IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 25, 2019, accepted April 9, 2019, date of publication April 12, 2019, date of current version April 25, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2910828*

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

**SHANGPING WANG**[1], **SHASHA JIA**[1], **AND YALING ZHANG**[2]

[1]School of Science, Xi'an University of Technology, Xi'an 710054, China
[2]School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710054, China

Corresponding author: Shasha Jia (jshasha25@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572019, and in part by the Key Project of the Natural Science Foundation of Shaanxi Province of China under Grant 2016JZ001.

**ABSTRACT** In attribute-based searchable encryption (ABSE) scheme, data owners can encrypt their data with access policy for security consideration, and encrypt keywords to obtain keyword index for privacy keyword search, and data users can search interesting keyword on keyword indexes by keyword search trapdoor. However, many existing searchable encryption schemes only support single keyword search and most of the existing attribute-based encryption (ABE) schemes have high computational costs at user client. These problems significantly limit the application of attribute-based searchable encryption schemes in practice. In this paper, we propose a verifiable and multi-keyword searchable attribute-based encryption (VMKS-ABE) scheme for cloud storage, in our new scheme, multi-keyword can be searched and the search privacy is protected. That is, the cloud server can search the multi-keyword with keyword search trapdoor but it does not know any information about the keywords searched. In the proposed scheme, many computing tasks are outsourced to the cloud proxy server, which greatly reduces the computing burden at the user client. Besides, the scheme also supports the verification of the correctness of the outsourced private key. The proposed scheme is proved secure that the keyword index is indistinguishable under the adaptive keyword attacks in the general group model, and the ciphertext is selective secure under selective plaintext attacks in the random oracle model. The security and experimental results show that our scheme is suitable for practicability.

**INDEX TERMS** Attribute-based encryption, verifiable outsourcing, multi-keyword search, adaptive security.

## I. INTRODUCTION

With the development of cloud computing, many of information can be shared through computer networks. The cloud server (CS) can provide users with a variety of services, such as outsourcing commission calculations and data storage. Users can store their large amounts of data to the CS and share data with other users. For the purpose of the security of storage data and user's privacy, data is usually stored in encrypted form in CS. However, under this environment users will encounter a difficulty problem of how to search keyword in ciphertext. Searchable Encryption (SE) is a cryptographic technology that has been developed for many years, which supports users' keyword search in ciphertext. In the meanwhile, it can save a lot of network and computational overhead for user, and take advantage of the huge computing power of CS.

The SE technology mainly solves the problem of how to use the server to complete the search for interesting keywords when the data is encrypted and stored in CS, but CS is not completely trusted. How to improve the efficiency of keyword search while reducing local computing load is still a problem to be solved. Most of existing schemes support single-keyword search. Single-keyword search waste network bandwidth and computing resources, as this search method returns a large number of results, this means that the

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

search result is not accurate. That is, when a data user uses multi- keyword search, the cloud server will return relatively few number of files containing these multi-keyword, thus the search result is much more accurate than when a data user uses one keyword search. In order to solve this problem, multi-keyword search is proposed.

Most of existing attribute-based encryption (ABE) schemes have high computational costs at user client. These problems greatly limit the applications of ABE schemes in practice. To solve the problems of network bandwidth waste and high computational cost, we propose a verifiable and multi-keyword searchable attribute-based encryption (VMKS-ABE) scheme for cloud storage, in which many computing tasks are outsourced to cloud proxy server to reduce local computing burden, the scheme also supports the verification of the correctness of outsourced private keys. In our new scheme multi-keyword can be searched and the search privacy is protected, which can greatly improve the accuracy of keyword search.

## A. RELATED WORK

### 1) SEARCHABLE ENCRYPTION

Song *et al.* [1] first proposed the concept of searchable encryption (SE), which provides a basic method for searching on encrypted cloud data. Dong *et al.* [2] used RSA public key encryption algorithm and proxy encryption technology to implement a SE scheme in a multi-user environment. Li and Xu [3] proposed ABSE scheme based on the attribute encryption algorithm, and proved that the scheme can achieve indistinguishable safety against chosen keyword plaintext attacks under the selective model of attribute set. Subsequently, many experts and scholars published their solutions about the problem of how to conduct secure keyword search in encrypted data [4]–[6]. To encrypt the data, and enable users who have corresponding access rights to search encrypted data. Sun *et al.* [7], and Dong *et al.* [8] constructed ABSE schemes to implement fine-grained access control and search for encrypted data. Attribute-based keyword search has been focused extensively because it can implement flexible access policy. Notably, the computation cost and communication cost in existing ABSE schemes are linear with the number of required attributes. Ye *et al.* [9] constructed ABSE with constant-size ciphertexts schemes, the schemes realizes a constant calculation cost and the ciphertext size remains unchanged. Moreover, because data destruction and improper operation, the CS may return error search answers. Consequently, it is very significant to ensure the correctness of returned answers in semi-trusted cloud environment. Under these circumstances, Chai and Gong [10] proposed the first keyword search scheme that can provide verifiable search capabilities.

### 2) ATTRIBUTE-BASED ENCRYPTION (ABE)

The concept of ABE was proposed by Sahai and Waters [11]. ABE can be classified into two types: one is the key-policy attribute-based encryption (KP-ABE) [12]; the other is the ciphertext-policy attribute-based encryption (CP-ABE) [13]. In the CP-ABE schemes, the ciphertext is related to an access policy, and private key of each user is related to the attribute set of the user. Users can decrypt a ciphertext only if his/her attribute set satisfies the access policy of the ciphertext. In the KP-ABE schemes, the attribute set and access policy are opposite to those described in the CP-ABE scheme. In the decryption process, only if a user's attributes set satisfies the access policy, the use can do decryption correctly. After attribute-based encryption schemes were proposed, there are many research works about ABE, such as CP-ABE schemes [14], [15], ABE schemes with hidden-policy [16]–[18], hierarchical attribute-based encryption schemes [19], [20], multi-authorization center ABE schemes [21] and traceable ABE schemes [22], [23]. However, in the above ABE schemes, the number of operations in the decryption process is associated with the complexity of access policy, and the user's computing power is limited. Therefore, how to decrease the user's computational load becomes an urgent problem to be solved. Green *et al.* [15] provided an ABE scheme in which partial decryption operations are outsourced to the CS. Wang *et al.* [24] proposed an adaptive security outsourcing CP-ABE scheme. But, they only considered the requirements of decryption outsourcing. Rui *et al.* [25] proposed a fully outsourced ciphertext-policy ABE scheme that for the first time achieves outsourced key generation, encryption and decryption simultaneously. However, although CS has strong computing power, it is not completely trustworthy. CS is usually regarded as honest but curious. To ensure that CS can perform the ciphertext conversion process correctly, Lai *et al.* [26] proposed a verifiable outsourced ABE scheme that can verify the correctness of decryption. Their scheme adds additional information to the ciphertext and this information is used for verification. To decrease the length of encrypted ciphertext, Mao *et al.* [27] presented a new verifiable ABE scheme based on the scheme proposed by Lai *et al.* [26]. Instead of encrypting the random message independently, the scheme [27] concatenates random message with original message before encrypting them. This greatly reduces the size of the original ciphertext, decreases the communication cost of the solution. Li *et al.* [28] proposed a new outsourced ABE scheme which supports both secure outsourced key-issuing and decryption. In 2016, Wang *et al.* [29] introduced the concept of verifiable outsourcing, that is, key generation center, data owner and data user can outsource their computational tasks to corresponding service providers to reduce local loads. The above schemes mainly focuses on verifiability of outsourced decryption for the authorized users. In 2017, Li *et al.* [30] proposed an ABE solution with verifiable outsourced decryption (referred to as full verifiability of outsourced decryption), which can simultaneously check the correctness of conversion passwords of authorized users and unauthorized users.

## B. OUR CONTRIBUTIONS

Based on the scheme named ABE with verifiable outsourced decryption [26], a verifiable and multi-keyword searchable attribute-based encryption (VMKS-ABE) scheme in cloud storage is proposed in this paper. The scheme [26] only supports outsourcing of decryption, compared with scheme [26], our goal is to design a more comprehensive ABE scheme that can solve several problems in practice. The scheme of [26] consists of the seven algorithms, including Setup, KeyGen, Encrypt, Decrypt, GenTkout, Transformout, Decryptout, in which involves decryption outsourcing. However, our framework adds outsourced private key generation, outsourced private key verification, outsourced encryption, keyword encryption, trapdoor generation and test algorithms compared to scheme [26]. The outsourced private key generation algorithm outsources part of the private key to the cloud proxy server to reduce the computational complexity of the private key generation stage. The outsourced private key verification algorithm is used to verify the correctness of the outsourced private key. The outsourced decryption algorithm is used to reduce the computational complexity of the decryption phase. The word encryption algorithm, trapdoor generation algorithm and matching algorithm are used to implement the keyword search function.

Specifically, our scheme supports three functions: (1) multiple keyword searches; (2) full outsourcing; (3) verifiability of outsourced private keys. Therefore, by changing some specific constructions of the algorithm, the three main advantages of our scheme can be extended to the general attribute-based encryption scheme (such as the encryption scheme without considering the local computing burden, or the encryption scheme lacking the ciphertext search). Achieve the ability to reduce local computing storage and accurately search ciphertext. The specific features of our schemes are as follows:

1) In our scheme multi-keyword can be searched, and the search privacy is protected. That is, CS can search the multi-keyword with keyword search trapdoor but it does not know any information about the keywords searched. Considering that keyword search is indispensable for ABE in practice, and our scheme supports multiple keyword search, so our scheme is also a combination of ABE and SE.
2) In our scheme, most of the computational burden is outsourced to cloud proxy server to reduce local computing task at user client, including private key generation, encryption, and decryption algorithm.
3) Additionally, our scheme also supports the verification of outsourced private keys. As the outsourced private key generation service provider is not completely trusted, the attribute authority cannot judge whether the requested result is honestly returned. Therefore, it is necessary to verify the correctness of outsourced private keys.
4) Under the general group model, the security of the scheme is proved that the keyword index is indistinguishable under the adaptive keyword attacks, and the ciphertext is selective security against chosen plaintext attacks (CPA) in the random oracle model.

## II. PRELIMINARIES

### A. BILINEAR MAP

*Definition 1 (Bilinear Maps [31]):* Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative cyclic groups of prime order $p$, $g$ be a generator of $\mathbb{G}$. The bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ satisfies the following properties:

1) Bilinear: $\forall a,\ b \in Z_p,\ e(g^a, g^b) = e(g, g)^{ab}$ holds.
2) Non-degenerate: $\exists g \in \mathbb{G},\ e(g, g) \neq 1$.
3) Computability: $\forall u, v \in \mathbb{G},\ e(u, v)$ can be effectively calculated.

### B. ACCESS STRUCTURE

*Definition 2 (Access Structure [32]):* Assuming $\{P_1, P_2, \cdots, P_n\}$ is a set of participants, if for any set $B, C$, there are $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$ the access structure $\mathbb{A} \subseteq 2^{\{P_1,P_2,\cdots,P_n\}}$ is monotonous. An access structure is collection of non-empty subset of the set $\{P_1, P_2, \cdots, P_n\}$. The collection in the access structure $\mathbb{A}$ is called the authorization collection, and the collection not in the access structure $\mathbb{A}$ is called the unauthorized collection.

### C. LINEAR SECRET SHARING SCHEME

*Definition 3 (Linear Secret Sharing Scheme (LSSS) [33]):* LSSS $\Pi$ defined on the entity set $P$ satisfies the following two points.

1. A shared composition for every entity forms a vector on $\mathbb{Z}_p$.

2. For LSSS $\Pi$, there is a $l \times n$ sharing matrix $\mathbf{M}$ and a mapping from $\{1, 2, \cdots, l\}$ to $P$. Randomly choosing $\mathbf{v} = \{s, v_2, \cdots, v_n\} \in Z_p^n$, where the secret to be shared is $s \in \mathbb{Z}_p$, then $\mathbf{Mv}$ is the vector of $l$ shares of the secret $s$ according to $\Pi$, which $(\mathbf{Mv})_i$ belongs to entity $\rho(i)$ and record as $\lambda_i$.

Each of LSSS the above definitions has the nature of linear reconstruction. Assuming $\Pi$ is a LSSS corresponding to the access policy $\mathbb{A}$, for any authorization set $S \in \mathbb{A}$, let defined as $I \subset \{i : \rho(i) \in S\}$ and $I \subset \{1, 2, \cdots, l\}$. If $\{\lambda_i\}$ it is a valid sharing of secret $s$ based on $\Pi$, there is a constant set $\{\omega_i \in Z_p\}_{i\in I}$ then $\sum_{i\in I} \omega_i\lambda_i = s$; for any non-authorized set, there will exist a vector $\mathbf{w} \in Z_p^n$, such that $\mathbf{w} \cdot (1, 0, \cdots, 0)^T = -1$ and $\mathbf{w} \cdot \mathbf{M}_i = 0 \quad$ for all $\ i \in I$.

### D. GENERAL BILINEAR GROUP MODEL

*Definition 4 (General Bilinear Group Model [34]):* We consider two random encodings $\psi_0, \psi_1$ of the additive group $\mathbb{Z}_p$, that is injective maps $\psi_0, \psi_1$: $\mathbb{Z}_p \rightarrow \{0, 1\}^m$, where $m > 3 \log p$. For $i = 0, 1$, let $\mathbb{G}_i = \{\psi_i(x) : x \in \mathbb{Z}_p\}$. We are given oracles to compute the induced group action on $\mathbb{G}$, $\mathbb{G}_T$ and an oracle to compute a non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. Give a random oracle to represent the hash function $H$, which refer to $\mathbb{G}$ as a general bilinear group.

## III. SYSTEM FRAMEWORK AND SECURITY MODEL

In this part, we define the symbols which will be used in our scheme. We provide a system description and a system model of VMKS-ABE scheme, and further provide a security model of the scheme.

### A. SYMBOL DESCRIPTION

| Symbols | Instructions |
|---|---|
| $\lambda$ | security parameter |
| $U$ | universal set of attributes |
| $S$ | user attributes |
| $IK$ | Intermediate private key |
| $SK_o$ | Outsourcing private key |
| $SK_L$ | User's private key |
| $RK$ | The user's retrieval key |
| $m$ | Plaintext Message |
| $w_j\,(j=1,2,\cdots,n)$ | Keywords |
| $w_{j'}\,(j'=1,2,\cdots,d)$ | Keywords of the query |
| $WD$ | Keywords set |
| $WD'$ | Keywords set of the query |
| $CT$ | ciphertext |
| $I$ | indexes |
| $TD$ | trapdoor |
| $E$ | Partially decrypted ciphertext |

### B. FRAMEWORD OF VMKS-ABE SCHEME

The VMKS-ABE scheme consists of six entities: $CS$, Cloud proxy server ($CPS$), Attribute authority ($AA$), Outsourced private key generation service provider ($OKGPS$), Data owner ($DO$) and Users ($U$). The relationship between them is shown in Fig 1.

$AA$ is a completely trusted third party in the system. It is responsible for establishing the system, generating and distributing public parameters. Meanwhile, $AA$ generates local private key and retrieval key for each authorized user, the intermediate private key and sent it to $CPS$.

$OKGPS$ generates the outsourced private key by using the public parameters.

$DO$ encrypts data which he intends to share and transmit it to $CS$.

When $U$ wants to access encrypted data, he can decrypt a ciphertext to obtain plaintext data only when his attributes satisfy corresponding access policy in the ciphertext.

When $AA$ gains outsources private key sent by $OKGPS$, $CPS$ provides the authentication function for outsourced private key; when $DO$ wants to encrypt the message, $CPS$ is responsible for partial encryption; when $U$ wants to decrypt a message, $CPS$ takes charge of partial decryption.

$CS$ has a large amount of storage space to store the ciphertext and index uploaded by $DO$ and it can perform matching tests. When $CS$ receives a trap door sent from $U$, it executes match operation between trapdoor and index, then the match result is returned to $U$.

Assume that $CS$ is honest and curious. That is, the $CS$ can honestly execute the algorithm according to the protocol. But at the same time, it will analyze and guess the data it receives to get extra information.

*Definition 5: Our VMKS-ABE scheme contains the following six algorithms:*

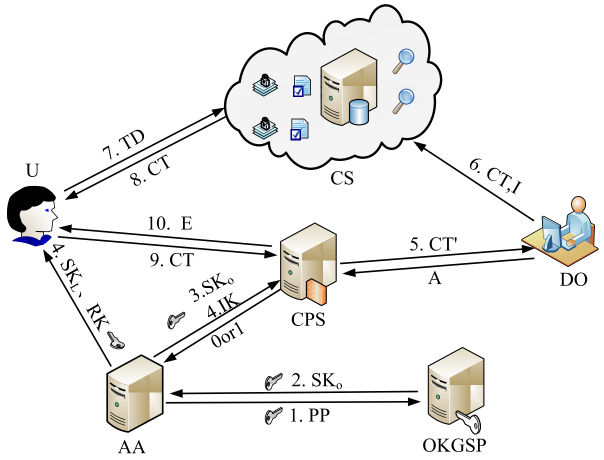


**FIGURE 1.** Frame of VMKS-ABE scheme.

#### 1. SYSTEM SETUP

- **Setup** $(\lambda, U) \rightarrow PP, MSK$. The setup algorithm is executed by $AA$. It inputs security parameters $\lambda$, attributes universal set $U$, outputs the public parameters $PP$ and master secret key $MSK$. $AA$ publishes $PP$ and keeps $MSK$ secretly.

#### 2. KEY GENERATION

- **Outsourcing KeyGen** $(PP, S) \rightarrow SK_o$. The outsourcing private key generation algorithm is executed by $CPS$. It inputs $PP$ and a set of attributes $S$, outputs outsource private key $SK_o$, and sends $SK_o$ to $AA$.
- **Outsourcing KeyGen verification** $(PP, SK_o) \rightarrow b \in \{0, 1\}$. The outsourced private key verification algorithm is executed by $CPS$. The algorithm inputs $PP$ and $SK_o$. If the verification is succeed, the algorithm outputs is 1. Otherwise, outputs 0.
- **KeyGen** $(MSK, S, SK_o) \rightarrow SK_L, IK, RK$. The private key generation algorithm is executed by $AA$, and the algorithm inputs $MSK$, the users attribute set $S$, and $SK_o$. It outputs local private key $SK_L$, intermediate private key $IK$, and retrieval key $RK$. Among them, the local private key and retrieval key are sent to the user, and the intermediate private key is sent to $CPS$.

#### 3. ENCRYPTION

- **Outsource encryption** $(PP, \mathbb{A} = (M, \rho)) \rightarrow CT'$. The outsource encryption algorithm is executed by $CPS$, the algorithm inputs $PP$, and access policy $\mathbb{A}$, outputs intermediate ciphertext $CT'$.
- **Encryption** $\left(PP, CT', m, WD\right) \rightarrow CT, I$. The encrypt-ion algorithm is divided into two steps, one step is message encryption, and the other step is keyword encryption. Both algorithms are executed by $DO$. **a) Message encryption** $\left(PP, CT', m\right) \rightarrow CT$. The algorithm inputs public parameters $PP$, messages $m$, and intermediate ciphertext $CT'$. It outputs ciphertext $CT$ and sends to $CS$.

**b) Keyword encryption** $(PP, WD) \rightarrow I$. The algorithm inputs $PP$, keyword set $WD$, outputs the indexes $I$ and transmits to $CS$.

*4. TRAPDOOR GENERATION*

- **Trapdoor generation** $\left(w_{j'}, SK_L\right) \rightarrow TD$. The user inputs $SK_L$ and the keyword set $WD'$ that the wants to be queried to generate a trapdoor $TD$ and sends it to $CS$.

*5. SEARCH*

- **Test** $(I, TD) \rightarrow 0 or 1$. The $CS$ takes trapdoors $TD$ and index $I$ as input. If the trapdoor and index can match successfully, the algorithm outputs 1, otherwise outputs 0.

*6. DECRYPTION*

- **Outsourcing decryption** $(PP, IK, CT) \rightarrow E$. The algorithm performs decryption of ciphertext $CT$ through $CPS$ under the access policy $(\mathbf{M}, \rho)$. It inputs the $PP$, $IK$ and corresponding ciphertext $CT$. If the attribute does not satisfy the access policy, the algorithm outputs $\perp$. Otherwise, it outputs partially decrypted ciphertext $E$ and sends it to $U$.
- **Decryption** $(RK, E) \rightarrow m$. It inputs partially decrypted ciphertext $E$ and retrieval key $RK$, outputs $m$.

### C. SECURITY MODEL OF VMKS-ABE SCHEME

We consider the semantically secure for VMKS-ABE scheme. We define a security game for the keyword index, we consider the indistinguishable of keyword index against the adaptive chosen keyword attacks. In this game, the adversary can get the trapdoor of the keyword set which he wants to inquire, but cannot distinguish the encrypted ciphertexts of the keyword set $WD_0$ and the keyword set $WD_1$. The security interactive game of challenger $\mathcal{C}$ and adversary $\mathcal{A}$ as follows:

- Challenger $\mathcal{C}$ executes the setup algorithm to get $PP$ and sends $PP$ to $\mathcal{A}$.
- A can issue the inquiry of search trapdoor adaptively to the challenger $\mathcal{C}$ about the keyword set WD' related to attribute set S.
- $\mathcal{A}$ commits two keyword set $WD_0$ and $WD_1$ as challenge. In addition, $\mathcal{A}$ gives a challenge access policy $\mathbb{A}$. The limitation is that the attribute set $S$ cannot satisfy $\mathbb{A}$. The $\mathcal{C}$ randomly chooses $b \in \{0, 1\}$ to generate the challenge index of $WD_b$ and sends to $\mathcal{A}$.
- $\mathcal{A}$ repeats the query similar to phase 2). The limitation is $WD_0$ and $WD_1$ cannot be queried.
- Finally, $\mathcal{A}$ outputs guess $b'$ of $b$, if $b = b'$, $\mathcal{A}$ wins the game.

The advantage of $\mathcal{A}$ in the above game is defined as $Adv = \left|\Pr[b = b'] - 1/2\right|$.

*Definition 6 ($IND - CKA$): If the advantage for all probability polynomial time adversary in the above game is negligible, then VMKS-ABE scheme is of the security that the keyword index is indistinguishable against the adaptive chosen keyword attacks.*

We now give an indistinguishable definition of the CP-ABE under CPA. If adversary $\mathcal{A}$ submits a challenge access policy $\mathbb{A}^*$ before the setup phase, it is called selective security. The security interactive game between the challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows:

- **Initialization:** Adversary $\mathcal{A}$ gives a challenge access policy $\mathbb{A}^*$ to $\mathcal{C}$.
- **Setup:** runs the setup algorithm to get $MSK$ and $PP$, then $\mathcal{C}$ sends $PP$ to $\mathcal{A}$.
- **Phase 1:** $\mathcal{C}$ initializes an empty table $T$ and an empty collection $D$. $\mathcal{A}$ issues the following adaptive query:
- *1*) The adaptive query of outsourcing private key on attribute set $S$
- $\mathcal{C}$ runs the outsourcing private key generation algorithm on the attribute set $S$ to get the outsourcing private key $SK_o$. $\mathcal{C}$ sets $D = D \cup \{S\}$ and sends $SK_o$ to $\mathcal{A}$.
- *2) The adaptive query of the private key on attribute set S*
- $\mathcal{C}$ looks up whether the entry $(S, SK_o, IK, RK)$ in $T$. If such entry exists, $\mathcal{C}$ returns the private key $IK$. Otherwise, $\mathcal{C}$ runs the outsourced private key generation algorithm and the private key generation algorithm adds entry $(S, SK_o, IK, RK)$ table $T$ and returns the private keys $IK$ to $\mathcal{A}$.
- **Challenge:**$\mathcal{A}$ submits two equal-length messages $m_0$, $m_1$ and an access policy $\mathbb{A}^*$ to $\mathcal{C}$. For all $S \in D$, the restriction is that the attribute set$S$cannot satisfy $\mathbb{A}^*$. $\mathcal{C}$ randomly selects $\beta \in \{0, 1\}$, sets $CT^* = Encrypt\{PP, m_\beta, \mathbb{A}^*\}$ and sends $CT^*$ to $\mathcal{A}$.
- **Phase 2:** $\mathcal{A}$ issues the adaptive query of outsourcing private key similar to in Phase 1, with restriction is that the attribute set $S$ does not satisfy the access policy $\mathbb{A}^*$.
- **Guess:** $\mathcal{A}$ outputs guess $\beta' \in \{0, 1\}$ of $\beta$. If $\beta = \beta'$, $\mathcal{A}$ win the game.

The advantage of $\mathcal{A}$ in the above game is defined as $Adv = \left|\Pr[\beta = \beta'] - 1/2\right|$.

*Definition 7: If for all probability polynomial time adversary who can win the above game with negligible advantage. Then VMKS-ABE scheme is selectively secure against chosen plaintext attacks.*

## IV. OUR CONSTRUCTION

In this section, we constructed VMKS-ABE scheme based on the scheme [26] presented by Lai *et al.* The scheme [26] only supports outsourcing decryption, which reduces the computational load in the decryption phase, but the computational load in other phases is still huge and does not support keyword search. In order to reduce the local load, our scheme outsources most of the computational burden of private key generation, encryption, and decryption algorithm to the appropriate service providers; Due to low efficiency of single keyword search, the search results are not accurate. In order to achieve accurate search result, we put forward a multi-keyword search algorithm, which greatly improves the accuracy of search queries and reduces the waste of

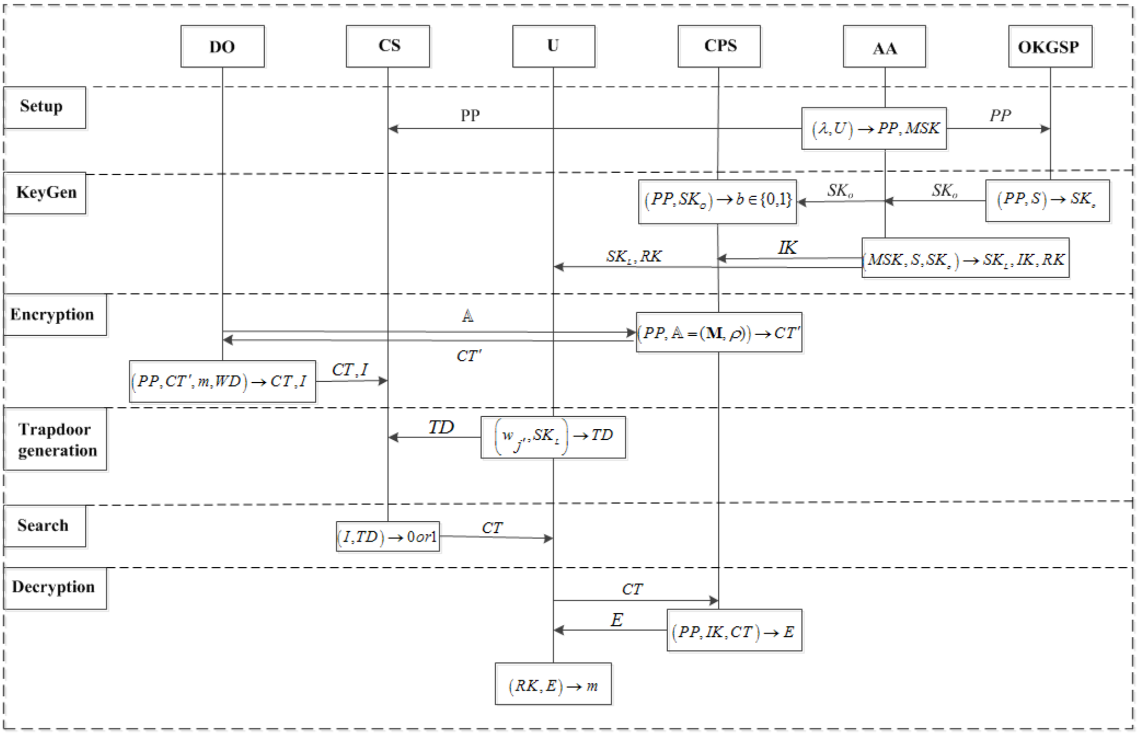

DO
CS
U
CPS
AA
OKGSP
Setup
PP
$(\lambda, U) \rightarrow PP, MSK$
PP
KeyGen
$(PP, SK_o) \rightarrow b \in \{0,1\}$
$SK_o$
$SK_o$
$(PP, S) \rightarrow SK_o$
$SK_L, RK$
$IK$
$(MSK, S, SK_o) \rightarrow SK_L, IK, RK$
Encryption
$\mathbb{A}$
$(PP, \mathbb{A} = (\mathbf{M}, \rho)) \rightarrow CT'$
$CT'$
$(PP, CT', m, WD) \rightarrow CT, I$
$CT, I$
Trapdoor generation
$TD$
$(w_{j'}, SK_L) \rightarrow TD$
Search
$(I, TD) \rightarrow 0 or 1$
$CT$
Decryption
$CT$
$E$
$(PP, IK, CT) \rightarrow E$
$(RK, E) \rightarrow m$

**FIGURE 2.** Flow chart of VMKS-ABE scheme.

computing resources. In addition, we added an algorithm flow chart behind the solution to make the solution clearer.

### 1) SYSTEM SETUP

**Setup** $(\lambda, U) \rightarrow PP, MSK$. The algorithm inputs security parameters $\lambda$ and the universe set of attributes $U = \{1, 2, \cdots, N\}$, where $\lambda$ is the binary size of prime number $p$, $AA$ calls the algorithm to establish the system, $AA$ selects a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, where the $\mathbb{G}$ and $\mathbb{G}_T$ are multiplicative cyclic groups of prime order $p$, $g$ is a generator of $\mathbb{G}$, $AA$ randomly selects $a, \alpha, z \in \mathbb{Z}_p^*$, for each attribute $i \in U$, $AA$ randomly chooses $s_i \in \mathbb{Z}_p^*$. Suppose $H : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$ is a one-way hash function. $AA$ outputs public parameters $PP$ and master key $MSK$ as follows.

$$PP = \{\mathbb{G}, \mathbb{G}_T, p, e, g, g^a, g^z, e(g, g)^\alpha, T_i = g^{s_i} \forall i \in U, H\},$$
$$MSK = \{\alpha, z\}.$$

Afterwards, $AA$ publishes $PP$ and keeps $MSK$ secretly.

### 2) KEY GENERATION

**Outsourcing KeyGen** $(PP, S) \rightarrow SK_O$. $AA$ calculates $g^\alpha$ and sends it to $OKGSP$. The algorithm inputs $PP$ and a set of attributes $S$ (the number of attributes is $n$, where $n \subseteq N$), and it randomly selects $t$ from $\mathbb{Z}_p^*$, then computes

$$K' = g^\alpha g^{at}, K_0' = g^t, K_i' = T_i^t \quad \forall i \in S.$$

The algorithm outputs the outsourced private key $SK_o = \{K', K_0', K_i'\}_{i \in S, n \subseteq N}$ and sends to the $AA$.

**Outsourcing KeyGen verification** $(PP, SK_o) \rightarrow b \in \{0, 1\}$. The algorithm inputs $PP$ and $SK_o$. When $CPS$ receives an authentication request from the $AA$, it verifies the following equation:

$$e(K', g) = e(K_0', g^a) e(g, g)^\alpha, e(K_0', T_i) = e(K_i', g) i \in S.$$

If the above equation holds, the algorithm outputs 1. Otherwise, it outputs 0.

**KeyGen** $(MSK, S, SK_o) \rightarrow SK_L, IK, RK$. The algorithm inputs $PP$, the user's attribute set $S$ (the number of attributes is $n$,where $n \subseteq N$), and $SK_o$. $AA$ randomly selects $y, u \in \mathbb{Z}_p^*$ where $y$ has multiplicative inverse, then $AA$ calculates the local private key $SK_L$, intermediate private key $IK$ and retrieval key $RK$ as follows:

$$SK_L = \{K = g^{yz}, K_1 = g^y\},$$
$$IK = \{\bar{K} = K'^{y/z} g^u, K_0 = K_0'^{y/z}, K_i = K_i'^{y/z}\}_{i \in S, n \subseteq N},$$

where

$$RK = \{z/y, g^u\}.$$

$n$ is the number of attributes of the user.

Finally, $AA$ sends $SK_L$ and $RK$ to $U$, and sends $IK$ to $CPS$.

### 3) ENCRYPTION

**Outsourcing encryption** $(PP, \mathbb{A} = (\mathbf{M}, \rho)) \rightarrow CT'$. The algorithm inputs $PP$ and the access policy $(\mathbf{M}, \rho)$, $\mathbf{M}$ is a $l \times n$ matrix, $\rho$ is a map that associates row of the matrix $\mathbf{M}$ to attributes. $\forall i \in [1, l]$, the algorithm selects random$r_i \in \mathbb{Z}_p^*$, calculates $C_i' = T_{\rho(i)}^{-r_i}, D_i' = g^{r_i}$, it outputs intermediate ciphertext $CT' = \{C_i', D_i', \}_{i \in [1,l]}$.

**Encryption**$\big(PP, CT', m, WD\big) \rightarrow CT, I$

① **Message encryption** $\left(PP, CT', m\right) \rightarrow CT$. The algorithm inputs $PP$, messages $m$ and intermediate ciphertext $CT'$, and sets $\mathbf{v} = \{s, v_2, \cdots, v_n\} \in \mathbb{Z}_p^{*n}$, where $s$ is the secret exponent to be shared, for $\forall i \in [1, l]$, $\mathbf{M}_i$ representing the $i$th rows of $\mathbf{M}$. The algorithm calculates:

$$\lambda_i = \mathbf{M}_i\mathbf{v}$$
$$C = m \cdot e(g, g)^{\alpha s}, \quad C_1 = g^s,\ C_i = g^{a\lambda_i}C_i',\ D_i = D_i'.$$

Afterwards, the algorithm outputs $CT = \{C, C_1, \{C_i, D_i\}_{i\in[1,l]}\}$ and sends to $CS$.

② **Keywords encryption** $(PP, WD) \rightarrow I$. The algorithm inputs $PP$, keyword set $WD$, where $WD$ contains $d$ keywords. The algorithm randomly chooses $r, r_1 \in \mathbb{Z}_p^*$, for any keywords $\forall w_j \in WD(j = 1, 2, \cdots, d)$. It calculates $W_j = g^{H(w_j)zr}$, $W_1 = g^{r_1}$, $W_2 = g^{r_1 z}$, $W_3 = g^{ar}$, where $w_j$ represents $j$th keyword in $WD$. Subsequently, the algorithm outputs the index $I = \{\{W_j\}_{j\in[1,d]}, W_1, W_2, W_3\}$ and transmits to $CS$.

### 4) TRAPDOOR GENERATION

**Trapdoor generation** $\left(w_{j'}, SK_L\right) \rightarrow TD$. The user inputs $SK_L$ and the set of keywords $WD' \subseteq WD$ that he wants to query. For any keyword $w_{j'} \in WD'(j' = 1, 2, \cdots, d')$, the user randomly selects $\beta \in \mathbb{Z}_p^*$ and calculates:

$$T_1 = K^\beta, \quad T_2 = K_1^\beta,\ T_3 = \prod_{j'=1}^{d'} g^{H(w_{j'})z},\ T_4 = g^a$$

where $w_{j'} \in WD'$, $U$ sends trapdoor $TD = \{T_1, T_2, T_3, T_4\}$ to $CS$.

### 5) SEARCH

**Testing** $(I, TD) \rightarrow 0or1$. $CS$ takes trapdoor $TD$ and index $I$ as input. Then $CS$ checks whether the following equation holds:

$$e(T_2, W_2)e(\prod_{k=1}^{d'} W_{j_k}, T_4) = e(T_1, W_1)e(W_3, T_3). \tag{1}$$

Note that the above calculations involve a matching of keywords in the index and keywords in the trapdoor. In the keyword encryption phase, the algorithm encrypts $d$ keywords to generation index. The user generates a trapdoor with respect to $d'$ keywords that he wants to query, where $d' \leq d$. In order to check whether equation (1) holds, the number of selections for choosing $d'$ keywords from$d$ keywords is $C_d^{d'} = \frac{d\times(d-1)\times\cdots(d-d'+1)}{d'!}$, hence, the trapdoor and index need to match at most $C_d^{d'} = \frac{d\times(d-1)\times\cdots(d-d'+1)}{d'!}$ times, if there is one match success, it means that the equation (1) holds. In this case, $CS$ returns 1, otherwise 0.

### 6) DECRYPTION

**Outsourcing decryption** $(PP, IK, CT) \rightarrow E$. The algorithm decrypts ciphertext $CT$ under the access policy $(\mathbf{M}, \rho)$. It inputs $PP$, the intermediate private key $IK$, and $CT$ related to the access policy $(\mathbf{M}, \rho)$. If attribute set $S$ does not satisfy $\mathbb{A}$, the algorithm outputs $\perp$. Otherwise, lets $I = \{i : \rho(i) \in S\}$ and $I \subset \{1, 2, \cdots, l\}$, calculates constant $\omega_i \in \mathbb{Z}_p^*$ such that $\sum_{i\in I} \omega_i\mathbf{M}_i = (1, 0, \cdots, 0)$; then it runs outsource decryption operation as follows:

$$\begin{aligned} E &= \frac{e(C_1, \bar{K})}{(\prod_{i\in I} (e(C_i, K_0)e(K_{\rho(i)}, D_i))^{\omega_i})} \\ &= e(g, g)^{\alpha sy/z}e(g, g)^{su}. \end{aligned}$$

It outputs partially decrypted ciphertext $E$ and sends it to $U$.

**Decryption** $(RK, E) \rightarrow m$. After the user receives $E$ from $CPS$, $U$ inputs the retrieval key $RK$ and partially decrypted ciphertext $E$, calculates $m = \frac{C\cdot e(C_1, g^u)^{z/y}}{E^{z/y}}$. At last outputs the message $m$.

# V. SECURITY ANALYSIS

## A. CORRECTNESS

### 1) CORRECTNESS OF VERIFICATION OF THE SEARCH TESTING

$$e(T_2, W_2)e(\prod_{j=1}^{d'} W_j, T_4) = e(g^{y\beta}, g^{r_1 z})e(\prod_{j=1}^{d'} g^{H(w_j)zr}, g^a),$$

$$e(T_1, W_1)e(W_3, T_3) = e(g^{y\beta z}, g^{r_1})e(g^{ar}, \prod_{j'=1}^{d'} g^{H(w_{j'})z}).$$

If and only if $\{w_{j'}|j' = 1, 2, \cdots, d'\} = \{w_j|j = 1, 2, \cdots, d'\}$, the above two formulas are equal, the first one is the left of equation (1), the second is the right of equation (1), so in this case the equation (1) holds.

### 2) CORRECTNESS VERIFICATION OF OUTSOURCING DECRYPTION

$$\begin{aligned} E &= \frac{e(C_1, \bar{K})}{(\prod_{i\in I} (e(C_i, K_0)e(K_{\rho(i)}, D_i))^{\omega_i})} \\ &= \frac{e(g^s, (g^\alpha g^{at})^{y/z} g^u)}{(\prod_{i\in I} (e(g^{a\lambda_i} T_{\rho(i)}^{-r_i}, g^{ty/z})e(T_{\rho(i)}^{ty/z}, g^{r_i}))^{\omega_i})} \\ &= \frac{e(g^s, (g^\alpha g^{at})^{y/z})e(g^s, g^u)}{(\prod_{i\in I} (e(g^{a\lambda_i}, g^{ty/z})e(T_{\rho(i)}^{-r_i}, g^{ty/z})e(T_{\rho(i)}^{ty/z}, g^{r_i}))^{\omega_i})} \\ &= \frac{e(g^s, (g^\alpha g^{at})^{y/z})e(g^s, g^u)}{(\prod_{i\in I} e(g^{a\lambda_i}, g^{ty/z})^{\omega_i})} \\ &= \frac{e(g, g)^{s\alpha y/z}e(g, g)^{saty/z}e(g^s, g^u)}{e(g, g)^{\sum_{i\in I} \omega_i\lambda_i aty/z}} \\ &= e(g, g)^{s\alpha y/z}e(g^s, g^u). \end{aligned}$$

## B. SECURITY PROOF

We first consider the security of keyword index, we use the security game for the indistinguishable of keyword index against the adaptive chosen keyword attacks.

*Theorem 1: Under the general group model, for any adversary $\mathcal{A}$, let $q$ be a bound on the sum number of group elements it receives from queries it makes to the oracles for the hash function groups $\mathbb{G}$ and $\mathbb{G}_T$, and the bilinear map $e$ in the* *interaction with security game of the indistinguishable of keyword index against the adaptive chosen keyword attacks. The advantage of the adversary in the game of indistinguishable of keyword index against the adaptive chosen keyword attacks is $\mathcal{O}(q^2/p)$.*

*Proof:* We consider the challenger $\mathcal{C}$ and adversary $\mathcal{A}$ to play the following game. $\mathcal{A}$ maintains two lists of pairs,

$$L_{\mathbb{G}} = \{\langle F_{0,l}, \psi_{0,l}\rangle : l = 1, \cdots, \mathcal{T}_0\}$$
$$L_{\mathbb{G}_T} = \{\langle F_{1,l}, \psi_{1,l}\rangle : l = 1, \cdots, \mathcal{T}_1\}$$

where, $F_{0,l}$ and $F_{1,l}$ are multi-variant polynomials for$\mathcal{A}$'s queries. $\psi_{0,l}$ and $\psi_{1,l}$ are random strings in $\{0, 1\}^*$ for the results of each query, where $\psi_{0,l} = \psi_0(F_{0,l})$, $\psi_{1,l} = \psi_1(F_{1,l})$. We initialize $F_{0,l} = 1$, $F_{1,l} = 1$, thus, $g = \psi_0(1)$, $g_T = \psi_1(1)$, $g^x = \psi_0(x)$, $e(g, g)^y = \psi_1(y)$. Now, we present the detailed oracle queries of $\mathcal{A}$ as follows:

*Group action*. Given two operands $\psi_i(x)$ and $\psi_i(y)$, where $x, y \in \mathbb{Z}_p$, $i \in \{1, 2\}$, if $\psi_i(x)$ and $\psi_i(y)$ are not in the list $L_{\mathbb{G}}$ and $L_{\mathbb{G}_T}$, return $\perp$; otherwise, $\mathcal{C}$ calculates $F = x + y(\bmod p)$ and checks whether $F$ is in the list $L_{\mathbb{G}}$ and $L_{\mathbb{G}_T}$ If so, $\mathcal{C}$ returns $\psi_i(F)$; otherwise, $\mathcal{C}$ sets $\psi_i(F)$ to a random string in $\{0, 1\}^*$ distinct from any strings already in $L_{\mathbb{G}}$ and $L_{\mathbb{G}_T}$. Finally, $\mathcal{C}$ adds $\langle F, \psi_i(F)\rangle$ to $L_{\mathbb{G}}$ and $L_{\mathbb{G}_T}$ and replies to $\mathcal{A}$ with the string $\psi_i(F)$.

*Bilinear pairing*. Given two operands $\psi_0(x)$ and $\psi_1(y)$, if $\psi_0(x)$ and $\psi_1(y)$ are not in the list $L_{\mathbb{G}}$ and $L_{\mathbb{G}_T}$, return $\perp$; otherwise, $\mathcal{C}$ calculates $F = xy(\text{mod} p)$ and checks whether $F$is in the list $L_{\mathbb{G}}$ and $L_{\mathbb{G}_T}$. If so, $\mathcal{C}$ returns $\psi_i(F)$; otherwise, $\mathcal{C}$ sets $\psi_i(F)$ to a random string in $\{0, 1\}^*$ distinct from any strings already in $L_{\mathbb{G}}$ and $L_{\mathbb{G}_T}$. Finally, $\mathcal{C}$ adds $\langle F, \psi_i(F)\rangle$ to $L_{\mathbb{G}}$ and $L_{\mathbb{G}_T}$and replies to $\mathcal{A}$ with the string $\psi_i(F)$.

With the above basic group operations, the security interactive game of challenger $\mathcal{C}$ and adversary $\mathcal{A}$ as follows:

1) Challenger $\mathcal{C}$ randomly selects $a, \alpha, z \in \mathbb{Z}_p^*$, $H : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$, where $H$ is hash function. Then $\mathcal{C}$ sets $PP = \{g^a, g^z, e(g, g)^\alpha, H\}$ and sends to $\mathcal{A}$.
2) $\mathcal{A}$ issues trapdoor queries with respect to the keyword set $WD' = \{w_{j'}\}_{j' \in [1,d']}$ and randomly selects $z, y, t \in \mathbb{Z}_p^*$. $\mathcal{C}$ calculates $K = g^{yz}$, $K_1 = g^y$ to obtain private key $SK = \{K, K_1\}$. Afterwards, $\mathcal{C}$ generates the trapdoor of keyword set $WD'$.
$$TD = \{T_1{=}K^t, T_2 = K_1^t, T_3 = \prod_{j'=1}^{d'} g^{H(w_{j'})z}, T_4 = g^a\}.$$
Finally, $\mathcal{C}$ sends the trapdoor $TD$ to $\mathcal{A}$.
3) $\mathcal{A}$ submits keyword sets $WD_0 = \{w_{j_0}\}_{j \in [1,d]}$ and $WD_1 = \{w_{j_1}\}_{j \in [1,d]}$ as challenge. In addition, the adversary $\mathcal{A}$ gives a challenge access policy $\mathbb{A}$. Subsequently, $\mathcal{C}$ throws a fair coin to choose $b \in \{0, 1\}$, generates a challenge index of $WD_b$. It randomly selects $r', r'' \in \mathbb{Z}_p^*$, lets
$$W_j = g^{H(w_{j_b})z^{r'}}, W_1 = g^{r'}, W_2 = g^{r''z}, W_3 = g^{ar'}.$$
The challenge index is $I^* = \{\{W_j\}_{j \in [1,d]}, W_1, W_2, W_3\}$, $\mathcal{C}$ sends the challenge index to $\mathcal{A}$.
4) $\mathcal{A}$ repeats the queries similar to phase 2) with the restriction that $WD_0$ and $WD_1$ can no longer be queried.
5) Finally, $\mathcal{A}$ outputs guess $b'$ of $b$, where $b' \in \{0, 1\}$.

**TABLE 1.** Possible items in the random oracle query group $\mathbb{G}$.

| $a$ | $yt$ | $r''$ |
|---|---|---|
| $z$ | $yzt$ | $ar'$ |
| $y$ | $H(w_{j'})z$ | $r''z$ |
| $yz$ | $H(w_{j_b})zr'$ | |

In the above security game, the adversary can query at most $q$ times. Specifically, the adversary needs to distinguish $W_j = g^{H(w_{j1})z^r\prime}$ and $W_j = g^{H(w_{jb})z_{r\prime}}$. We can consider a modified game, which uses $W_j = g^\theta$ instead of $W_j = g^z r\prime$ in real challenge index. The adversary needs to distinguish between $g^\theta$ and $g^z r\prime$, where $\theta$ is randomly choose from $\mathbb{Z}_p^*$. The probability of distinguishing $g^\theta$ from $g^z r\prime$ is half of the probability of distinguishing $W_j = g^{H(w_{j_1})zr\prime}$ from $W_j = g^{H(w_{j_0})zr\prime}$.

Next, we will conduct a detailed analysis of the $\mathcal{C}'$ simulation. In the general group model, as long as there is no unexpected collision, the $\mathcal{C}'$ simulation is perfect. That is, we think of an oracle query as being a rational function$\delta = \eta/\xi$ in the variable $\theta, \alpha, a, z, t, r', r''$. When two queries correspond to different rational functions, due to the random selection of the values of these variables, the rational function will have an unexpected collision, that is, When $\eta \neq \eta'\xi \neq \xi'$, then $\delta = \eta/\xi = \eta'/\xi' = \delta'$.

Our current condition is that such accidental collision do not occur in $\mathbb{G}$ or $\mathbb{G}_\text{T}$. For any pair of queries within a group, which associated with different rational functions $\eta/\xi$ and $\eta'/\xi'$, a collision happens if and only if non-zero polynomials $\eta\xi' - \xi\eta' = 0$. Based on the article [35], the probability of a collision occurring is defined as $\mathcal{O}(1/p)$. Under the constraint conditions, the probability of any such collision occurring is at most $\mathcal{O}(q^2/p)$. Therefore, we assume that no such collision occurs, and its probability is defined as $1 - \mathcal{O}(q^2/p)$.

In Table 1, we enumerate all rational function queries in $\mathbb{G}$, which uses $\theta$ instead of $zr'$ in regard to the analysis of $\mathcal{C}'$ simulation, because $\theta$ only exists in$W_j = g^\theta$, if a collision happens, there is $\gamma \neq 0$, we have$\delta - \delta' = \gamma zr' - \gamma\theta$. Our analysis shows that it is almost impossible for adversary $\mathcal{A}$ to construct an inquiry about$\gamma zr'$:

If a collision occurs, then $\gamma zr' = \gamma\theta + \delta' - \delta$, for any elements in group $\mathbb{G}$, it can be seen from Table 1 that for the inquiry of $\delta$ in $g^\delta = \psi(\delta)$ and $\delta'$ in $g^{\delta'} = \psi(\delta')$, the probability that the above formula holds is $\mathcal{O}(1/p)$. So the probability of a collision occurs is negligible.

Therefore, $\mathcal{A}$ is almost impossible to construct an inquiry about $\gamma zr'$.

Next we consider the security of our scheme for its indistinguishable of ciphertext under the chosen plaintext attacks (CPA).

*Theorem 2: If CP-ABE scheme [14] is selectively secure against CPA, then our scheme is selectively secure under chosen plaintext attacks (CPA).*

*Proof:* Assuming that an adversary $\mathcal{A}$ can break our scheme with a non-negligible advantage under the selectively chosen plaintext attacks model, then we can construct an algorithm $\mathcal{B}$ that can break the scheme [14] with a non-negligible advantage under the selectively chosen plaintext attacks model.

Let $\mathcal{C}$ be the challenger corresponding to $\mathcal{B}$ in the selectively CPA-secure game of literature [14]. Performs the following steps:

*Initialization:* $\mathcal{A}$ gives $\mathcal{B}$ a challenge access policy $\mathbb{A}^*$. $\mathcal{B}$ transmits $\mathbb{A}^*$ to $\mathcal{C}$ as its challenge access policy and $PP$ of scheme [14] are given.

$$PP' = \{p, \mathbb{G}, \mathbb{G}_T, e, g^a, e(g, g)^\alpha, T_i = g^{s_i} \forall i \in U\}.$$

*Create:* $\mathcal{B}$ randomly selects $z \in \mathbb{Z}_p^*$, and an anti-collision hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$, then $\mathcal{B}$ sends $PP$ to $\mathcal{A}$.

$$PP = \{p, \mathbb{G}, \mathbb{G}_T, e, g^a, g^z, e(g, g)^\alpha, T_i = g^{s_i} \forall i \in U, H\}.$$

*Phase1:* $\mathcal{B}$ initializes an empty table $T$ and an empty collection $D$. $\mathcal{A}$ issues the following adaptively queries:

1) *The adaptive inquire of outsourcing private key on attribute set S*

   When $\mathcal{A}$ issues an adaptive query of the outsourcing private key associated with attribute set $S$, $\mathcal{B}$ informs $\mathcal{C}$ to execute outsourcing private key generation algorithm to get the outsourcing private key $SK_o$. Then $\mathcal{B}$ sets $D = D \cup \{S\}$ and sends $SK_o$ to $\mathcal{A}$.

2) *The adaptive inquire of private key on attribute set S*

$\mathcal{B}$ looks up whether the entry $(S, SK_o, IK, RK)$ in the table $T$. If such entry exists, $\mathcal{B}$ returns the private key $IK$. Otherwise, $\mathcal{B}$ selects random value $t, y, , u, z \in \mathbb{Z}_p^*$, sets

$$\overline{K} = g^{y/z} g^{at} g^u, K_0 = g^t, K_i = T_i^t \forall i \in S.$$

Finally, $\mathcal{B}$ adds the entry $(S, IK = \{\overline{K}, K_0, K_i\}, z/y, g^u)$ to the table $T$ and forward $IK$ to $\mathcal{A}$. Note that, $\mathcal{B}$ does not know the actual retrieving key $RK = \{\alpha z/y, g^u\}$.

*Challenge:* The adversary $\mathcal{A}$ commits two same length messages $m_0$ and $m_1$. $\mathcal{B}$ selects random bit $\beta \in \{0, 1\}$, two random messages $\tilde{m}_0$ and $\tilde{m}_1$ as well as an access policy $\mathbb{A}^*$, then seeds them to $\mathcal{C}$. $\mathcal{C}$ chooses a random bit $\gamma \in \{0, 1\}$, encrypts message $\tilde{m}_\gamma$ under the public parameters $PP$ and access policy $\mathbb{A}^*$ by using the encryption algorithm of the scheme [14], and transmits ciphertext $CT^*$ to $\mathcal{B}$. Afterwards, $\mathcal{B}$ selects a random vector $\mathbf{v} = \{s, v_2, \cdots, v_n\}$. For each row $\mathbf{M}_i$ of the matrix $\mathbf{M}$, $\mathcal{B}$ randomly selects $r_i' \in \mathbb{Z}_p^* i \in [1, l]$ and sets $C = m_\beta \cdot e(g, g)^{\alpha s}, C_1 = g^s, C_i = g^{a\lambda_i} T_{\rho(i)}^{-r_i'}, D_i = g^{r_i'}$.

In the end, the challenge ciphertext is $CT^* = \{C, C_1, C_i, D_i\}_{i \in [1,l]}$, which is sent to $\mathcal{A}$.

**TABLE 2.** Scheme function comparison.

| Schemes | Encryption outsourcing | Decryption outsourcing | Multiple keyword search | Access policy |
|---|---|---|---|---|
| [26] | No | Yes | No | LSSS |
| [36] | Yes | No | Yes | Tree |
| [37] | No | Yes | No | LSSS |
| [38] | No | Yes | Yes | LSSS |
| Our scheme | Yes | Yes | Yes | LSSS |

*Phase 2:* The adversary $\mathcal{A}$ issues the adaptive query of outsourcing key similar to Phase 1 with restriction is that the attribute set $S$ cannot satisfy the access policy $\mathbb{A}^*$ and $S$ is not included in $D$. $\mathcal{B}$ responds inquiries similar to Phase 1.

*Guess:* $\mathcal{A}$ outputs guesses $\beta' \in \{0, 1\}$ of $\beta$. $\mathcal{B}$outputs guess $\beta' \in \{0, 1\}$ of $\gamma$.

If $\beta = \gamma$, $\mathcal{B}$ give a perfect simulation for game. Therefore, if $\mathcal{A}$ can break our scheme with non-negligible advantage, Then we construct the algorithm $\mathcal{B}$ can be solved in the literature [14] with a non-negligible advantage. □

## VI. PERFORMANCE ANALYSIS

In this section, in order to analysis the function and performance of this scheme, we compares our scheme with the schemes in literature [26], [36]–[38] for encryption outsourcing, decryption outsourcing, multi-keyword search and access policy. The specific comparison items and results are shown in Table 2.

The schemes [26], [37] only support decryption outsourcing; the scheme [36] implements multi-keyword search; the scheme [38] support decryption outsourcing and multi-keyword search; our VMKS-ABE scheme in cloud storage supports both encryption and decryption outsourcing as well as multi-keyword search.

We give a comparison of local computational cost of our scheme with the schemes in [26], [36]–[38]. The results are showed in Table 3, which $E$ represents the exponentiation operation in the calculation, $P$ represents the pair operation in the calculation, $n$ is the number of attributes of the user, $L$ the number of attributes in the policy, and $j$ the number of keywords in the middle of encryption, $j'$ represents the number of keywords in search. From table 3 we can see in the private key generation phase, the amount of calculation increases linearly with the number of attributes of the user; in the encryption phase, [26], [37], [38] does not support encryption outsourcing, and their calculation amount grows linearly with the number of attributes in the access policy.

Our scheme outsources part of the encryption task to *CPS* which is similar to that [36], but the scheme in [36] does not have a decryption phase and only supports multiple keyword searches. In the decryption phase, although [26], [37], [38] support decryption outsourcing,[26], [37] decryption time is related to the number of attributes in the access policy, but our scheme and [38] decryption time is constant and our scheme takes less time than [38].

Fig 3 is a comparison of our scheme with other schemes in the private key generation phase. As we can be seen from

**TABLE 3.** Comparison of calculations quantities.

| Schemes | KeyGen | Encryption | Decryption | Search | Security |
|---|---|---|---|---|---|
| [26] | $(n+3)E$ | $(6L+6)E$ | $(4L+2)P$ | --- | CPA |
| [36] | $(2n+2)E$ | $(2L+3+j)E$ | --- | $(j'+6)P$ | CKA |
| [37] | $(2n+1)E$ | $(2L+1)P+3LE$ | $LE$ | --- | CPA |
| [38] | $2nE$ | $(2L+2)E+(j+2)P$ | $2P$ | $(2L+j')P$ | CKA&CPA |
| Our scheme | $(n+6)E$ | $(L+6+j)E$ | $P+2E$ | $(j'+3)P$ | CKA&CPA |

---: no such operation in the scheme

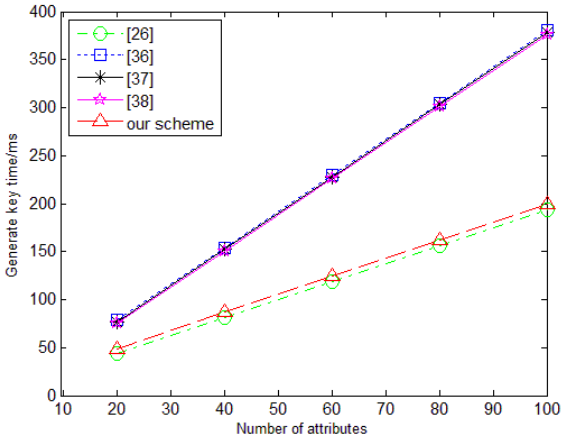


**FIGURE 3.** Key generation time.

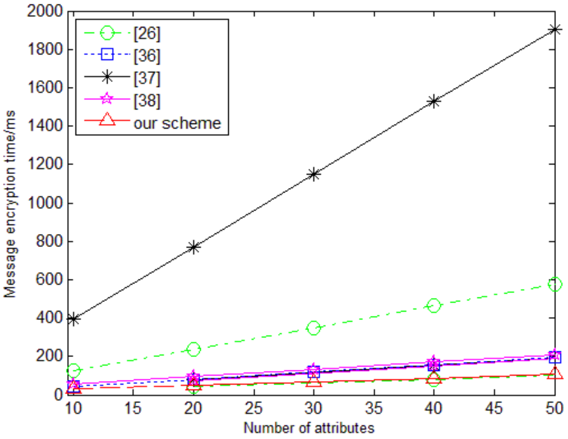


**FIGURE 4.** Encrypted message time.

Fig 3, the key generation time of our scheme is smaller than that in [26], [36]–[38] as the number of user's attributes increases.

Encryption phase: our encryption phase is divided into keywords encryption and message encryption. In the message encryption phase, the comparison result of message encryption is shown in Fig 4. It can be seen from Fig 4 that the encryption time of our scheme wins over that of other schemes. In the keywords encryption phase, the comparison result is shown in Fig 5. It can be seen from Fig 5 that our scheme is better than the scheme in [36], [38].

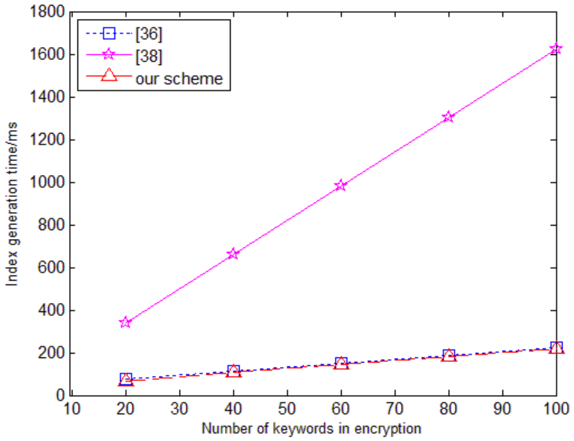


**FIGURE 5.** Encryption keyword time.

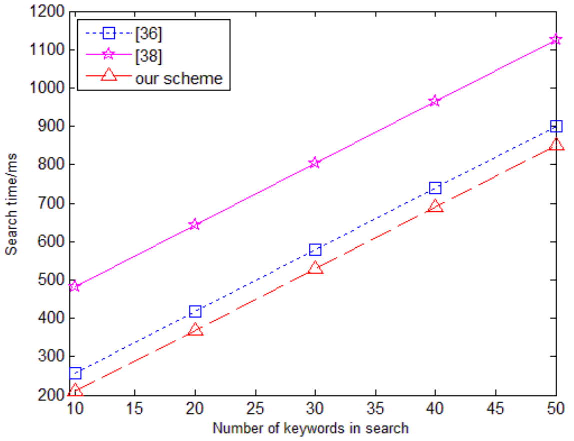


**FIGURE 6.** Search time.

The search phase: we compare the search phase of our scheme with the scheme in [36], [38] to get Fig 6. As it can be seen from Fig 6, the search phase of our scheme requires less time under the same query.

Decryption phase: we compare the decryption phase of our scheme with the schemes in [26] and [37], [38] to obtain Fig 7, from Fig 7 shows that the schemes in [26] and [37], [38] are less efficient than our scheme.

Therefore; from a general perspective, the performance of our scheme proposed in this paper has improved, and our new system is more suitable for practical usage.

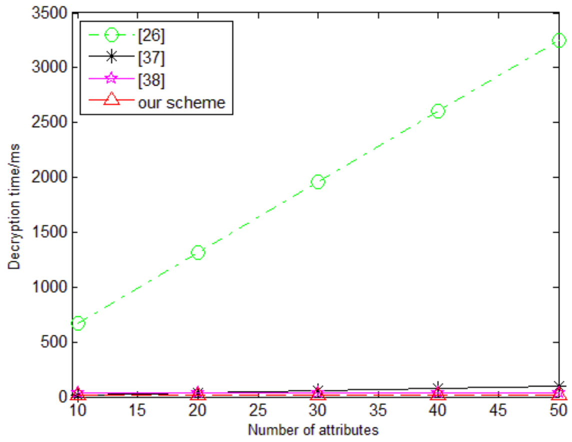


**FIGURE 7.** Decryption time.

## VII. CONCLUSION

In this article we proposed VMKS-ABE scheme. In our scheme, we combine the verifiable of the correctness of outsourced private key with multi-keyword search based on attribute encryption. In the general group model, the security of keyword index is proved. Under the random oracle model, the ciphertext is proved to be selectively secure.

Since the security in the general group model is much weak than in the standard model, it is worth constructing verifiable and multi-keyword searchable scheme in the standard model.

## REFERENCES

[1] D. X. Song, D. Wanger, and A. Perrig, "Practical techniques for searches on encrypted data," *Proc. IEEE Symp. Secur. Privacy*, Berkeley, CA, USA, May 2000, pp. 44–55.

[2] C. Dong, G. Russello, and N. Dulay, "Shared and searchable encrypted data for untrusted servers," in *Data and Applications Security XXII*, Berlin, Germany: Springer, Jul. 2008, pp. 127–143.

[3] S. Li and M. Xu, "Attribute-based public encryption with keyword search," *Chin. J. Comput.*, vol. 37, no. 5, pp. 1017–1024, Jun. 2014. doi: 10.3724/SP.J.1016.2014.01017.

[4] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," *J. Comput. Secur.*, vol. 19, no. 5, pp. 895–934, 2011.

[5] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in *Advances in Cryptology (CRYPTO)*. Berlin, Germany: Springer, Aug. 2007, pp. 535–552.

[6] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–5. doi: 10.1109/INFCOM.2010.5462196.

[7] 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. doi: 10.1109/TPDS.2014.2355202.

[8] Q. Dong, Z. Guan, and Z. Chen, "attribute-based keyword search efficiency enhancement via an online/offline approach," in *Proc. IEEE 21st Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2015, pp. 298–305.

[9] Y. Ye, 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.

[10] Q. Chai and G. Gong, "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 917–922.

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

[12] 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.*, Oct. 2006, pp. 89–98.

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

[14] B. Waters, "ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography-PKC*, Berlin, Germany: Springer, Mar. 2011, pp. 53–70.

[15] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. USENIX Secur. Symp.*, Berkeley, CA, USA, Aug. 2011, pp. 34–49.

[16] 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.

[17] J. Li, H. Wang, Y. Zhang, and J. Shen, "Ciphertext-policy attribute-based encryption with hidden access policy and testing," *KSII Trans. Int. Inf. Syst.*, vol. 10, no. 7, pp. 3339–3352, Jul. 2016.

[18] 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. doi: 10.1109/JIOT.2018.2825289.

[19] Z. Wan, J. Liu, and R. H. Deng, "HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 2, pp. 743–754, Apr. 2012.

[20] H. Deng *et al.*, "Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts," *Inf. Sci.*, vol. 275, pp. 370–384, Aug. 2014.

[21] D. Li, C. Jie, J. Liu, Q. Wu, and W. Liu, "Efficient CCA2 secure revocable multi-authority large-universe attribute-based encryption," *Int. Symp. Cyberspace Saf. Secur.*, Oct. 2017, pp. 103–118.

[22] 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.

[23] S. Wang, K. Guo, and Y. Zhang, "Traceable ciphertext-policy attribute-based encryption scheme with attribute level user revocation for cloud storage," *PLoS ONE*, vol. 13, no. 9, Sep. 2018, Art. no. e0203225. doi: 10.1371/journal.pone.0203225.

[24] H. Wang, Z. Zheng, L. Wu, and Y. Wang, "Adaptively secure outsourcing ciphertext-policy attribute-based encryption," *J. Comput. Res. Develop.*, vol. 52, no. 10, pp. 2270–2280, Dec. 2015. doi: 10.7544/issn1000-1239.2015.20150497.

[25] R. Zhang, H. Ma, and Y. Lu, "Fine-grained access control system based on fully outsourced attribute-based encryption," *J. Syst. Softw.*, vol. 125, pp. 344–353, Mar. 2017. doi: 10.1016/j.jss.2016.12.018.

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

[27] X. Mao, J. Lai, Q. Mei, K. Chen, and J. Weng, "Generic and efficient constructions of attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Depend. Sec. Comput.*, vol. 13, no. 5, pp. 533–546, May 2016.

[28] J. Li, X. Huang, J. Li, X. Chen, and Y. Xiang, "Securely outsourcing attribute-based encryption with checkability," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2201–2210, Aug. 2014. doi: 10.1109/TPDS.2013.271.

[29] H. Wang, D. He, J. Shen, Z. Zheng, C. Zhao, and M. Zhao, "Verifiable outsourced ciphertext-policy attribute-based encryption in cloud computing," *Soft Comput.*, vol. 21, no. 24, pp. 7325–7335, Jun. 2017.

[30] J. Li, Y. Wang, Y. Zhang, and J. Han, "Full verifiability for outsourced decryption in attribute based encryption," *IEEE Trans. Services Comput.*, to be published. doi: 10.1109/TSC.2017.2710190.

[31] 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.

[32] 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.

[33] D. Zheng, A. Wu, Y. Zhang, and Q. Zhao, "Efficient and privacy-preserving medical data sharing in Internet of things with limited computing power," *IEEE Access*, vol. 6, pp. 28019–28027, May 2018. doi: 10.1109/ACCESS.2018.2840504.

[34] B. Dan, X. Boyen, and E. J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, May 2005, pp. 440–456.

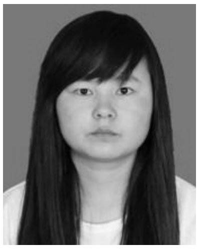
[35] J. T. Schwartz, "Fast polynomial algorithms for verification of polynomial identities," *J. Assoc. Comput.*, vol. 27, no. 4, pp. 701–717, Jan. 1980. doi: 10.1007/3-540-09519-5_72.

[36] Y. Miao, J. Ma, X. Liu, F. Wei, Z. Liu, and X. A. Wang, "m$^2$-ABKS: Attribute-based multi-keyword search over encrypted personal health records in multi-owner setting," *J. Med. Syst.*, vol. 40, no. 11, pp. 246–258, Nov. 2016. doi: 10.1007/s10916-016-0617-z.

[37] K. Zhang, J. Ma, J. Liu, and H. Li, "Adaptively secure multi-authority attribute-based encryption with verifiable outsourced decryption," *Sci. China Inf. Sci.*, vol. 59, no. 9, Aug. 2016, Art. no. 099105. doi: 10.1007/s11432-016-0012-9.

[38] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 715–725, Oct. 2017. doi: 10.1109/TSC.2016.2542813.

**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, in 1989, and the Ph.D. degree in cryptology from Xidian University, Xi'an, in 2003. He is currently a Professor with the Xi'an University of Technology. His current research interests include cryptography and information security.



**SHASHA JIA** received the B.S. degree, in 2016. She is currently the M.S. degree with the Xi'an University of Technology, Xi'an, China. Her research interests include information security and modern cryptography.

**YALING ZHANG** received the B.S. degree in computer science from Northwest University, Xi'an, China, in 1988, and the M.S. degree in computer science and the Ph.D. degree in mechanism electron engineering from the Xi'an University of Technology, Xi'an, in 2001 and 2008, respectively, where she is currently a Professor. Her current research interests include cryptography and network security.