Received April 29, 2019, accepted May 15, 2019, date of publication May 20, 2019, date of current version June 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917772

Multi-Keyword Searchable and Data Verifiable Attribute-Based Encryption Scheme for Cloud Storage

JIN SUN, LILI REN, SHANGPING WANG, AND XIAOMIN YAO
School of Science, Xi'an University of Technology, Xi'an 710054, China

Corresponding author: Lili Ren (liren0608@163.com)

This work was supported in part by the National Natural Science Youth Foundation of China under Grant 61303223, and in part by the National Natural Science Foundation of China under Grant 61572019.

ABSTRACT In a data sharing system, it is a basic requirement for a user, who has an appropriate privilege to perform keyword retrieval for encrypted documents stored in the cloud. Although traditional searchable encryption technology can provide data protection and retrieval characteristic, there are some main issues should also be considered. First, most existing attribute-based searchable encryption schemes only support single-keyword search, which may return abundant irrelevant search results, resulting in a waste of computational and broadband resources. Second, the user often needs to seek some data related to some particular keywords but his attributes may be altered frequently. Third, the cloud server is not completely loyal which sometimes returns a fraction of erroneous search results. Focus on these issues, a practical multi-keyword searchable encryption scheme is proposed for data integrity verification and attribute revocation by combining the ciphertext policy attribute-based encryption (CP-ABE) and auditing ideas. The scheme on one hand supports multi-keyword search which avoids the cloud server yield ample irrelevant documents by narrowing the search scope, and the other hand can implement effectively attribute revocation by entrusting ciphertext updates to the powerful cloud server, thereby preventing access by illegal users. Furthermore, third-party audits use verification algorithms to ensure the correctness of search results and reduce the amount of computing by end users. The most critically, the scheme proved to be resistant to selective plaintext attacks and selective keyword attacks under the general group model. The extensive experimental results demonstrate that the scheme is more expressive, efficient, and feasible in the practical applications.

INDEX TERMS Access control, keywords search, data verification, attributes revocation.

I. INTRODUCTION

Nowadays, with cloud computing [1] developing, cloud storage is an emerging storage technology for users to store and access data. A mass of enterprises and individuals outsource shared their data to these cloud platforms for store and manage. However, at the same time as bringing convenience, cloud storage is along with the security problem of data and the risk of user privacy disclosure. This is owing to the fact that when a large amount of data stored in the cloud server is away from the data owners' physical control. In addition, data stored in the cloud inevitably leads to data security problems, such as data are falsified, lost, illegally accessed or stolen.

So, how to guarantee the security and confidentiality of the data? One of the traditional ways to enhance file privacy is to encrypt the data before outsourcing it onto a cloud server. Then, Waters and Sahai [2] first introduced a new attribute-based encryption (ABE) method, that is considered to be an efficient encryption medium with fine-grained access control in cloud storage. However, when a number of encrypted data is stored on the cloud server, it will make the retrieval over encrypted data extremely difficult. In this case, the most straightforward solution is that the user downloads encrypted data sets from the cloud and decrypts them locally before searching. Obviously, this method not only increases

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

the communication overhead, but also accumulates the computational cost of the client. So, it is not feasible in the case of large data size, insufficient network bandwidth, and weak client computing power. Thus, how to search encrypted files is a problem in practice.

Searchable encryption [3]–[5] technology enables users to retrieve over ciphertexts via keywords and search data of their interest. On this basis, a large number of public key encryption schemes based on keyword search [6],[7] are proposed. Most of these schemes not support multiple keyword search, but the strength of single keyword search is not enough which would output a lot of irrelevant search data and cause low search efficiency. Therefore, the research of efficient multi-keyword searchable encryption scheme has high theoretical value and practical significance.

However, the cloud server is not completely trusted and it is curious about some sensitive information in real-world applications. Although it will perform the search operation correctly, there are some dishonest behaviors: 1) tamper with the received encrypt data, 2) collude with malicious users to send wrong search results, 3) save more storage space to earn more benefits, delete part of the encrypted data, and then fake the relevant search result when the user needs it. Hence, it is required for a scheme supporting keyword search verification to ensure the accuracy of the search data.

Besides, in searchable encryption schemes for fine-grained access control, CP-ABE technology is suitable for granting access rights based on user attributes rather than user lists. Thus, during the in-depth study of CP-ABE, attribute revocation mechanism has attracted more and more attention. Especially in practical applications, due to the change of user rights, the system to manage the attributes according to actual needs is required. Since multiple users can share an attribute, once any attribute of the shared is revoked, it will inevitably affect other users who have the revocation attribute. In the traditional solution, the data needs to be re-encrypted and the user key needs to be updated [8]–[11]. However, the disadvantage of this method is that a large amount of computational overhead is required when a lot of data is saved at the cloud server or lots of users exist in the system. Therefore, attribute revocation is a challenging problem.

Based on the above problems, in the cloud storage, the existing ABE schemes implement verifiable keyword search or effective attribute revocation independently, and there is no solution that simultaneously solves the above problems efficiently and simply. Because the system parameters between a keyword searchable program and an attribute revocable program are different, and their concerns are usually different too, it is difficult to directly integrate existing technologies. Furthermore, even though firsthand combination is possible, it may lead to the redundancy of certain parameters. But in fact, these problems need to be solved at the same time to meet the actual needs in the cloud environment. Therefore, it is necessary to construct a verifiable, multi- keyword searchable and attribute revocable ABE scheme to handle data access in cloud storage and to ensure data security in cloud environment. Aim to address these issues, in this paper, we firstly probe verifiable multi-keyword search via a trusted third party, and then to discuss effective attribute revocation method. As a further illustrate, we evaluate the scheme's performance by using the real-world datasets, in which it shows our paper more feasible and more efficient in practice.

Our contributions: In the scheme, we proposed a multi-keyword search scheme which supports data integrity verification and attribute revocation simultaneously by utilizing CP-ABE technique and signature algorithm. The key contributions of our scheme can be summarized as follows:

1) We proposed a multi-keyword searchable encryption technique that allows legitimate users to retrieve ciphertext based on some specified keywords. At the same time, it is guaranteed that the attacker cannot obtain the keyword information from the user query through the keyword ciphertext.
2) We introduced a trusted third-party entity, and then use the signature mechanism to verify the integrity of the search results returned by the unauthentic cloud server.
3) We proposed an efficient instant attribute revocation method, which only needs to update the key and ciphertext associated with the revocation attribute during the entire revocation period. In addition, the attribute revocation algorithm revokes a certain attribute of the user and does not affect other users' attribute.
4) Simulation experiments based on actual data sets show that the paper is efficient and feasible in practical application scenarios.

II. RELATED WORK

A. CIPHERTEXT-POLICY ATTRIBUTE-BASED ENCRYPTION

Attribute-based encryption (ABE)introduced by Ali *et al.* [1] is considered an expansion of identity-based encryption (IBE) [12] by treating identity as a set of attributes. It is a flexible and versatile solution for fine-grained access control of encrypted data in cloud computing. So far, there are two types of ABE schemes, namely ciphertext policy attribute-based encryption (CP-ABE) and key policy attribute-based encryption (KP-ABE), which are proposed by Goyal *et al.* [13]. The CP-ABE scheme applies to the ciphertext is related with the access policy, and the user's key is related with the attributes. The user can decrypt the ciphertext only if the user's attribute set satisfies the ciphertext access control. The KP-ABE scheme is the opposite of CP-ABE.

In this paper, we used the CP-ABE mechanism, since the CP-ABE mechanism more applicable to the current cloud storage system. To this day, a lot of CP-ABE schemes offered secure data access policy, but these proposals are still flawed when used in reality. In 2013, Hur [14] introduced an CP-ABE data sharing method which utilized a less efficient tree access structure to complete user revocation through proxy encryption, and the solution was short of data integrity verification. In 2015, Jian *et al.* [15] based on CP-ABE constructed a hybrid encryption and data access control scheme, it was proved that the scheme is flexible and efficient, but the scheme did not support keyword searching feature. Su *et al.* [16] introduced a practical searchable CP-ABE scheme in cloud storage in 2017, the scheme protect the data privacy and user's queries privacy, but the solution lack of data integrity verification. In 2018, Cui *et al.* [17] put forward a CP-ABE with partially hidden access structures, whereas in the scheme the user's attribute revocation and data integrity verification remains to be studied. And then more CP-ABE schemes are available in the literature [18]–[21].

B. KEYWORD SEARCH ON THE ENCRYPTED DATA

In order to facilitate retrieval on encrypted data, Song *et al.* [3] first proposed the concept of searchable encryption. After that, the public key encryption with keyword search scheme was presented by Boneh *et al.* [4], where the data owner first uploads the encrypted file and the related keyword index to the cloud server, then the user generates a search token according to the private key and the keyword of interest, and then he sends the token to the cloud server. The cloud server uses the received token to search for data of interest for the user and returns the search results to the user. Wang *et al.* [22] introduced a keyword search and attribute revocation scheme in 2016, which provides keyword search function and supports user's multiple attributes revocation. In 2017, Qiu *et al.* [23] put forth a keyword search scheme for hidden policy, and proved that the scheme is secure under the general group model. In 2018, Yin *et al.* [24] proposed a searchable CP-ABE scheme. Because majority of the schemes [25]–[27] can't support multi-keyword search, to compensate this, amount multi-keyword search [28]–[33] schemes have been constructed. In 2016, Li *et al.* [28] presented a fine-grained multi-keyword search scheme over encrypted cloud storage data, which enhance better users' practice and more accurate search. In 2017, a multi-keyword multi-sever searchable encryption program based on cloud storage was put forward by Huang *et al.* [29], which is demonstrated to be secure resist adaptive chosen keyword challenge. However, none of the above schemes support data integrity verification, some even do not countenance user attribute revocation.

C. VERIFIABLE KEYWORD SEARCH

In real-world applications, the cloud server is usually regarded as a curious and semi-honest entity. In other words, the cloud server sometimes only perform a few search tasks and output a small number of incorrect search data to the user. So it is necessary to perform keyword search verification. In recent years, some verifiable keyword search schemes are proposed [34]–[36]. In 2015, Zheng *et al.* [35] introduced a verifiable CP-ABKS searchable encryption scheme which use different access control policies to encrypt different keywords during the encryption process. In the process of the execution of the verification algorithm by the server, different return information is generated for different access policies, and the user can determine whether the server strictly performs the verification algorithm according to the returned information. The shortcomings of this scheme are that the adoption of a less efficient tree access structure, the search operation is related to the attribute and the attribute revocation function is not implemented. In 2016, Sun *et al.* [36] proposed a plausibility check of search results and a fine-grained authorization scheme, which resulted in a large amount of computational overhead as the amount of attributes increasing in the scheme. Although the above scheme achieves a verifiable keyword search to some extent via using a bloom filter, the search result would not be exactly verified because of the bloom filter's high false positive rate. Moreover, these proposals cannot be widely applied.

TABLE 1. Symbols table.

Symbols	Description
PK	public key
MK	master key
PAK_x	public attribute key
uid	the identity of DU
RL_x	the attribute revocation list set of users
S_{uid}	attributes set of the DU
SK_u	the private key of the DU
pk_0, sk_0	public-private key pair of the DO
k	content secret key
I_W	the index of the keywords
$T_{W'}$	search token
CT	ciphertext
Sig	signature set
$AUK_{x'}$	attribute update key

D. ATTRIBUTE REVOCATION BASED ON ATTRIBUTE ENCRYPTION

In recent years, revocation mechanism has been paid more and more attention. In the basic ABE data sharing system, it is great important to implement an efficient attribute revocation mechanism. Then, attribute revocation based on attribute encryption schemes [22], [37], [38] with different features have been presented. In 2016, Sun *et al.* [36] used proxy re-encryption technology to construct a scheme with verifiable keyword seek and support user attribute revocation, but the computational cost are huge. In the same year, Wang *et al.* [39] constructed an attribute encryption scheme with two attribute revocation lists, which is a promotion of a scheme containing a single attribute list, yet it has no keyword search function. In 2018, Liang *et al.* [40] introduced a KP-ABE scheme with attribute revocation, which achieved fine-grained data access and ensured data deletion. However, the scheme utilized AND Gate's access structure and cannot perform keyword search.

III. PRELIMINARIES

In this section, we briefly describe some of the background knowledge used in this article. Table. 1 lists some of the symbols used in this paper.

A. BILINEAR PAIRING [41]

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. Let g one generator of $\mathbb{G}$. A bilinear pairing operation constructed as with the following properties:

1) Bilinearity: For $\forall u, v \in \mathbb{G}$ and all $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$ holds.
2) Non-Degeneracy: $e(g, g) \neq 1$.
3) Computability: For all $\forall u, v \in \mathbb{G}$, $e(u, v)$ can be efficiently computed.

B. ACCESS STRUCTURE

In an ABE scheme, in order to achieve fine-grained access control, the following access policy structure [41] is described.

Let $P = \{P_1, P_2, \cdots, P_n\}$ be a set of n parties. For $\forall B, C$, if $B \subseteq \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$, we say $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}}$ is monotonous. An access structure is a collection $\mathbb{A}$ of non-empty subsets of $P = \{P_1, P_2, \cdots, P_n\}$, namely $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}}\{\emptyset\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets.

C. LINEAR SECRET SHARING SCHEME (LSSS)

Let $P = \{P_1, P_2, \cdots, P_n\}$ be a set of parties. $(\mathbf{M}, \rho)$ indicates an access policy $\mathbb{A}$, where $\mathbf{M}$ is a shared generator matrix with l rows and n columns. For all $1 \leq i \leq l$, $\rho(i)$ is a function which maps the i row of $\mathbf{M}$ to the different attributes. The following two parts form a linear secret sharing scheme [13]:

1) Let $s \in \mathbb{Z}_p$ be a secret, and $r_2, \cdots, r_n \in \mathbb{Z}_p$ be randomly selected. The column vector $\mathbf{v} = (s, r_2, \cdots, r_n)$, then computes $\lambda_i = (\mathbf{M} \cdot \mathbf{v})_i$, where λ_i is the sub-secret share value of the party $\rho(i)$.
2) Let $S \in \mathbb{A}$ be any authorized set, and $I \subset \{1, 2, \cdots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ satisfie $\sum_{i \in I} \omega_i \mathbf{M}_i = (1, 0, \cdots, 0)$. The recovered secret will be $\sum_{i \in I} \omega_i \lambda_i = s$. These constants $\{\omega_i\}$ can be found in polynomial time.

D. GENERIC BILINEAR GROUP [42]

Let ϕ_1, ϕ_2 be two random encodings, $\mathbb{Z}_p^+ \rightarrow \{0, 1\}^{\mathcal{K}}$ where $\mathbb{Z}_p^+$ is an addition group, satisfied $\mathcal{K} > 3log(p)$. The groups $\mathbb{G}$, $\mathbb{G}_T$ are described as $\mathbb{G} = \{\phi_1(\mathcal{X}) | \mathcal{X} \in \mathbb{Z}_p\}$, $\mathbb{G}_T = \{\phi_2(\mathcal{X}) | \mathcal{X} \in \mathbb{Z}_p\}$, where $\mathbb{G}$ represents a generic bilinear group, g represents $\phi_1(1)$, $g^{\mathcal{X}}$ denotes $\phi_1(\mathcal{X})$, $e(g, g)$ represents $\phi_2(1)$ and $e(g, g)^{\mathcal{X}}$ denotes $\phi_2(\mathcal{X})$.

E. CRYPTOGRAPHIC ASSUMPTION [43]

Let $a, s, b_1, \cdots, b_q$ are randomly chosen from $\mathbb{Z}_p$, and g is the generator of $\mathbb{G}$. Given

$$\begin{aligned} \mathbf{y} = (&g, g^s, g^a, \cdots, g^{a^q}, , g^{a^{q+2}}, \cdots, g^{a^{2q}}, \\ &\forall_{1 \leq j \leq q} \; g^{sb_j}, g^{a/b_j}, \cdots, g^{a^q/b_j}, , g^{a^{q+2}/b_j}, \cdots, g^{a^{2q}/b_j}, \\ &\forall_{1 \leq k, j \leq q, k \neq j} \; g^{asb_k/b_j}, \cdots, g^{a^q sb_k/b_j}). \end{aligned}$$

It is difficult to distinguish an effective tuple $e(g, g)^{a^{q+1}s} \in \mathbb{G}_T$ from a random element $R \in \mathbb{G}_T$.

An algorithm $\mathcal{B}$, which returns $z \in \{0, 1\}$, has an advantage ϵ in solving the q-parallel BDHE problem in $\mathbb{G}$ if

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

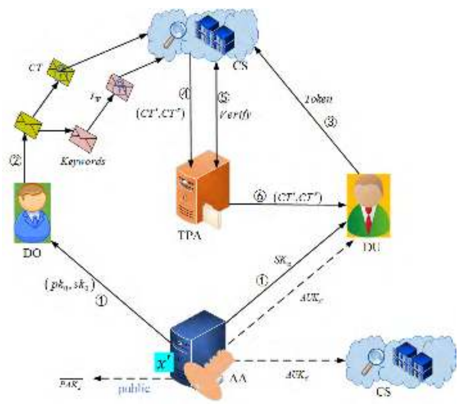

FIGURE 1. System model.

Decisional q-Parallel Bilinear Diffie-Hellman Exponent assumption (q-parallel BDHE) be described as: if no probability polynomial time adversary with a non-negligible advantage to solve the decisional q-parallel BDHE problem, we say the assumption holds.

IV. SYSTEM AND SECURITY MODEL

A. SYSTEM MODEL

Our scheme comprises five entities: Trusted Third Party Audit (TPA), Cloud Server (CS), Attribute Authority (AA), Data Owners (DO) and Data Users (DU). Fig. 1 shows the system model of our scheme.

1) CLOUD SERVER (CS)

The Cloud Server (CS) has sufficient storage capacity and computing power which supply storage and data retrieval services, but it is not completely trusted since it is usually offered by the third parties. Upon receiving the user's access request, the CS executes the seek task using the received search token. Once the keyword index in some data files matches the submitted search token, the CS returns corresponding ciphertext to the TPA for verification of the search result. Besides, the CS is also in charge of some calculating operations for the revocation phase, for instance ciphertext updates.

2) TRUSTED THIRD PARTY AUDIT (TPA)

A TPA provides a result verification service that verifies whether the data returned from the cloud server is completely correct because DO cannot fully control its remote data.

3) ATTRIBUTE AUTHORITY (AA)

Attribute authority is completely trustworthy. It is in charge of the system setup and the registration of data users. In addition, AA takes charge of attribute revocation based on the data user's dynamic role.

4) Data Owners (DO)

The data owner is responsible for encrypting shared data and keyword sets. He first runs the keyword index algorithm and then encrypts the shared data with a symmetric encryption algorithm. In addition, DO encrypts the content key under its own defined access control. Finally, DO uploads the relevant ciphertexts to the CS.

5) Data Users (DU)

Each DU has a private key related to his attributes set. They can generate corresponding search token by applying keyword, and then exploit the token to find relevant data on the cloud server. Once the search content is received from the TPA, if the data user's attributes meet the access policy in the ciphertexts, the ciphertexts can be decrypted.

B. SYSTEM ARCHITECTURE

Our scheme contains eleven algorithms as follows:

- ***Setup***$(1^{\mathcal{K}}) \rightarrow (PK, MK)$*:* The AA runs the setup algorithm. It intakes a security parameter $\mathcal{K}$, and returns the public key PK and master key MK.
- ***KeyGen***$(MK, S_{uid}) \rightarrow \{SK_u, (pk_0, sk_0)\}$*:* The AA runs private key generation algorithm. It intakes a master key MK and an attributes set S_{uid} of the DU, then returns the private key SK_u of the DU and the public-private key pair (pk_0, sk_0) of the DO.
- ***IndexGen***$(PK, W) \rightarrow I_W$*:* The DO runs keywords index generation algorithm. It intakes the public key PK and the keywords set W, then returns the index I_W of the keywords.
- ***Encrypt***$(PK, k_\theta, \mathbb{A}) \rightarrow (CT, sig_\theta)$*:* The DO runs this algorithm. It intakes the public key PK, the content key k_θ and the access policy $\mathbb{A}$, then outputs the ciphertexts set CT and the signature sig_θ.
- ***Token***$(PK, W', SK_u) \rightarrow T_{W'}$*:* The DU runs the token generation algorithm. It inputs the public key PK, the keyword set W' to be searched and the private key SK_u, then outputs the search token $T_{W'}$.
- ***Search***$(T_{W'}, I_W)$*:* The CS runs search algorithm to verify whether the token $T_{W'}$ submitted by the DU matches the keyword index I_W uploaded by the DO. If they match, CS returns the relevant search result to TPA, otherwise it returns symbol $\perp$.
- ***Verify***$(sig_\theta, C_\theta, PK, pk_0)$*:* The TPA runs the verification algorithm. After receiving the search result, TPA checks the correctness of the returned result by interacting with the CS, and then returns the corresponding ciphertexts to the DU.
- ***Decrypt***$(SK_u, CT', C_\theta) \rightarrow k_\theta$*:* The DU runs the decryption algorithm. It inputs the corresponding ciphertexts and private key. If his attributes set satisfies the access policy embedded in the ciphertext, then outputs the corresponding content key k_θ, otherwise returns $\perp$.
- ***UKeyGen***$(PK, MK, x') \rightarrow (AUK_{x'}, \overline{PAK_{x'}})$*:* The AA runs the update key generation algorithm. It intakes the public key PK, the master key MK and the revoked attribute x', then output attribute update key $AUK_{x'}$ and the updated public attribute key $\overline{PAK_{x'}}$. Finally, the attribute update key $AUK_{x'}$ and the attribute revocation list $RL_{x'}$ are sent to the user whose attribute x' was canceled and the CS, aim to update the relevant private key and ciphertext respectively.
- ***KeyUpdate***$(SK_u, AUK_{x'}) \rightarrow SK'_u$*:* The DU whose attribute x' was revoked runs the private key update algorithm. It intakes the current private key SK_u and attribute updated key $AUK_{x'}$, then returns a new private key SK'_u.
- ***CTUpdate***$(CT', AUK_{x'}) \rightarrow CT''$*:* The CS runs the ciphertext update algorithm. It inputs the ciphertext CT' associated with the revoked attribute x' and attribute update key $AUK_{x'}$, then outputs an updated ciphertext CT''.

C. SECURITY MODEL

The security of our scheme relies on the general bilinear group model and the cryptographic assumption. The paper designed two security games to prove the security of our scheme, which including the keyword privacy game and the indistinguishability against selective ciphertext-policy and chosen plaintext attack (IND-sCP-CPA) game.

1) KEYWORD PRIVACY GAME

Setup: Challenger $\mathcal{B}$ runs the setup algorithm to generate the public key PK and master key MK, then sends the public key PK to adversary $\mathcal{A}$, and holds master key MK privately.

Phase 1: $\mathcal{A}$ would iterate asked the $\mathcal{O}_{SK_u}$ and $\mathcal{O}_{Token}$ oracles. $\mathcal{B}$ maintains a keyword set catalogue $\mathcal{L}_W$ which initial value is empty.

- $\mathcal{O}_{SK_u}(PK, MK)$*:* It takes as public key PK and the master key MK owned by $\mathcal{B}$, then $\mathcal{B}$ returns the corresponding private key SK_u to adversary.
- $\mathcal{O}_{Token}(PK, SK_u, W^*)$*:* Give an asked keyword set W^* and the corresponding private key SK_u, $\mathcal{B}$ generates a search token before issuing it to $\mathcal{A}$, then $\mathcal{B}$ adds the keyword set W^* to the keyword set catalogue $\mathcal{L}_W$.

Challenge: $\mathcal{A}$ sends two equal-length keyword set W_0 and W_1, where $W_0, W_1 \notin \mathcal{L}_W$, $\mathcal{B}$ first chooses a random bit $b \in \{0, 1\}$ and generates an index I_{W_b} of the keyword set W_b, and then sends it to $\mathcal{A}$. The restriction for $W_0, W_1 \notin \mathcal{L}_W$ is that $\mathcal{A}$ cannot guess bits b from $\mathcal{O}_{Token}$ oracle.

Phase 2: $\mathcal{A}$ submits the same inquiry as **Phase1**, but the only restraint is the keyword set W_0, W_1 cannot be asked to $\mathcal{O}_{Token}$ oracle.

Guess: $\mathcal{A}$ sends a guess bit $b' \in \{0, 1\}$. The $\mathcal{A}$ wins if $b = b'$.

Therefore, the advantage of $\mathcal{A}$ in breaking the keyword privacy game is described as $Adv_{\mathcal{A}}(1^{\mathcal{K}}) =| Pr[b' = b] - \frac{1}{2} |$. Then our scheme is keyword secure if all probability polynomial time adversary $\mathcal{A}$ have at most a negligible advantage in this security game.

2) IND-sCP-CPA GAME

Init: Adversary $\mathcal{A}$ determines an access structure $\mathbb{A}^*$ that he hopes to dare it.

Setup: Duplicate the above-mentioned security game's **Setup**.

Phase 1: $\mathcal{A}$ can repeatedly ask the $\mathcal{O}_{SK_u}$ and $\mathcal{O}_{AUK}$ oracles.

- $\mathcal{O}_{SK_u}(uid, S_{uid})$*:* $\mathcal{A}$ can sends queries for private key SK_u by submitting (uid, S_{uid}), where the restraint that S_{uid} does not satisfy the access structure $\mathbb{A}^*$.
- $\mathcal{O}_{AUK}(x')$*:* $\mathcal{A}$ can also do the attribute update key query by sending any attribute x'.

Challenge: $\mathcal{A}$ sends two equal-length messages k_{θ_0} and k_{θ_1}. Challenger $\mathcal{B}$ selects a random bit $b \in \{0, 1\}$ and encrypts k_{θ_b} under the access structure $\mathbb{A}^*$, then sends the ciphertext CT'^* to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ may ask more secret keys and update keys for other attribute sets. It is same as **Phase1**.

Guess: $\mathcal{A}$ sends a guess bit $b' \in \{0, 1\}$. The $\mathcal{A}$ wins if $b = b'$.

Therefore, the advantage of $\mathcal{A}$ in breaking the game is described as $Adv_{\mathcal{A}}^{IND-sCP-CPA}(1^{\mathcal{K}}) =\mid Pr[b' = b] - \frac{1}{2} \mid$. Then our scheme is IND-sCP-CPA secure if all probability polynomial time adversary $\mathcal{A}$, there is at most a negligible advantage in this security game.

V. A CONCRETE CONSTRUCTION

In this part, we will introduce a specific multiple keyword searchable encryption scheme that support data integrity verification and attribute revocation.

A. SYSTEM INITIALIZATION

Setup($1^{\mathcal{K}}$) →(PK,MK): AA chooses two bilinear groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p. $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a bilinear map and g, g_0 are generators of $\mathbb{G}$. Let $H : \{0, 1\}^* \rightarrow \mathbb{G}$, $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$ are two hash functions. It first outputs a global public parameter $\mathcal{GP} = \{\mathbb{G}, \mathbb{G}_T, H, H_1, e, p, g, g_0\}$. AA defines U as attribute universe, selects a random element $v_x \in \mathbb{Z}_p$ for each attribute $x \in U$, and calculates the public attribute key $PAK_x = g^{v_x}$. Then, it selects random numbers $a, \alpha, \beta \in \mathbb{Z}_p^*$, the public key and the master key are published as

$$PK = \{\mathcal{GP}, g^a, g^\alpha, e(g, g)^\beta, \quad \{PAK_x \mid x \in U\}\},$$
$$MK = \{\alpha, g^\beta, \{v_x \mid x \in U\}\}.$$

Finally, for each attribute $x \in U$, RL_x denote the attribute revocation list.

B. KEY GENERATION

KeyGen(MK,S_{uid})→{SK_u, (pk_0, sk_0)}: The AA runs key generation algorithm. When a DU adds in the system, AA distributes an identity uid and attributes S_{uid} to him. Then it selects $t \in \mathbb{Z}_p^*$, calculates $D = g^\beta \cdot g^{at}$, $D' = g^t$, $D'' = g^{a\alpha}$, and for $\forall : x \in S_{uid}$ calculates $D_x = H(x)^{t/v_x}$. The algorithm outputs the private key of the DU as:

$$SK_u = (D, D', D'', D_x).$$

After that, AA randomly chooses $r' \in \mathbb{Z}_p^*$ and calculates $g^{r'}$, then outputs the DO's public/secret key pair (pk_0, sk_0), where $pk_0 = g^{r'}$, $sk_0 = r'$.

C. DATA ENCRYPTION

Step 1 (Keywords index generation):

IndexGen(PK,W)→ I_W: Given the file set $F = (F_1, F_2, \cdots, F_d)$, where d indicates the number of files. DO first extracts keywords $W = \{w_1, w_2, \cdots, w_m\}$ from certain file $F_\theta (1 \leq \theta \leq d)$ and constructs index for it, where m indicates the number of keywords in the keyword set. For the keyword $w_j \in W$, DO selects $\sigma \in \mathbb{Z}_p^*$ and computes $\bar{I} = g^{\alpha\sigma}$, where $1 \leq j \leq m$. Then, he chooses τ_θ for F_θ and computes $I_{\theta,j} = g^{a(\sigma+\tau_\theta)} g^{\sigma H_1(w_j)}$, $I'_\theta = g^{\tau_\theta}$, he sets the index for the keyword w_j as

$$\{Iw_j\} = (\bar{I}, \{I_{\theta,j}, I'_\theta\}).$$

Finally, the DO generations the index for the keyword set W as $I_W = \{Iw_j\}_{1 \leq j \leq m}$.

Step 2 (File encryption):

To encrypt F_θ, the DO uses the content secret key k_θ (the k_θ is randomly selected in secret key space) and a symmetric encryption algorithm to obtain ciphertexts $C_\theta = Enc_{k_\theta}(F_\theta)$, then DO encrypts k_θ by the following encryption algorithm.

Encrypt(PK,k_θ,(M,ρ)) → CT: Algorithm inputs PK, k_θ and an access policy $(\mathbf{M}, \rho)$, where $\mathbf{M}$ is a matrix of $l \times n$, and function ρ maps each row of $\mathbf{M}$ to different attribute. It selects random values $y_2, \cdots, y_n \in \mathbb{Z}_p$ as the components of the vector $\mathbf{v} = (s, y_2, \cdots, y_n)$. For $1 \leq i \leq l$, the shares of the secret s are calculated as $\lambda_i = \mathbf{M}_i \cdot \mathbf{v}$, where $\mathbf{M}_i$ denotes the vector composed by the i-th row of $\mathbf{M}$. Then, algorithm randomly selects $r_1, \cdots, r_l \in \mathbb{Z}_p$ and calculates the ciphertexts as

$$CT' = \{(\mathbf{M}, \rho), \widetilde{C} = k_\theta \cdot e(g, g)^{\beta s}, \overline{C} = g^s,$$
$$\forall i \in 1, \cdots, l : \hat{C}_i = (g)^{a\lambda_i} H(\rho(i))^{r_i}, \ \check{C}_i = (g^{v_{\rho(i)}})^{r_i}\}.$$

In addition, the DO generates the signature $sig_\theta = (H(id_\theta) \cdot g_0^{H_1(C_\theta)})^{r'}$ for file $F_\theta (1 \leq \theta \leq d)$, where id_θ is the identity of the file F_θ. Finally, the DO uploads the ciphertexts $CT = (I_W, C_\theta, CT',)$ and the signature sig_θ to the CS.

D. KEYWORD SEARCH

Step 1 (Token generation):

Token(PK,W',SK_u) → $T_{W'}$: When the DU releases a search ask for the keyword set $W' = (w'_1, w'_2, \cdots, w'_{m'})$, it chooses $\pi \in \mathbb{Z}_p^*$ and calculates $t_1 = g^{a\pi} \cdot \prod_{j=1}^{m'} g^{\pi H_1(w'_j)}$, $t_2 = g^{\alpha\pi}$, $t_3 = D''^{\pi}$, where m' indicates the number of keyword asked by the DU. Then, DU sets the search token as

$$T_{W'} = (t_1, t_2, t_3).$$

Finally he sends $T_{W'}$ to CS.

Step 2 (Search data):

Search($T_{W'}, I_W$): When obtaining a search query of the DU, CS verifies whether the submitted token $T_{W'}$ matches some of the keyword indexes $Iw_j \in I_W$ by the equation

$$e(\prod_{j=1}^{m'} I_{\theta,j}, t_2) = e(\bar{I}, t_1)e(I'_\theta, t_3).$$

In terms of the above equation, it is a matching of the token with the index. In the keyword index generation process, the DO encrypts m keyword to get $\{Iw_j\}_{j\in[1,m]}$. In the token generation process, the DU releases a search ask for m' keyword and generation a token $T_{W'}$, particularly $m' \leq m$. When the CS obtained token $T_{W'}$, in order to the above equation holds, the CS randomly choose m' index from $\{Iw_j\}_{j\in[1,m]}$ and perform multiplication operations. According to the mathematical statistics and probability theory, the total number of random selections is $C_m^{m'} = \frac{m(m-1)(m-2)\cdots(m-m'+1)}{m'!}$. Then, CS matches the multiplication of index $\{Iw_j\}_{j\in[1,m]}$ with token $T_{W'}$. As long as there is one successful match in the $C_m^{m'}$ times matching, it proves that the search succeeds. If the search succeeds, the above equation holds. The CS returns search ciphertext $CT = \{I_W, C_\theta, CT'\}$ to TPA. Otherwise, returns $\perp$.

E. VERIFICATION

Verify(sig_θ,C_θ,PK,pk_0): After getting the related data C_θ, TPA chooses $\mu_\theta \in \mathbb{Z}_p^*$ for the ciphertext C_θ with the identity id_θ, and interacts with CS as follows:

1) TPA sents μ_θ to the CS;
2) CS calculates $\bar{\delta} = \mu_\theta H_1(C_\theta)$ and $\xi = (sig_\theta)^{\mu_\theta}$, where $sig_\theta = (H(id_\theta) \cdot g_0^{H_1(C_\theta)})^{r'}$. Then, CS sends $(\bar{\delta}, \xi)$ to TPA;
3) TPA checks the correctness of C_θ by equation

$$e(\xi, g) = e(H(id_\theta)^{\mu_\theta} \cdot g_0^{\bar{\delta}}, pk_0).$$

If the above equation holds, TPA returns tuple (CT', C_θ) to DU, otherwise it discards CT'.

F. FILE DECRYPTION

Decrypt(SK_u,CT',C_θ): The DU obtained a verified ciphertext (CT', C_θ) from the TPA. If the attribute of the DU is not in the revocation list RL_x, and the attribute in SK_u satisfies the access policy embedded in the ciphertext, DU can obtain a content key k_θ by running the following decryption algorithm. The algorithm be proceed as follows:

Let $I \subset \{1, 2, \cdots, l\}$ be described as $I = \{i : \rho(i) \in S_{uid}\}$ and $\{\lambda_i\}$ are valid shares of s, then according to the properties of LSSS, there exist a set of constants $\{\omega_i \in \mathbb{Z}_p\}_{i\in I}$ satisfying $\sum_{i\in I} \omega_i\lambda_i = s$. The decryption algorithm first calculates:

$$A = \frac{\prod_{i\in I} e(\hat{C}_i, D')^{\omega_i}}{\prod_{i\in I} e(\check{C}_i, D_{\rho(i)})^{\omega_i}} = e(g, g)^{ats}.$$

The DU computes k_θ as $k_\theta = \frac{\tilde{C}\cdot A}{e(D,\bar{C})}$ subsequently, and then decrypts the file with k_θ.

G. ATTRIBUTE REVOCATION

If an attribute x' is cancelled from a few data users, AA puts the identity of these data users to the attribute revocation set $RL_{x'}$. The specific way is as follows:

Step 1 (Update key generation):

UKeyGen(PK,MK,x')→ ($AUK_{x'}, \overline{PAK}_{x'}$): This algorithm intakes the public key PK, the master private key MK, the revoked attribute x'. For the revoked attribute x', AA selects a random element $\bar{v}_{x'} \in \mathbb{Z}_p^*(\bar{v}_{x'} \neq v_{x'})$, and then calculates the attribute update key by $AUK_{x'} = \frac{\bar{v}_{x'}}{v_{x'}}$. In addition, AA calculates the new public attribute key by using the attribute update key just updated as

$$\overline{PAK}_{x'} = (PAK_{x'})^{\frac{\bar{v}_{x'}}{v_{x'}}} = (PAK_{x'})^{AUK_{x'}}.$$

Finally, AA announces the new public attribute key $\overline{PAK}_{x'}$. Meanwhile, AA sends the attribute update key $AUK_{x'}$ and the attribute revocation list $RL_{x'}$ to the DU whose identity $uid \notin RL_{x'}$ and the CS, aiming to update the private key and ciphertext associated with the revoked attribute x'.

Step 2 (Update user private key):

KeyUpdate(SK_u,$AUK_{x'}$) → SK'_u: This algorithm intakes the user's private key SK_u and the attribute update key $AUK_{x'}$. Then, the user performs a private key update procedure and returns a new private key SK'_u as follows:

$$\begin{aligned} SK'_u &= \{\bar{D} = D, \bar{D'} = D', \bar{D''} = D'', \\ &for\ x \neq x' : \bar{D}_x = D_x, \\ &for\ x = x' : \bar{D}_x = (D_x)^{AUK_{x'}}\}. \end{aligned}$$

Step 3 (Ciphertexts update):

CTUpdate(CT',$AUK_{x'}$) → CT'': It inputs the ciphertext CT' related to the cancelled attribute x' and the attribute update key $AUK_{x'}$, then CS runs this algorithm returns updated ciphertext CT'':

$$\begin{aligned} CT'' = \{&\tilde{C}' = \tilde{C}, \bar{C'} = \bar{C}, \\ &\forall i \in 1, \cdots, l : \hat{C}'_i = \hat{C}_i, \\ for\ rho(i) \neq x' &: \check{C}'_i = \check{C}_i, \\ for\ \rho(i) = x' &: \check{C}'_i = (\check{C}_i)^{AUK_{\rho(i)}}\}. \end{aligned}$$

VI. CORRECTNESS AND SECURITY PROOFS

A. CORRECTNESS PROOF

1) Correctness of the keywords search:

Judge: $e(\prod_{j=1}^{m'} I_{\theta,j}, t_2) = e(\bar{I}, t_1)e(I'_\theta, t_3)$

Left side of the equation can represent as:

$$\begin{aligned} e(\prod_{j=1}^{m'} I_{\theta,j}, t_2) &= e(g^{a(\sigma+\tau_\theta)} g^{\sigma\sum_{j=1}^{m'} H_1(w'_j)}, \quad g^{\alpha\pi}) \\ &= e(g, g)^{\alpha\pi a(\sigma+\tau_\theta)} e(g, g)^{\alpha\pi\sigma\sum_{j=1}^{m'} H_1(w'_j)} \end{aligned}$$

Right side of the equation can represent as:

$$
\begin{aligned}
&e(\overline{I}, t_1)e(I'_\theta, t_3)\\
&= e(g^{\sigma\alpha}, g^{a\pi}\cdot\prod_{j=1}^{m'} g^{\pi H_1(w'_j)})e(g^{\tau_\theta}, g^{a\pi\alpha})\\
&= e(g,g)^{\sigma\alpha\pi\sum_{j=1}^{m'}H_1(w'_j)}e(g,g)^{\sigma\alpha a\pi}e(g,g)^{\tau_\theta\pi\alpha a}\\
&= e(g,g)^{\sigma\alpha\pi\sum_{j=1}^{m'}H_1(w'_j)}e(g,g)^{\alpha a\pi(\sigma+\tau_\theta)}
\end{aligned}
$$

2) Correctness of the verification phase:
Judge: $e(\xi, g) = e(H(id_\theta)^{\mu_\theta}\cdot g_0^{\overline{\delta}}, pk_0)$
Process:

$$
\begin{aligned}
e(\xi, g) &= e((sig_\theta)^{\mu_\theta}, g)\\
&= e(((H(id_\theta)\cdot g_0^{H_1(C_\theta)})^{r'})^{\mu_\theta}, g)\\
&= e(H(id_\theta)^{\mu_\theta}\cdot g_0^{\mu_\theta H_1(C_\theta)}, g^{r'})\\
&= e(H(id_\theta)^{\mu_\theta}\cdot g_0^{\overline{\delta}}, pk_0)
\end{aligned}
$$

3) The correctness of file decryption:

$$
\begin{aligned}
A &= \frac{\prod_{i\in I} e(\hat{C}_i, D')^{\omega_i}}{\prod_{i\in I} e(\check{C}_i, D_{\rho_i})^{\omega_i}}\\
&= \frac{\prod_{i\in I} e(g^{a\lambda_i}H(\rho(i))^{r_i}, g^t)^{\omega_i}}{\prod_{i\in I} e((g^{v_{\rho(i)}})^{r_i}, H(\rho(i))^{t/v_{\rho(i)}})^{\omega_i}}\\
&= \frac{\prod_{i\in I} e(g^{a\lambda_i}, g^t)^{\omega_i}\cdot\prod_{i\in I} e(H(\rho(i))^{r_i}, g^t)^{\omega_i}}{\prod_{i\in I} e(g^{r_i}, H(\rho(i))^t)^{\omega_i}}\\
&= e(g,g)^{at\sum_{i\in I}\omega_i\lambda_i}\\
&= e(g,g)^{ats}\\
k_\theta &= \frac{\widetilde{C}\cdot A}{e(D, \overline{C})} = \frac{k_\theta\cdot e(g,g)^{\beta s}\cdot e(g,g)^{ats}}{e(g^\beta\cdot g^{at}, g^s)}
\end{aligned}
$$

B. SECURITY PROOF

Theorem 1: Given a one-way hash function H_1. Our scheme is proved secure under the general bilinear group model, which can resist selectively chosen keyword attack.

Proof: In keyword privacy game, the target of $\mathcal{A}$ is to differentiate between $g^{a(\sigma+\tau_\theta)}g^{\sigma H_1(W_0)}$ and $g^{a(\sigma+\tau_\theta)}g^{\sigma H_1(W_1)}$. With inputting a number $f \in \mathbb{Z}_p$, the ascendancy of $\mathcal{A}$ in differentiating between g^f and $g^{a(\sigma+\tau_\theta)}g^{\sigma H_1(W_0)}$ is the same as that of differentiating between g^f and $g^{a(\sigma+\tau_\theta)}g^{\sigma H_1(W_1)}$. To simply expressed, let's assume that $\mathcal{A}$ can differentiate between g^f and $g^{a(\sigma+\tau_\theta)}$, and then a modified keyword privacy game between $\mathcal{A}$ and $\mathcal{B}$ as shown below.

Setup: $\mathcal{B}$ randomly chooses $a, \alpha \in \mathbb{Z}_p^*$ firstly and return a public key $PK = (g, g^\alpha, g^a)$ to $\mathcal{A}$, then $\mathcal{B}$ holds the master key $MK = \{\alpha\}$ privately.

Phase 1: $\mathcal{A}$ makes the private key and token queries to the oracles $\mathcal{O}_{SK_u}$ and $\mathcal{O}_{Token}$.

- $\mathcal{O}_{SK_u}(PK, MK)$: $\mathcal{B}$ calculates $D'' = g^{a\alpha}$ and $PK = (g, g^\alpha, g^a)$, then return D'' to $\mathcal{A}$.
- $\mathcal{O}_{Token}(PK, W^*, D'')$: Inputting PK, W^* and D'', $\mathcal{B}$ randomly chooses $\pi \in \mathbb{Z}_p^*$ and generates a search token $T_{W'} = (t_1, t_2, t_3)$ of the keyword set W^*, where $t_1 = g^{a\pi}\cdot\prod_{j=1}^{m'} g^{\pi H_1(w'_j)}$, $t_2 = g^{\alpha\pi}$, $t_3 = D''^{\pi}$, and then $\mathcal{B}$ adds the W^* to the keyword set catalogue $\mathcal{L}_W$.

Challenge: $\mathcal{B}$ inputting equal length keyword sets W_0 and W_1, where $W_0, W_1 \notin \mathcal{L}_W$. $\mathcal{B}$ selects $\sigma, \tau_\theta \in \mathbb{Z}_p^*$, and chooses a random bit $b \in \{0,1\}$, if $b = 0$, he outputs $\overline{I} = g^{\alpha\sigma}, I_{\theta,j} = g^f, I'_\theta = g^{\tau_\theta}$, otherwise, he returns $\overline{I} = g^{\alpha\sigma}$, $I_{\theta,j} = g^{a(\sigma+\tau_\theta)}, I'_\theta = g^{\tau_\theta}$.

Phase 2: The same procedures as **Phase1**, but the only limitation is that the keyword sets W_0, W_1 cannot be asked to $\mathcal{O}_{Token}$ oracle.

In summary, if $\mathcal{A}$ could construct $e(g,g)^{h'a(\sigma+\tau_\theta)}$ from the item $g^{h'}$ of the query return, then he can differentiate between $g^{a(\sigma+\tau_\theta)}$ and g^f. Yet, we still demand to prove that $\mathcal{A}$ can not gain $e(g,g)^{h'a(\sigma+\tau_\theta)}$ from the item $g^{h'}$ with a non-negligible advantage in the keyword privacy game.

We express two functions ϕ_1, ϕ_2 which map from the domain $\mathbb{Z}_p$ to a set of p^3 elements in the general group model. Let $\mathbb{G} = \{\phi_1(\mathcal{X})|\mathcal{X} \in \mathbb{Z}_p\}$, $\mathbb{G}_T = \{\phi_2(\mathcal{X})|\mathcal{X} \in \mathbb{Z}_p\}$, and distinguishing an element's advantage is negligible in the mapping between ϕ_1 and ϕ_2. Then, we explore the advantage of $\mathcal{A}$ constructing $e(g,g)^{h'a(\sigma+\tau_\theta)}$ from the item $g^{h'}$.

Now we consider how to gain $e(g,g)^{h'a(\sigma+\tau_\theta)}$ with the item $g^{h'}$. Because item σ can only be found from item $\alpha\sigma$, the item h' should contain α so as to get $e(g,g)^{h'a(\sigma+\tau_\theta)}$. Put another way, given $h' = h''\alpha$, $\mathcal{A}$ will attempt to construct $e(g,g)^{h''\alpha a(\sigma+\tau_\theta)}$, yet he still requires to get $h''\alpha a\tau_\theta$ from the item τ_θ and αa. However, α is the master key which is privately-owned for $\mathcal{B}$, so $\mathcal{A}$ cannot get $e(g,g)^{h''\alpha a(\sigma+\tau_\theta)}$ in either case.

In the end, we conclude that $\mathcal{A}$ cannot distinguish between $g^{a(\sigma+\tau_\theta)}$ and g^f, so $\mathcal{A}$ is less likely to distinguish between $g^{a(\sigma+\tau_\theta)}g^{\sigma H_1(W_0)}$ and $g^{a(\sigma+\tau_\theta)}g^{\sigma H_1(W_1)}$. Therefore, $\mathcal{A}$ can't break keyword privacy games with a non-negligible advantage.

Theorem 2: Assume the decisional q-parallel BDHE assumption holds in groups $\mathbb{G}$ and $\mathbb{G}_T$, there is no probability polynomial time adversary $\mathcal{A}$ who can break the security of our scheme with a non-negligible advantage.

Proof: Assume there exists a probability polynomial time adversary $\mathcal{A}$, if $\mathcal{A}$ can attack the security of our scheme with a non-negligible advantage $\varepsilon_1 = Adv_{\mathcal{A}}^{IND-sCP-CPA}$ in the selective security model, then we built a imitator $\mathcal{B}$ to settle the decisional q-parallel BDHE assumption with a non-negligible advantage $\frac{\varepsilon_1}{2}$.

The challenger $\mathcal{C}$ first randomly selects $a, s, b_1, \cdots, b_q$ in $\mathbb{Z}_p$ and sets:

$$
\begin{aligned}
\mathbf{y} = (&g, g^s, g^a, \cdots, g^{a^q}, , g^{a^{q+2}}, \cdots, g^{a^{2q}},\\
&\forall_{1\le j\le q}\ g^{sb_j}, g^{a/b_j}, \cdots, g^{a^q/b_j}, , g^{a^{q+2}/b_j}, \cdots, g^{a^{2q}/b_j},\\
&\forall_{1\le k,j\le q,k\ne j}\ g^{asb_k/b_j}, \cdots, g^{a^q sb_k/b_j}).
\end{aligned}
$$

Then $\mathcal{C}$ chooses a random bit $\upsilon \in \{0,1\}$; if $\upsilon = 0$, $\mathcal{C}$ sets $T = e(g,g)^{a^{q+1}s}$; if $\upsilon = 1$, $\mathcal{C}$ selects a random element T in $\mathbb{G}_T$.

Init: $\mathcal{B}$ received a decisional q-parallel BDHE challenge $(\mathbf{y}, T)$, $\mathcal{A}$ claims a challenge access structure $(\mathbf{M}^*, \rho^*)$, where $\mathbf{M}^*$ is a $l^* \times n^*$ matrix, and $l^*, n^* \leq q$.

Setup: $\mathcal{B}$ randomly chooses a elements $\beta' \in \mathbb{Z}_p$ and sets $e(g,g)^{\beta} = e(g^a, g^{a^q}) \cdot e(g,g)^{\beta'}$ which implicitly sets $\beta = \beta' + a^{q+1}$.

Let X be a scope of indices i where $\rho^*(i) = x$, $\mathcal{B}$ randomly chooses $z_x \in \mathbb{Z}_p$ for each attribute. Then $\mathcal{B}$ programs $H(x)$ as: If $X = \varnothing$, $H(x) = g^{z_x}$;else,

$$H(x) = g^{z_x} \prod_{i \in X} g^{a\mathbf{M}^*_{i,1}/b_i} g^{a^2\mathbf{M}^*_{i,2}/b_i} \cdots g^{n^*\mathbf{M}^*_{i,n^*}/b_i}.$$

In addition, for whichever attribute x, $\mathcal{B}$ initializes v_x by randomly choosing $v_x \in \mathbb{Z}_p$ and sets $PAK_x = g^{v_x}$.

Phase 1: $\mathcal{B}$ keeps a list tuple (uid, S_{uid}, SK_u) represented as $\mathcal{L}_{SK_u}$, which is initialized to null. $\mathcal{A}$ can query the following oracles in polynomial form:

- $\mathcal{O}_{SK_u}(uid, S_{uid})$: Suppose $\mathcal{B}$ receives a secret key query for (uid, S_{uid}), where S_{uid} does not satisfy the access structure $(\mathbf{M}^*, \rho^*)$, the specific process is as follows:

  If (uid, S_{uid}) has been asked before, simulator $\mathcal{B}$ retrieves SK_u from the $\mathcal{L}_{SK_u}$. If not, $\mathcal{B}$ chooses a vector $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_{n^*}) \in \mathbb{Z}_p^*$ such that $\eta_1 = -1$ and $\mathbf{M_i^*} \cdot \boldsymbol{\eta} = 0$ for all $i, \rho^*(i) \in S_{uid}$. According to the properties of LSSS, such a vector must exist. Then $\mathcal{B}$ randomly picks $r \in \mathbb{Z}_p$ and sets t as:

  $$t = r + \eta_1 a^q + \eta_2 a^{q-1} + \cdots + \eta_{n^*} a^{q+1-n^*}.$$

  The following, $\mathcal{B}$ performs the form by setting:

  $$D' = g^r \cdot \prod_{i=1,\cdots,n^*} (g^{a^{q+1-i}})^{\eta_i} = g^t.$$

  Through the definition of t, we noticed that g^{at} contains a term of $g^{-a^{q+1}}$, which will be cancelled out with the unknown term in g^{β} during constructing D. $\mathcal{B}$ computes D as:

  $$D = g^{\beta'} g^{ar} \cdot \prod_{i=2,\cdots,n^*} (g^{a^{q+2-i}})^{\eta_i}.$$

  Now $\mathcal{B}$ computes D_x for all $x \in S_{uid}$. For every attribute $x \in S_{uid}$, we let $D_x = (D')^{(}z_x/v_x)$ when there is no i such that $\rho^*(i) = x$. For the attribute $x \in S_{uid}$ used in the access structure, the terms of the form g^{a^{q+1}/b_i} are difficult to imitate. Fortunately, we have $\mathbf{M}_i^* \cdot \boldsymbol{\eta} = 0$, so the term g^{a^{q+1}/b_i} will be eliminated. $\mathcal{B}$ calculates D_x as:

  $$\begin{aligned} D_x &= (D')^{z_x/v_x} \prod_{i \in X} \prod_{j=1,\cdots,n^*} ((g^{a^j/b_i})^r)^{\mathbf{M}^*_{i,j}/v_x} \\ &\quad \cdot \prod_{i \in X} \prod_{j=1,\cdots,n^*} \prod_{\substack{k=1,\cdots,n^* \\ k \neq j}} ((g^{a^{q+1+j-k}/b_i})^{\eta_k})^{\mathbf{M}^*_{i,j}/v_x} \\ &= H(x)^{t/v_x}. \end{aligned}$$

  Finally, $\mathcal{B}$ adds the private key $SK_u = \{D, D', \{D_x\}_{x \in S_{uid}}\}$ to $\mathcal{L}_{SK_u}$ and sends it to $\mathcal{A}$.

- $\mathcal{O}_{AUK}(x')$: $\mathcal{A}$ sends an attribute x' for updating the attribute key request. $\mathcal{B}$ randomly selects a number $\overline{v}_{x'} \in \mathbb{Z}_p$ as the new attribute key of x', and then outputs attribute update key $AUK_{x'} = \overline{v}_{x'}/v_{x'}$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ submits two equal lengths of challenge informations $k_{\theta_0}, k_{\theta_1}$ to $\mathcal{B}$. Then $\mathcal{B}$ chooses a random bit $b \in \{0, 1\}$ and computes CT'^* as follows:

$$\widetilde{C}^* = k_{\theta_b} \cdot T \cdot e(g^s, g^{\beta'}), \quad \overline{C}^* = g^s.$$

It is hard for $\mathcal{B}$ to imitate $\hat{C}_i^*$ since it includes terms $g^{a_j s}$ that $\mathcal{B}$ cannot imitate. However, $\mathcal{B}$ can do the secret splitting, so that these terms be eliminated. For simple description, $\mathcal{B}$ randomly selects $y'_2, \cdots, y'_{n^*} \in \mathbb{Z}_p$, then shares the secret s by using vector

$$\mathbf{V} = (s, sa + y'_2, sa^2 + y'_3, \cdots, sa^{n^*-1} + y'_{n^*}) \in \mathbb{Z}_p^{n^*}.$$

For $i = 1, \cdots, l$, we describe R_i as the set of all $k \neq i$ making $\rho^*(i) = \rho^*(k)$. $\mathcal{B}$ also selects random value $r'_1, \cdots, r'_l \in \mathbb{Z}_p$. By implicitly setting $r_i = -r'_i - sb_i$, $\mathcal{B}$ calculates $\check{C}_i^*$ and $\hat{C}_i^*$ as:

$$\begin{aligned} \check{C}_i^* &= g^{(-r'_i - sb_i) \cdot v_{\rho^*(i)}}, \\ \hat{C}_i^* &= H(\rho^*(i))^{-r'_i} \cdot (g^{sb_i})^{-z_{\rho^*(i)}} \cdot \prod_{j=2,\cdots,n^*} (g^a)^{\mathbf{M}^*_{i,j} \cdot y'_j} \\ &\quad \cdot \prod_{k \in R_i} \prod_{j=1,\cdots,n^*} (g^{a^j sb_i/b_k})^{-\mathbf{M}^*_{k,j}}. \end{aligned}$$

Finally, $\mathcal{B}$ sends the ciphertext $CT'^* = \{\widetilde{C}^*, \overline{C}^*, \check{C}_i^*, \hat{C}_i^*\}$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ continues to perform queries similar to **Phase1**.

Guess: $\mathcal{A}$ will output a guess b' of b. If $b' = b$, $\mathcal{B}$ returns $v' = 0$ to guess $T = e(g,g)^{a^{q+1}s}$. Otherwise, $\mathcal{B}$ returns $v' = 1$ means that T is a random element in $\mathbb{G}_T$. If $v = 0$, $\mathcal{A}$ gets an effective ciphertext of k_b. In this case, the advantage of $\mathcal{A}$ is ϵ_1, hence $Pr[b' = b | v = 0] = 1/2 + \epsilon_1$. Because $\mathcal{B}$ returns $v' = 0$ when $b' = b$, hence $Pr[v' = v | v = 0] = 1/2 + \epsilon_1$. If $v = 1$, $\mathcal{A}$ can't get information about b, hence $Pr[b' = b | v = 1] = 1/2$. When $b' \neq b$, $\mathcal{B}$ returns $v' = 1$, hence $Pr[v' = v | v = 1] = 1/2$. Therefore, the total advantage of $\mathcal{B}$ in solving the decisional q-parallel BDHE assumption is :

$$\begin{aligned} Pr &= [v' = v] - \frac{1}{2} \\ &= \frac{1}{2} Pr[v' = v | v = 0] + \frac{1}{2} Pr[v' = v | v = 1] - \frac{1}{2} \\ &= \frac{1}{2}(\frac{1}{2} + \epsilon_1) + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} \\ &= \frac{\epsilon_1}{2}. \end{aligned}$$

VII. PERFORMANCE ANALYSIS

A. FUNCTIONAL COMPARISON

We compared some existing schemes with ours from keyword search, attribute revocation, result verification, and access control, as indicated in Table. 2. The symbol "$\surd$" indicates

TABLE 2. Functional comparison.

Schemes	Attribute revocation	Keyword search	Result verification	Access control
[17]	×	×	×	LSSS
[23]	×	√	×	AND-gate
[38]	√	×	×	Tree
[40]	√	×	×	LSSS
[42]	×	√	√	Tree
Ours	√	√	√	LSSS

TABLE 3. Storage cost comparison.

Schemes	[23]	[36]	[39]	ours
Setup	$(4+\sum_{i=1}^{n_a} n_i)\lvert\mathbb{G}\rvert$ $+(2+\sum_{i=1}^{n_a} n_i)\lvert\mathbb{Z}_p\rvert$	$(2+3n_a)\lvert\mathbb{G}\rvert$ $+(1+3n_a)\lvert\mathbb{Z}_p\rvert$	$(4+n_a)\lvert\mathbb{G}\rvert$ $+(2+3n_a)\lvert\mathbb{Z}_p\rvert$	$(2n_a+1)\lvert\mathbb{Z}_p\rvert$ $+6\lvert\mathbb{G}\rvert$
KeyGen	$(2+2n_a)\lvert\mathbb{G}\rvert$	$(1+2n_a)\lvert\mathbb{G}\rvert$ $+\lvert\mathbb{Z}_p\rvert$	$(1+3n_{a,u})\lvert\mathbb{G}\rvert$ $+\lvert\mathbb{Z}_p\rvert$	$(4+n_{a,u})\lvert\mathbb{G}\rvert$ $+\lvert\mathbb{Z}_p\rvert$
IndexGen	$(2+2n_a)\lvert\mathbb{G}\rvert$	$(2+n_{a,u})\lvert\mathbb{G}\rvert$	$3\lvert\mathbb{G}\rvert$	$3\lvert\mathbb{G}\rvert$
Encryption	–	–	$(2+2n_a)\lvert\mathbb{G}\rvert$	$(2+2n_{a,u})\lvert\mathbb{G}\rvert$
Token	$(1+2n_a)\lvert\mathbb{G}\rvert$ $+\lvert\mathbb{Z}_p\rvert$	$(1+2n_{a,u})\lvert\mathbb{G}\rvert$ $+\lvert\mathbb{Z}_p\rvert$	$2\lvert\mathbb{G}\rvert$	$3\lvert\mathbb{G}\rvert$

that the scheme has this function, and "×" indicates the opposite situation. As can be seen from the Table. 2, our scheme is more functional which supports the above characteristics simultaneously, and these features make our scheme more suitable for practical applications.

B. STORAGE COST COMPARISON

The number of elements in the group determines the space that is occupied. For ease of comparison, we define some symbols for storing the cost. $|\mathbb{Z}_p|$ and $|\mathbb{G}|$ respectively represent the bit length of the element in the domain $\mathbb{Z}_p$ and group $\mathbb{G}$, n_i represents the number of possible values of the attribute i, n_a represents the number of all attributes in the scheme, and $n_{a,u}$ represents the number of attributes owned by a user in the scheme, n_x indicates the number of attributes revoked, and φ represents the number of search results. Table. 3 below gives the storage cost. In Table. 3, we compare the storage overhead of our paper with the literatures [23], [36], [39] in system setup, key generation, index generation, encryption, and token generation. We mainly consider the bit length of the elements in the domain $\mathbb{Z}_p$ and the group $\mathbb{G}$.

C. CALCULATION COST COMPARISON

Before analyzing the computational cost, some main time-consuming operational symbols are given: bilinear pairwise operation P, exponential operation E. Here we ignore the computation overhead of the hash function because it is highly efficient compared with other expensive operations. In Table 4, we compare the calculation cost of our paper with the literatures [23], [36], and [39] in key generation, system setup, token generation, file search, key update, ciphertext update and verification. It can be seen from Table. 4 that the scheme [23] does not support key update, ciphertext update and verification algorithms. The computational complexity of our solution and other schemes in the system setup stage and key generation stage are linearly increasing with the number of attributes, but in the key generation phase, our scheme only needs 1 exponential operation related to the attribute, and the scheme [23], [39], and [36] need 2, 3 and 2 exponential operations respectively, which makes our scheme is more efficient than other schemes. In addition, the search computation cost of our scheme and the scheme [39] are constant $3P$ and $E + 2P$ respectively, but the search calculation cost of the schemes [36] and [23] are positively correlated with the number of attributes. In addition, since the result authentication mechanism performs the verification algorithm, this solution brings additional computational and storage costs.

D. EXPERIMENTAL SIMULATION

In order to analyze the actual performance of the system and related literatures, this paper uses a real data set and PBC (Pairing-Based Cryptography) library [44] to carry out a series of simulation experiments. The simulation is executed on a Windows machines with 3.40GHz Intel(R) Core (TM) i7-2630 CPU and 8GB ROM. JDK 1.7.5, MyElipse 10 and JPBC 2.0.0 are software runtime. In the experiment, the exponent operation E and the pair operation P, $|\mathbb{Z}_p| = 160bit$, $|\mathbb{G}| = 1024\ bit$ are mainly considered. For ease of description, this paper assumes that the number of attributes is $|n_a| = |n_{a,u}| \in [10, 50]$. The experimental results of the main algorithm are given below.

As shown in Fig. 2(a)–2(d), the storage costs of system setup, key generation, index generation, and token generation algorithms are described, respectively. As shown in Fig. 2(a) and Fig. 2(b), in the system setup and key generation phase, the storage overhead of the solution is far lower than that of the schemes [23], [36], and [39]. For instance, while setting $|n_a| = |n_{a,u}| = 30$, the storage cost of system setup algorithm

TABLE 4. Calculation cost comparison.

Schemes	[23]	[36]	[39]	ours
Setup	$(2+\sum_{i=1}^{n_a} n_i)E$	$(1+3n_a)E+P$	$(2+2n_a)E+P$	$(4+n_a)E+P$
KeyGen	$(2+2n_a)E$	$(3+2n_a)E$	$(2+3n_{a,u})E$	$(4+n_{a,u})E$
Token	$(2n_a+1)E$	$(1+2n_{a,u})E$	$E+2P$	$3E$
Search	$E+(2n_a+1)P$	$E+(n_a+1)P$	$E+2P$	$3P$
CTUpdate	–	$(2+n_{a,u})E$	n_xE	n_xE
Verification	–	–	–	$(\varphi+1)E+2P$

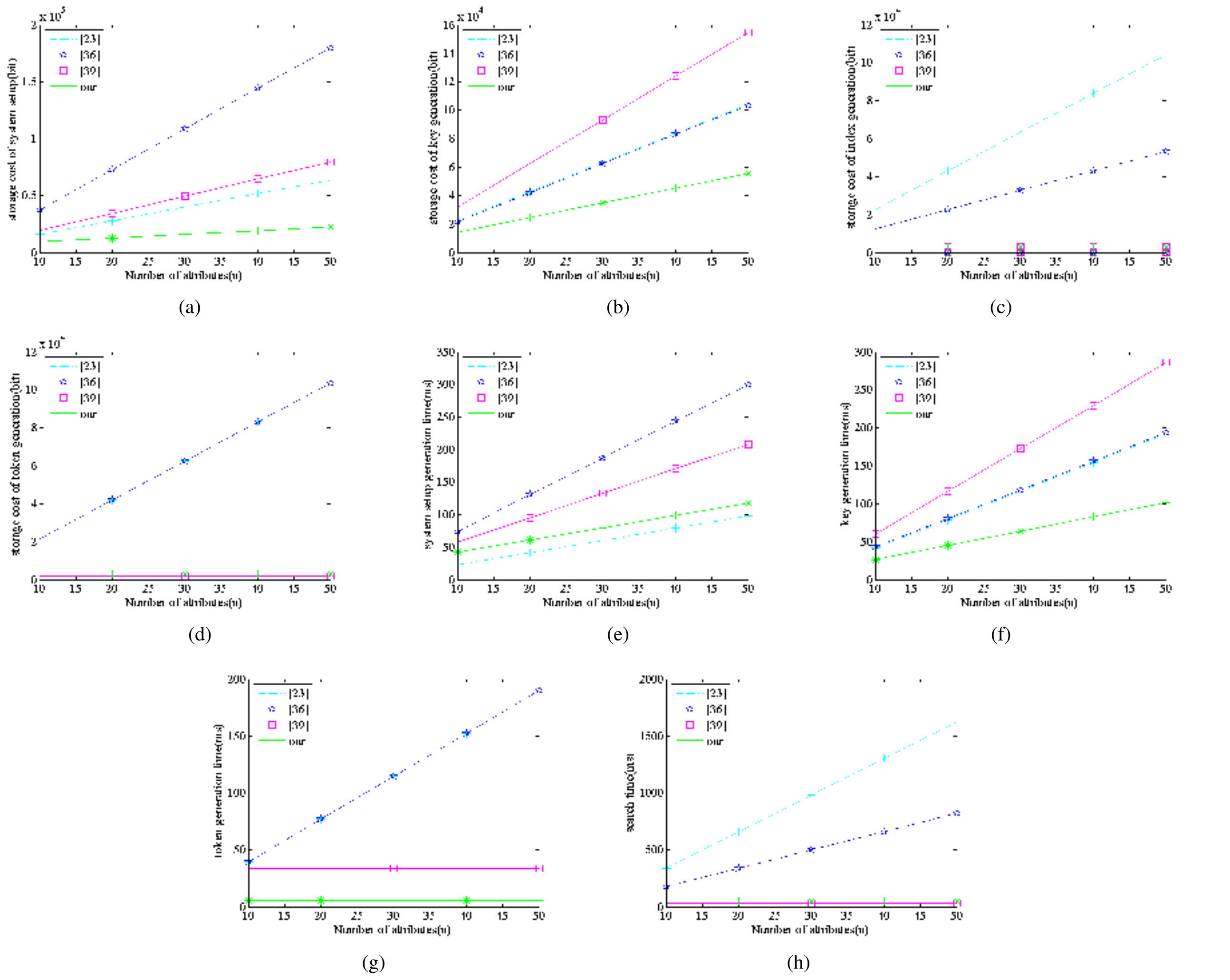

FIGURE 2. Performance evaluation. (a) Storage cost of system setup. (b) Storage cost of KeyGen. (c) Storage cost of IndexGen. (d) Storage cost of TokenGen. (e) System setup time. (f) Key generation time. (g) Token generation time. (h) Search time.

in [36], [39], and [23] is 144288 *bit*, 64576 *bit*, 51776 *bit*, respectively, but our scheme only required 19104 *bit*. So in fact, when $|n_a| \gg |n_{a,u}|$, our scheme is more efficient. From Fig. 2(c) and Fig. 2(d), it can be found that in index generation and token generation algorithms our scheme and the scheme [39] have basically the same storage cost, and are far lower than the schemes [36] and [23]. Therefore, ours scheme is more suitable for entities with limited resources, especially mobile terminals and sensor nodes.

As shown in Fig. 2(e)–2(h), the time costs of system setup, key generation, token generation and search algorithms are described, respectively. As shown in Fig. 2(e), the system setup time of our scheme and the scheme [23] are basically the same, which are smaller than the schemes [36] and [39]. As shown in Fig. 2(f), the key generation time of our scheme is much smaller than the other three schemes. From Fig. 2(g) and Fig. 2(h), it can be found that the time required for our scheme in the token generation phase and search phase is not affected by the number of attributes, which is a very small constant, but the schemes [23] and [36] grow linearly with the number of attributes. In summary, our scheme is efficient and feasible in practical applications.

VIII. CONCLUSION

Compared with the previous search encryption scheme, this paper not only implements multi-keyword search without reducing the search efficiency, but also introduces TPA to verify the correctness of the search results and realize the function of user attribute revocation. Our scheme has proven to be secure against selectively chosen keyword attack in the general bilinear group model and be resistant to selective plaintext attacks. Then the feasibility of our scheme was verified by both theory and experiment. Additionally, with the needs of practical applications and further research, the problems that we need to be considered are making use of blockchain technology to design more secure, high search efficiency, and support dynamic dataset search schemes.

REFERENCES

[1] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges," *Inf. Sci.*, vol. 305,, pp. 357–383, Jun. 2015. doi: 10.1016/j.ins.2015.01.025.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT*, R. Cramer, Ed. Berlin, Germany: Springer, 2005, pp. 457–473.

[3] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2000, pp. 44–55.

[4] B. Dan, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer. 2004, pp. 506–522.

[5] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," in *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 9, pp. 2546–2559, Sep. 2016. doi: 10.1109/TPDS.2015.2506573.

[6] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 11, pp. 2594–2608, Nov. 2016. doi: 10.1109/TIFS.2016.2590944.

[7] Y. Yang, J. Han, W. Susilo, T. H. Yuen, and J. Li, "ABKS-CSC: Attribute-based keyword search with constant-size ciphertexts," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5003–5015, Dec. 2016. doi: 10.1002/sec.1671.

[8] 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.

[9] L. Zu, Z. Liu, and J. Li, "New ciphertext-policy attribute-based encryption with efficient revocation," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. (CIT)*. Sep. 2014, pp. 281–287.

[10] K. Yang and X. Jia, "ABAC: Attribute-based access control," in *Security for Cloud Storage Systems*. New York, NY, USA: Springer, 2014, pp. 39–58. doi: 10.1109/mc.2015.33.

[11] T. Naruse, M. Mohri, and Y. Shiraishi, "Provably secure attribute-based encryption with attribute revocation and grant function using proxy re-encryption and attribute key for updating," *Hum.-Centric Comput. Inf. Sci.*, vol. 5, no. 1, p. 8, Dec. 2015. doi: 10.1186/s13673-015-0027-0.

[12] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology —CRYPTO*, G. R. Blakley and D. Chaum, Eds. Berlin, Germany: Springer, 1985, pp. 47–53.

[13] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. Inf. Commun. Technol.-EurAsia Conf. (CCS)*. New York, NY, USA: ACM, 2006, pp. 89–98.

[14] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2271–2282, Oct. 2013. doi: 10.1109/TKDE.2011.78.

[15] G. Jian, Z. Kang, and J. Heng-Zhan, "Data access control scheme based on CP-ABE in cloud storage," *J. Northeastern Univ.*, vol. 36, no. 10, pp. 1416–1421, Oct. 2015.

[16] H. Su, Z. Zhu, and L. Sun, "Practical searchable CP-ABE in cloud storage," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 180–185. doi: 10.1109/CompComm.2016.7924689.

[17] H. Cui, R. H. Deng, G. Wu, and J. Lai, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures," *Comput. Netw.*, vol. 133, pp. 157–165, Mar. 2018. doi: 10.1016/j.comnet.2018.01.034.

[18] H. Deng *et al.*, "Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts," *Inf. Sci.*, vol. 275, pp. 370–384, Aug. 2014. doi: 10.1016/j.ins.2014.01.035.

[19] J. Li, H. Wang, Y. Zhang, and J. Shen, "Ciphertext-policy attribute-based encryption with hidden access policy and testing," in *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 7, pp. 3339–3352, Jul. 2016. doi: 10.3837/tiis.2016.07.026.

[20] H. Wang and Y. Peng, "A CP-ABE access control scheme based on proxy re-encryption in cloud storage," in *Proc. Int. Conf. Cloud Comput. Secur. (ICCCS)*. Cham, Switzerland: Springer., 2018, pp. 413–425. doi: 10.1007/978-3-030-00009-7_38.

[21] K. Zhang, H. Li, J. Ma, and X. Liu, "Efficient large-universe multi-authority ciphertext-policy attribute-based encryption with white-box traceability," *Sci. China Inf. Sci.*, vol. 61, no. 3, Mar. 2018, Art. no. 032102. doi: 10.1007/s11432-016-9019-8.

[22] S. Wang, X. Zhang, and Y. Zhang, "Efficiently multi-user searchable encryption scheme with attribute revocation and grant for cloud storage," *PLoS ONE*, vol. 11, no. 11. Nov. 2016, Art. no. e0167157. doi: 10.1371/journal.pone.0167157.

[23] S. Qiu, J. Liu, Y. Shi, and R. Zhang, "Hidden policy ciphertext-policy attribute-based encryption with keyword search against keyword guessing attack," *Sci. China Inf. Sci.*, vol. 60, May 2017, Art. no. 052105. doi: 10.1007/s11432-015-5449-9.

[24] H. Yin *et al.*, "CP-ABSE: A ciphertext-policy attribute-based searchable encryption scheme," *IEEE Access*, vol. 7, pp. 5682–5694, 2019. doi: 10.1109/ACCESS.2018.2889754.

[25] K. Kaushik, V. Varadharajan, and R. Nallusamy, "Multi-user Attribute Based Searchable Encryption," in *Proc. IEEE 14th Int. Conf. Mobile Data Manage.*, Milan, Italy, Jun. 2013, pp. 200–205. doi: 10.1109/MDM.2013.94.

[26] K. Liang and W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1981–1992, Sep. 2015. doi: 10.1109/TIFS.2015.2442215.

[27] J. Cui, H. Zhou, H. Zhong, and Y. Xu, "AKSER: Attribute-based keyword search with efficient revocation in cloud computing," *Inf. Sci.*, vol. 423, pp. 343–352, Jan. 2018. doi: 10.1016/j.ins.2017.09.029.

[28] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 3, pp. 312–325, May/Jun. 2016. doi: 10.1109/TDSC.2015.2406704.

[29] H. Haiping, D. U. Jianpeng, H. Dai, and R. Wang, "Multi-sever multi-keyword searchable encryption scheme based on cloud storage," *J. Electron. Inf. Technol.*, vol. 39, no. 2, pp. 389–396, Feb. 2017.

[30] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 2, pp. 340–352, Feb. 2016. doi: 10.1109/TPDS.2015.2401003.

[31] Z. Fu, F. Huang, K. Ren, J. Weng, and C. Wang, "Privacy-preserving smart semantic search based on conceptual graphs over encrypted outsourced data," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 8, pp. 1874–1884, Aug. 2017. doi: 10.1109/TIFS.2017.2692728.

[32] H. Li, D. Liu, Y. Dai, T. H. Luan, and X. S. Shen, "Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 1, pp. 127–138, Mar. 2015. doi: 10.1109/TETC.2014.2371239.

[33] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 12, pp. 2706–2716, Dec. 2016. doi: 10.1109/TIFS.2016.2596138.

[34] D. Fiore and R. Gennaro, "Publicly verifiable delegation of large polynomials and matrix computations, with applications," in *Proc. ACM CCS*, Oct. 2012, pp. 501–512.

[35] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: Verifiable attribute-based keyword search over outsourced encrypted data," in *Proc. IEEE INFOCOM*, Apr./May 2015, pp. 522–530. doi: 10.1109/INFOCOM.2014.6847976.

[36] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, ''Protecting your right: Verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 4, pp. 1187–1198, Apr. 2016.

[37] D. Li, C. Jie, J. Liu, Q. Wu, and W. Liu, ''Efficient CCA2 secure revocable multi-authority large-universe attribute-based encryption,'' in *Proc. Int. Symp. Cybersp. Safety Secur.*, vol. 10581. Cham, Switzerland: Springer, Oct. 2017, pp. 103–118. doi: 10.1007/978-3-319-69471-9_8.

[38] 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. doi: 10.1109/JSYST.2017.2667679.

[39] S. Wang, X. Yu, and Y. Zhang, ''Revocable key-policy attribute-based encryption scheme with two revocation lists,'' *J. Electron. Inf. Technol.*, vol. 38, no. 6, pp. 1406–1411, 2016. doi: 10.11999/JEIT150845.

[40] L. Xue, Y. Yu, Y. Li, M. H. Man, X. Du, and B. Yang, ''Efficient attribute-based encryption with attribute revocation for assured data deletion,'' *Inf. Sci.*, vol. 479, pp. 640–650, Apr. 2018. doi: 10.1016/j.ins.2018.02.015.

[41] M. Green, S. Hohenberger, and B. Waters, ''Outsourcing the decryption of ABE ciphertexts,'' in *Proc. USENIX Secur. Symp.*. San Francisco, CA, USA, Aug. 2011, p. 34. doi: 10.1192/bjp.179.4.330.

[42] Y. Miao, J. Ma, Q. Jiang, X. Li, and A. K. Sangaiah, ''Verifiable keyword search over encrypted cloud data in smart city,'' *Comput. Elect. Eng.*, vol. 65, pp. 90–101, Jan. 2017.

[43] B. Waters, ''Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,'' in *Public Key Cryptography—PKC*, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds. Berlin, Germany: Springer, 2011, pp. 53–70.

[44] L. Helge , ''Pairing-based cryptography,'' *Math.iisc.ernet.in*, vol. 22, no. 3, pp. 573–590, 2005. doi: 10.1109/ISCC.2011.5983948.

JIN SUN received the B.S. degree in mathematics from Shaanxi Normal University, Xi'an, China, in 2000, the M.S. degree in applied mathematics from the Xi'an University of Technology, Xi'an, in 2005, and the Ph.D. degree in cryptology from Xidian University, Xi'an, in 2012. She is currently an Associate Professor with the Xi'an University of Technology. Her current research interests include cryptography, information security, and network security.

LILI REN received the B.S. degree from the School of Mathematics and Statistics, Tianshui Normal University, Tianshui, China, in 2017. She is currently pursuing the M.S. degree with the Xi'an University of Technology, Xi'an, China. Her research interests include cryptography and information security.

SHANGPING WANG received the B.S. degree in mathematics from the Xi'an University of Technology, Xi'an, China, in 1982, the M.S. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 1989, and the Ph.D. degree in cryptology from Xidian University, Xi'an, China, in 2003. He is currently a Professor with the Xi'an University of Technology. His current research interests include cryptography and information security.

XIAOMIN YAO received the B.S. degree from the School of Mathematics and Statistics, Tianshui Normal University, Tianshui, China, in 2018. She is currently pursuing the M.S. degree with the Xi'an University of Technology, Xi'an, China. Her research interests include information security and blockchain technology.