

Primarily, for more practical performance assessment, we choose a real-world dataset and execute a series of experiments using Stanford Pairing-Based Crypto (PBC) library [36] in VC++ 6.0. on a computer whose configuration is AMD, CPU, Ryzen 5 1500X@3.50GHZ and 8GB RAM. The size of $\mathbb{G}$ and $\mathbb{Z}_p$ is set to 512 bits whereas the size of $\mathbb{G}_T$ is set to 1024 bits, and the supersingular curve $y^2 = x^3 + x$ is employed when supplying ECC group.

Fig. 3 depicts the computation cost of the upload phase. In this phase, the patient needs to execute the encryption operation of medical records as well as the key encapsulation operation. The AES 128 encryption algorithm is used to test the encryption time of different medical records ranging from 10 M to 100 M. The computation cost of encryption increases linearly with the medical record size grows. When the file size is 30 M and 60M, the computation cost of encrypting the medical records is 347 ms and 634 ms, respectively. Due to data sharing, the patient needs to generate the key ciphertext. We can see that the computation cost of key encapsulation grows linearly with the file size when fixing the attribute number to 10. When the file size is 30 M and 60 M, generating the key ciphertext consumes 409.311 ms and 428.619 ms, respectively.

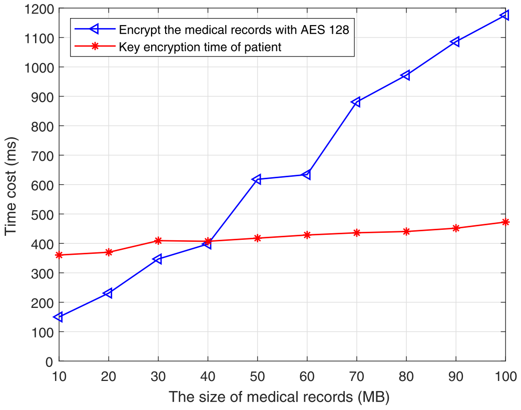


FIGURE 3. The computation cost for upload phase.

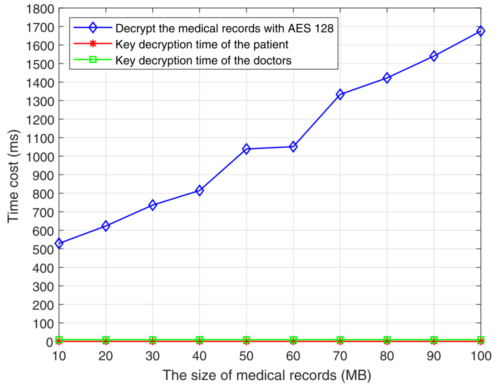
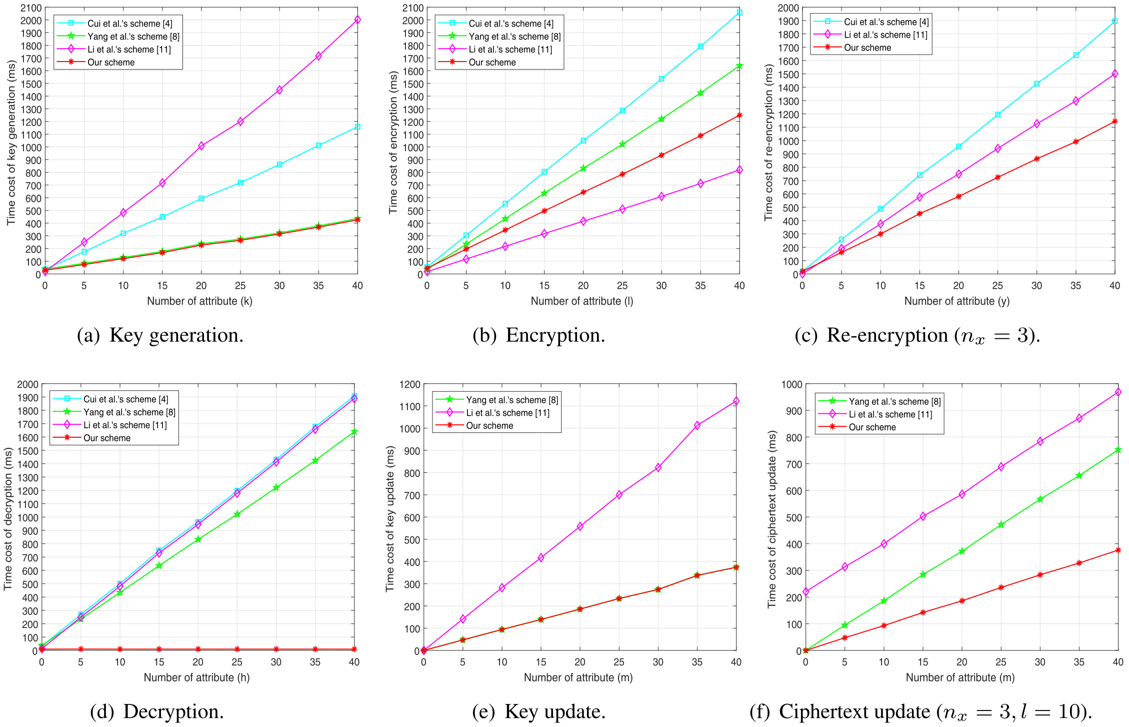


FIGURE 4. The computation cost for download phase.

Fig. 4 describes the computation cost of the download phase. In this phase, the patient and the doctors need to obtain the key and then execute the decryption operation of medical records. Specifically, the computation cost for a patient to



(a) Key generation.

(b) Encryption.

(c) Re-encryption ($n_x = 3$).

(d) Decryption.

(e) Key update.

(f) Ciphertext update ($n_x = 3, l = 10$).

FIGURE 5. The comparison of computation cost for ABE relevant phases.

obtain the random key is negligible since the symmetric key is stored in the local. Due to the outsourcing decryption, the doctors obtain the random key only executing one exponential operation. Besides, the AES 128 decryption algorithm is used to test the decryption time of different medical records ranging from 10 M to 100M. Similarly, the computation cost of decryption grows linearly with the medical record's size increases. When the file size is 50 M and 80 M, generating the ciphertext consumes 367 ms and 862 ms, respectively.

In addition, due to the fact that the random key K_F is encrypted under an access policy in ABE, we also measure the computation costs incurred by ABE relevant phases, and compare them with that of the existing schemes in Fig. 5. The attribute numbers are varying from 0 to 40. We repeat the experiment 100 trials, and set the averaged value as the final experimental result.

As shown in Fig. 5(a), Fig. 5(b) and Fig. 5(c), the computation costs of key generation, encryption and re-encryption increase linearly with the number of attributes in the proposed scheme and schemes [4], [8], [11]. In key generation phase, the consumed time of our scheme and [8] are approximately identical. However, the computation cost of our scheme is approximately 34%, 21% of schemes [4] and [11], respectively. In encryption phase, with the exception of [11], the computation cost of our scheme is efficient compared with schemes [4] and [8], and is only 60% of [4]. When the data is duplicate, our scheme needs to execute the re-encryption phase, and consumes the least exponential operations comparing with schemes [4] and [11]. Although the computation costs of updating the key and ciphertext increase linearly with the number of attributes in our scheme and schemes [8], [11], they are acceptable in order to ensure the patients' privacy. In addition, our scheme consumes the least time in the phases of key update and ciphertext update comparing with schemes [8] and [11] from Fig. 5(e) and Fig. 5(f). Furthermore, since the proposed scheme utilizes the outsourcing decryption, the decryption time is a constant while in schemes [4], [8], [11] increases linearly with the ciphertext policy's complexity grows from Fig. 5(d). The decryption cost of our scheme is approximately 9.5 ms, which is available for clients with limited computing power. Therefore, from the indicated results, our proposed scheme surpasses the existing schemes in terms of computation overhead while achieving the desired functionality requirements.

Next, we choose Jiang *et al.*'s scheme [19] and Yang *et al.*'s scheme [23] as comparisons to analyze the performance of physical copy search method in our scheme, and use *C*# language and System.Runtime.Numerics library version 4.1.1.0. to measure the computation costs of comparison schemes when searching duplicate file. Suppose that the eHealth system has 1000 prescriptions, we measure the maximum search time for all comparison schemes as the experiment results.

Fig. 6 shows the comparison of computation cost for different physical copy search methods between the proposed scheme and schemes in [19] and [23]. As can be seen, when the number of file is 400 and 1000, the scheme [19] consumes 210.645 ms and 232.869 ms, respectively. Besides, the server can search the physical copy with logarithm-level

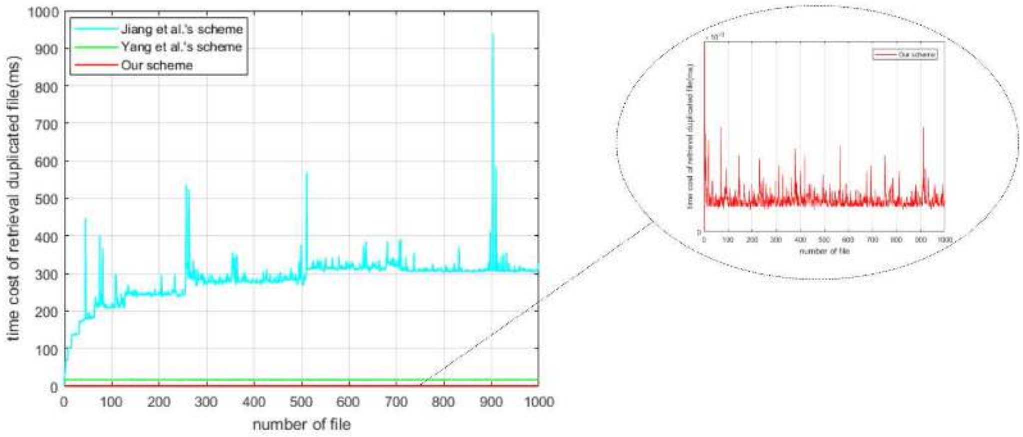


FIGURE 6. The comparison of computation cost for different physical copy search methods.

computation cost since the deduplication decision tree is employed in [19]. In the case of the same number of files, the computation cost of physical copy search in [23] is 26.029 ms and 25.987 ms, respectively. Thus the computation cost of physical copy search in [23] is constant-level, which is more efficient compared with [19] since B+ tree of order f is adopted and then two pairing operations are executed. Different from the above two methods, our scheme utilizes the nature of prime number to achieve the physical copy search, and the computation cost of our scheme is nearly a negligible constant, which is more efficient than schemes [19] and [23].

VIII. CONCLUSIONS

In this paper, we have proposed the first ABE-based deduplication scheme for eHealth systems, which realizes the efficient deduplication and attribute revocation. The proposed scheme allows patients to share their medical records with other parties who possess the access right, and promises the private cloud to delete the redundant copy of the identical medical records to save the storage overheads. Besides, our scheme realizes the attribute revocation to ensure the privacy of patients, and introduces the outsourcing decryption to reduce the computation burden on doctors. Finally, we conduct security and performance analysis to assess the availability of our scheme. The corresponding results reflect that the proposed scheme can realize efficient deduplication while ensuring the privacy of patients in eHealth systems.

REFERENCES

[1] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proc. 15th ACM Conf. Comput. Commun. Secur. (CCS)*, H. Krawczyk, Ed. New York, NY, USA, Oct. 2008, pp. 417–426.

[2] A. B. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 6632. Berlin, Germany: Springer, 2011, pp. 568–588.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Annu. Int. Cryptol. Conf.*, vol. 2139, 2001, pp. 213–229.

[4] H. Cui, R. H. Deng, Y. Li, and G. Wu, "Attribute-based storage supporting secure deduplication of encrypted data in cloud," *IEEE Trans. Big Data*, to be published.

[5] P. Ray and J. Wimalasiri, "The need for technical solutions for maintaining the privacy of EHR," in *Proc. 28th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBS)*, Aug./Sep. 2006, pp. 4686–4689.

[6] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, May 2007, pp. 321–334.

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

[8] K. Yang, X. Jia, and K. Ren, "Attribute-based fine-grained access control with efficient revocation in cloud storage systems," in *Proc. 8th ACM SIGSAC Symp. Inform. Comput. Commun. Secur.*, May 2013, pp. 523–528.

[9] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in *Proc. 22nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2002, pp. 617–624.

[10] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," in *Proc. 10th Int. Workshop Inf. Secur. Appl. (WISA)*, in Lecture Notes in Computer Science, vol. 5932, H. Y. Youm and M. Yung, Eds. Berlin, Germany: Springer, 2009, pp. 309–323.

[11] 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, Jun. 2018.

[12] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, "Message-locked encryption for lock-dependent messages," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 8042. Berlin, Germany: Springer, 2013, pp. 374–391.

[13] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Proc. IACR Cryptol. ePrint Archive*, P. Q. Nguyen, Ed., 2012, pp. 296–312.

[14] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in *Advances in Cryptology—CRYPTO*, A. Menezes, Ed. Berlin, Germany: Springer, 2007, pp. 535–552.

[15] S. Narayan, M. Gagné, and R. Safavi-Naini, "Privacy preserving EHR system using attribute-based infrastructure," in *Proc. ACM Cloud Comput. Secur. Workshop (CCSW)*, New York, NY, USA, 2010, pp. 47–52.

[16] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in *Proc. 18th ACM Conf. Comput. Commun. Secur. (CCS)*, J. Katz, Ed. New York, NY, USA, 2011, pp. 491–500.

[17] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proc. 5th ACM Symp. Inf. Comput. Commun. Secur. (ASIACCS)*, Apr. 2010, pp. 261–270.

[18] 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, Apr. 2019.

[19] T. Jiang, X. Chen, Q. Wu, J. Ma, W. Susilo, and W. Lou, "Secure and efficient cloud data deduplication with randomized tag," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 532–543, Mar. 2017.

[20] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, A. Juels, R. N. Wright, and S. D. Vimercati, Eds. New York, NY, USA, Oct. 2006, pp. 89–98.

[21] X. Xie, H. Ma, J. Li, and X. Chen, "An efficient ciphertext-policy attribute-based access control towards revocation in cloud computing," *J. Universal Comput. Sci.*, vol. 19, no. 16, pp. 2349–2367, Oct. 2013.

[22] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 3494, R. Cramer, Ed. Heidelberg, Germany: Springer-Verlag, 2005, pp. 457–473.

[23] X. Yang, R. Lu, J. Shao, A. Ghorbani, and X. Tang, "Achieving efficient and privacy-preserving multi-domain big data deduplication in cloud," *IEEE Trans. Services Comput.*, to be published.

[24] Q. Li, J. Ma, R. Li, X. Liu, J. Xiong, and D. Chen, "Secure, efficient and revocable multi-authority access control system in cloud storage," *Comput. Secur.*, vol. 59, pp. 45–59, Jun. 2016.

[25] Y. Zhang, A. Wu, and D. Zheng, "Efficient and privacy-aware attribute-based data sharing in mobile cloud computing," *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 4, pp. 1039–1048, Aug. 2018.

[26] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, "HealthDep: An efficient and secure deduplication scheme for cloud-assisted eHealth systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4101–4112, Sep. 2018.

[27] Y. Zhou, D. Feng, Y. Hua, W. Xia, M. Fu, F. Huang, and Y. Zhang, "A similarity-aware encrypted deduplication scheme with flexible access control in the cloud," *Future Gener. Comput. Syst.*, vol. 84, pp. 177–189, Jul. 2018.

[28] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography—PKC* (Lecture Notes in Computer Science), vol. 6571. D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds. Heidelberg, Germany: Springer, 2011, pp. 53–70.

[29] S. Wang, D. Zhang, Y. Zhang, and L. Liu, "Efficiently revocable and searchable attribute-based encryption scheme for mobile cloud storage," *IEEE Access*, vol. 6, pp. 30444–30457, 2018.

[30] J. Blasco, R. Di Pietro, A. Orfila, and A. Sorniotti, "A tunable proof of ownership scheme for deduplication using Bloom filters," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2014, pp. 481–489.

[31] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.

[32] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. 20th USENIX Conf. Secur.*, Aug. 2011, p. 34.

[33] J. H. Seo and K. Emura, "Revocable identity-based cryptosystem revisited: Security models and constructions," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 7, pp. 1193–1205, Jul. 2014.

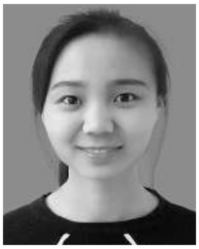
[34] S. Park, K. Lee, and D. H. Lee, "New constructions of revocable identity-based encryption from multilinear maps," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 8, pp. 1564–1577, Aug. 2015.

[35] 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 *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, in Lecture Notes in Computer Science, vol. 10892, B. Preneel and F. Vercauteren, Eds., 2018, pp. 516–534.

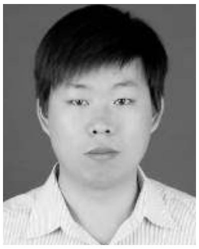
[36] B. Lynn, *The Stanford Pairing Based Crypto Library*. Accessed: Feb. 5, 2019. [Online]. Available: https://crypto.stanford.edu/pbc/

[37] D. Naor, M. Naor, and J. B. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 2139, J. Kilian, Ed. Cham, Switzerland: Springer, 2001.

HUA MA received the B.S. and M.S. degrees in mathematics from Xidian University, Xi'an, China, in 1985 and 1990, respectively, where she is currently a Professor. She has published more than 30 papers in refereed international conferences and journals. Her research interests include applied cryptography and cloud computing security.



YING XIE received the B.S. degree from Taiyuan Normal University, in 2017. She is currently pursuing the master's degree in mathematics with Xidian University. Her research focuses on network and information security.



JIANFENG WANG received the M.S. and Ph.D. degrees in mathematics and cryptography from Xidian University, in 2013 and 2016, respectively, where he is currently a Lecturer. His research interests include applied cryptography, cloud security, and database outsourcing. He is a member of ACM.

GUOHUA TIAN received the B.S. degree from the School of Mathematics and Information Science, Shaanxi Normal University, in 2016. He is currently pursuing the master's degree in mathematics with Xidian University. His research focuses on network and information security.

ZHENHUA LIU received the B.S. degree from Henan Normal University, in 2000, and the M.S. and Ph.D. degrees from Xidian University, China, in 2003 and 2009, respectively, where he is currently a Professor. His research interests include cryptography and information security.