Revocable Attribute-Based Encryption for Multi-Keyword Search in Clouds

Chun-I Fan, Si-Jing Wu, and Yi-Fan Tseng

Abstract—With the rapid advancement of cloud computing, users upload their files to the cloud server so that any user can access it remotely. To assure the data security, the data owner, typically, encrypts the data before outsourcing them to the cloud server. In addition, an encryption mechanism needs to enable the consumers to perform efficient searches of such encrypted data in the cloud storages through keywords, i.e. searchable encryption. However, most of searchable encryption is improper due to several limitations, such as the requirement of an online fully trusted third party, poor efficiency, high-overhead in user revocation, support of a single keyword search, etc. To mitigate such limitations, an attribute-based encryption scheme with fine-grained multi-keyword search is proposed. The new scheme supports the user revocation. In addition, the length of the ciphertext as well as the secret key do not grow linearly under the influence of the size of attribute set. The performance of the proposed scheme is better as compared to other related schemes. Hence, one can easily adopt the proposed scheme for the real life applications due to its flexibility in terms of its features, security and efficiency.

Index Terms—Attribute-Based Encryption, Multi-Key word Search, User Revocation, Fine-Grained Search, Clouds

I. Introduction

With the expansion of cloud computing, more and more users upload data to cloud servers, which makes cloud storage services play an important role in modern society. Since the classified information is outsourced to cloud servers, data owners often concern about the privacy of their data. As a result, data owners encrypt their private data before outsourcing them to cloud servers. On the other hand, how the data users use keywords to effectively search for the required information in a large number of encrypted files is also an important issue. We show the mentioned scenario in Figure 1. For the sake of keeping cloud servers from knowing any keyword information, the establishment of the credential data keyword index is regarded as a basic means. Such a technique can be adopted in a class of cryptographic primitives called searchable encryption.

Searchable encryption (SE) was primarily introduced by Song et al. [1] in 2000. In an SE scheme, a cloud server is allowed to search for encrypted files without revealing any information in keywords or plaintext data because the data owner can encrypt the potential keywords before uploading them with encrypted files. To avoid information leakage of keywords during searching the encrypted data, Boneh et al. [2] in 2004 presented a concept of public key encryption with single-keyword search system (PEKS) which can be adopted in the public key setting. Unfortunately, the scheme failed to achieve fine-grained access control on encrypted files. Thereafter, Li et al. [3] in 2010 proposed a fuzzy keyword search scheme using matching approximation of the classified data and the embedded keywords. For practical usability, Cao et al. [4] in 2014 presented a multi-keyword sequence search scheme, which enables data users to search in multiple keywords and receive results sorted by relevance. In 2015, Zheng et al. [5] proposed a certificateless keyword search scheme, but the scheme does not support the fine-grained search. In order to make the solutions more suitable for cloud servers, there are some key security challenges in terms of enhancing the search efficiency, search capabilities, and system security.

To provide flexibleness for accessing files, most applications use sophisticated access control mechanisms. Due to the fine-grained access control policy, attribute-based encryption (ABE) scheme is appropriate for the purpose mentioned above. Sahai and Waters [6] in 2005 primarily introduced the concept of ABE, which enables users to implement fine-grained access controls on the encrypted sensitive data. Following their precursory work, Goyal et al. [7] in 2006 presented two different types of ABE: key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). In KP-ABE schemes, each user’s private key is related to an access policy, and the ciphertext is associated with a set of attributes. A secret key can be

Fig. 1. The Scenario about the File Sharing in Clouds.
used to decrypt a ciphertext if and only if the attribute set associated with the ciphertext satisfies the access policy related to the user’s private key. The situation in CP-ABE schemes is the opposite. For multiple data users in a system sharing the confidential information, CP-ABE is more flexible than KP-ABE owing to the nature of CP-ABE. Hence, we focus on the ciphertext-policy setting in our work.

Based on the concept of ABE, Sun [8] and Zheng [9] in 2014 independently presented attributed-based encryption keyword search (ABKS) schemes that enable the data owner to decide the policy, which is related to the decision of whether a data user can decrypt and search the keyword as shown in Figure 2. However, there are three challenges in ABKS schemes needed to be addressed; first, how to prevent revoked users from decrypting files; second, how to avoid the size of the secret key and ciphertext increasing linearly with the number of attributes; third, how to search for the information effectively in the vast amount of data. To be implemented in the multi-owner application, Miao et al. [10] in 2016 proposed an ABKS scheme that supports multi-keyword search. Nevertheless, the scheme fails to protect the private information of the users’ attributes and we achieve higher efficiency for decryption because of using only one pairing. Thus, our scheme is more suitable for real-world cloud environments.

II. Preliminaries

In this section, we review the background knowledge about our scheme along with the properties of bilinear maps. Besides, we discuss the related mathematical assumptions and the access control structure in the scheme.

A. Bilinear Maps

In this section, we define the bilinear maps with its essential properties.

Definition II.1. Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative cyclic groups of prime order $p$, and $g$ be a generator of $\mathbb{G}$. A bilinear map $e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ satisfies the following properties:

- **Bilinearity:** $e(g^a, g^b) = e(g, g)^{ab}$, and $a, b \in \mathbb{Z}_p$.
- **Non-Degeneracy:** $e(g, g) \neq 1$.
- **Computability:** There exists an efficient algorithm to compute $e$.

B. Access Structure

In the proposed scheme, we use a series of "AND gates" on multi-value attributes as an access control structure.

Definition II.2. Let the total number of attributes be $n$. Let $U = \{u_1, u_2, ..., u_n\}$ be the universe attribute list, $V_i = \{v_{i,1}, v_{i,2}, ..., v_{i,j}\}$ be a set of possible values for $u_i$, where $j$ is the number of the possible values for $u_i$. Let $A = \{x_1, x_2, ..., x_n\}$ be the attribute list for a data user, where $x_i \in V_i$, and $S = \{s_1, s_2, ..., s_n\}$ be an access structure in the ciphertext, where $s_i \in V_i$. We denote that the user attribute list $A$ satisfies the access policy $S$ if and only if $x_i = s_i$, for all $i \in [1, n]$.

We give an example for the better comprehension of the access structure we use. Consider a university as the scenario. In such scenario, the universe attribute list would be $U = \{u_1, u_2, u_3\} = \{\text{Department}, \text{Position}, \text{Gender}\}$. The possible values of $u_1 = \text{Department}$ would be (“CS”, “EE”, ...), and the possible values of $u_2 = \text{Position}$ would be (“Professor”, “Student”, ...). An access structure would be like $S = \{\text{Department} = \text{CS}, \text{Position} = \text{Professor}\}$. A user with attribute list (“CS”, “Position”, “Male”) or (“CS”, “Position”, “Female”) will satisfy the access structure $S$. However, a user with attribute list (“EE”, “Student”, “Male”) will not pass the access structure. Besides, the attribute “name”, i.e. “Department” and “Position”, will be appended in the ciphertext, only the values, “CS” and “professor”, will be hidden. The reason of this setting is that we want to achieve

A. Contributions

Our construction is inspired by a selective-ID secure identity-based encryption (IBE) scheme presented by Boneh et al. [14] in 2004. We focus on the challenges in ABKS scheme we mentioned above and design revocable CP-ABE scheme with the fine-grained multi-keyword search. It supports not only constant-size secret key but also efficient decryption with only one pairing, which realizes a more flexible implementation.

Fig. 2. The Scenario about the ABKS Scheme.
more flexible access structure. If the attribute “name” is hidden as well, then all the values of a user must fit the values of the access structure. In such case, the access structure shown in the above scenario, i.e. “professor of CS department, regardless of gender”, cannot be achieved if the attribute “name” is hidden as well.

C. The Generalized DDH Assumption

In this section, we show the definition of the generalized decisional Diffie-Hellman assumption \[15\].

Definition II.3. Given \((g, g^a, g^b, g^c, Q)\) for \(a, b, c \in \mathbb{Z}_p\) and \(g \in \mathbb{G}\), decide whether \(Q = g^{abc}\) or a uniformly random element \(R \in \mathbb{G}\). A polynomial-time adversary \(A\) has an advantage \(\epsilon\) in solving the generalized decisional Diffie-Hellman problem if
\[
\Pr[C(g, g^a, g^b, g^c, Q = g^{abc})] - \Pr[C(g, g^a, g^b, g^c, Q = R)] \geq \epsilon.
\]

D. Identity-Based Encryption (IBE) and its Security Model

Since the proposed work is motivated from selective-ID secure IBE scheme, we briefly define its four algorithms and the underlined security model below.

- **Setup** \((1^*):\) The algorithm takes as inputs a security parameter \(1^*\). The private key generator (PKG) executes this algorithm to output the public parameter \(\text{param}\) and the secret key \(\text{MSK}\).
- **KeyGen** \((\text{MSK}, \text{ID}):\) The algorithm takes as inputs the secret key \(\text{MSK}\) and the public key \(\text{ID}\). The PKG runs it to output the secret key \(d_{\text{ID}}\) related to the given identity.
- **Encrypt** \((\text{param}, \text{M}, \text{ID}):\) The algorithm takes as inputs the public parameters \(\text{param}\), and a message file \(\text{M}\) under the public key \(\text{ID} \in \mathbb{Z}_p\). The data owner runs it to output a ciphertext \(\text{C}\).
- **Decrypt** \((d_{\text{ID}}, \text{C}):\) The algorithm takes as inputs the ciphertext \(\text{C}\) and the secret key \(d_{\text{ID}}\). The data user runs it to output the message file \(\text{M}\).

Next we provide the IND-sID-CPA (indistinguishable selective-ID-chose plaintext attacks) security model for an IBE scheme \[14\] which our scheme inherits to construct the proposed scheme.

Definition II.4 (IND-sID-CPA Security for IBE).

1) **Initialization:** The adversary \(A\) outputs a target identity \(ID^*\) to the challenger \(C\).
2) **Setup:** \(C\) runs **Setup** algorithm to produce public parameters and a master secret key, while sending the public parameters \(\text{param}\) to \(A\).
3) **Phase 1:** \(A\) is able to issue queries polynomially to \(C\) for private keys, and \(C\) responds by running **KeyGen** algorithms with the restriction that \(ID_i \neq ID^*\) and then sends the results to \(A\).
4) **Challenge:** \(A\) commits two equal-length messages \(M_0\) and \(M_1\) where \(M_0 \neq M_1\), \(C\) randomly selects \(\rho \in \{0, 1\}\), and runs **Encrypt** algorithm to send the ciphertext \(C_{\rho \text{h}} = \text{Encrypt}(\text{param}, ID^*, M_\rho)\) to \(A\).
5) **Phase 2:** The queries here are similar to the ones in Phase 1. \(A\) continues to query with the restriction that \(ID_i \neq ID^*\).
6) **Guess:** \(A\) outputs a guess as \(\rho' \in \{0, 1\}\). \(A\) wins the game if \(\rho' = \rho\).

In the game, we define the advantage of \(A\) in winning the game as follows:

\[
\text{Adv}^{\text{IND-sID-CPA}}_A = \left| \Pr[\rho' = \rho] - \frac{1}{2} \right|
\]

If \(\text{Adv}^{\text{IND-sID-CPA}}_A\) is negligible for every polynomial-time \(A\), the identity-based encryption scheme is said to be IND-sID-CPA secure.

E. Revocable Attribute-Based Encryption with Keyword Search and its Security Model

We define the attribute set as \(A = \{x_1, x_2, ..., x_n\}\) corresponding to the values of the attributes named \(1, ..., n\), and the keyword set as \(W = \{w_1, w_2, ..., w_m\}\) corresponding to the values of the keywords named \(1, ..., m\). Now, we define a revocable attributed-based encryption scheme with keyword search that contains eight algorithms as follows:

- **Setup** \((1^d, U):\) The algorithm takes as inputs a security parameter \(1^d\) and a universe set of attributes \(U\). The attribute authority runs it to outputs the public parameters \(PP\), and the secret parameters \(\text{MSK}\).
- **KeyGen** \((\text{MSK}, A, PP):\) The algorithm takes as inputs the secret parameters \(\text{MSK}\), a set of user attributes \(A\), and the public parameters \(PP\). The data user runs it to outputs the secret key \(\text{SK}\), which involves the information of data user’s attributes.
- **Encrypt** \((PP, M, W, S):\) The algorithm takes as inputs the public parameters \(PP\), a message file \(M\), the value of keywords set \(W\) extracted from \(M\), and an access policy \(S\). The data owner runs it to outputs a ciphertext \(C_{\rho h}\), which contains the information of the access policy.
- **TokenGen** \((W', SK, PP):\) The algorithm takes as inputs the interested keywords set \(W'\) where \(w'_j\) is the value of the keyword named \(j\) for each \(w'_j \in W'\), the secret key \(SK\), and the public parameters \(PP\). The data user runs it to outputs a search token \(T_w\).
- **Search** \((T_w, I_w):\) The algorithm takes as inputs the search token \(T_w\) and the keyword index \(I_w\). If the user attribute set \(A\) satisfies the access policy \(S\) and \(W' \subseteq W\), the cloud server checks the keyword names and runs the algorithm to verify the values of the keywords.
- **Decrypt** \((CT, SK):\) The algorithm takes as inputs a ciphertext \(CT\) and the secret key \(SK\). The data user runs it to output the message \(M\).
- **PKUpd** \((x_j, AK_i):\) The algorithm takes as inputs the revoked user attribute \(x_j\), and the secret number for each non-revoked user’s attributes \(AK_i\) for \(i \in [1, ..., n]\). The attribute authority runs it to output the revocation list \(RL_{x_j}\), an updated key \(\overline{PK_i}\), and an updated secret number \(\overline{AK_i}\) for \(i \in [1, ..., n]\).
- **SKUUpd** \((SK, RL_{x_j}, \overline{AK}_i):\) The algorithm takes as inputs the revocation list \(RL_{x_j}\), the updated secret number \(\overline{AK}_i\),
for $i \in [1, ..., n]$, and the original secret key $SK$ for the non-revoked user. The attribute authority runs it to output the updated secret key $\overline{SK}$ for the non-revoked user.

Next, we give the security model for the revocable attributed-based encryption with keyword search. We define the IND-CPA security game for an ABKS scheme as follows.

**Definition II.5 (IND-CPA Security for ABKS).**

1. **Initialization:** The adversary $A$ outputs a target access policy $S^*$ to the challenger $C$.
2. **Setup:** $C$ runs Setup algorithm to produce public parameters and a master secret key, while sending the public parameters to $A$.
3. **Phase 1:** $A$ is able to issue polynomially a number of queries to $C$ for private keys by issuing $(A, id)$. If the user attribute $A$ doesn’t satisfy the access policy $S^*$, $C$ runs KeyGen algorithm to gain the private key $SK$ and sends it to $A$.
4. **Challenge:** $A$ commits two equal-length messages $M_0$ and $M_1$ with the keyword set $W^*$. Now, $C$ randomly selects $\rho \in \{0, 1\}$, and runs Encrypt algorithm with $M_\rho$ to obtain the overall ciphertext $C^* = (I^*_0, CT^*)$, where $I^*_0$ is the index of keyword set and $CT^*$ is the ciphertext component, then sends $C^*$ to $A$.
5. **Phase 2:** The queries here are similar to the ones in phase 1. $A$ continues to query with the restriction that $A$ can’t query the same access policy as $S^*$.
6. **Guess:** $A$ outputs a guess as $\rho' \in \{0, 1\}$. $A$ wins the game if $\rho' = \rho$.

In the game, we define the advantage of $A$ in winning the game as follows:

$$Adv_A^{IND-CPA} = \Pr[\rho' = \rho] - \frac{1}{2}.$$ 

If $Adv_A^{IND-CPA}$ is negligible for every polynomial-time $A$, the revocable attributed-based encryption with keyword search is said to be IND-CPA secure.

### III. Our Construction

In this section, we present a revocable attribute-based scheme with multi-keyword search. Our scheme consists of eight algorithms: Setup, KeyGen, Encrypt, TokenGen, Search, Decrypt, PKU upd, SKU upd.

#### A. The Proposed Scheme

In this subsection, we describe the details of the proposed scheme as follows.

- **Setup($1^d, U$) $\rightarrow (PP, MSK).** Taking a security parameter $1^d$, a universe set of attributes $U = \{u_1, u_2, ..., u_n\}$ as inputs, the attribute authority let $G$ and $G_T$ be the multiplicative cyclic groups of prime order $p$, and $e : G \times G \rightarrow G_T$ be a bilinear map. Then it chooses three generators $g, h, u$ from $G$ and select one collision-resistant hash function: $H : \{0, 1\}^* \rightarrow \mathbb{Z}_p$, $\alpha, \beta \in \mathbb{Z}_p$ randomly. Compute $X = g^\alpha, Y = g^\beta$ and select $AK_i \in \mathbb{Z}_p$ randomly for each attribute $u_i, i \in [1, ..., n]$. Compute $PK_i = g^{AK_i}, i \in [1, n]$ as the public attribute key. It outputs the public parameters as $PP = (g, h, U, H, X, Y, \{PK_i\}_{i \in [1, n]})$ and the master secret key as $MSK = (\alpha, \beta, \{AK_i\}_{i \in [1, n]})$. Then it publicizes $PP$ and keeps $MSK$ secret.

- **KeyGen(MSK, A, PP) $\rightarrow SK.$** Taking the master secret key $MSK$, and a set of a user’s attributes $A = \{x_1, ..., x_n\}$, where $x_i \in U$ for $i = 1$ to $n$ as inputs, the attribute authority select $r \in \mathbb{Z}_p$ randomly. It then computes

$$f_1 = g^{\alpha + \beta r + \sum_{i=1}^{n} H(x_i)AK_i},$$

$$f_2 = h^{\alpha + \beta r + \sum_{i=1}^{n} H(x_i)AK_i},$$

$$f_3 = u^{\alpha + \beta r + \sum_{i=1}^{n} H(x_i)AK_i}.$$ 

It outputs the secret key as $SK = (r, f_1, f_2, f_3)$ which is related to the attribute set $A$.

- **Encrypt(PP, M, W, S) $\rightarrow C_{ph}.$** Taking the public parameters $PP$, a message file $M$, a keywords value set $W = \{w_1, ..., w_k\}$ where $w_j$ is the value of keyword $j$, the access policy $S = \{s_1, ..., s_n\}$, and the public attribute keys $\{PK_i\}_{i \in [1, n]}$ as inputs, the data owner selects $r_1, r_2, t \in \mathbb{Z}_p$ randomly, and computes

$$C = M \cdot e(g, g)^t,$$

$$C_1 = \left( X \prod_{i=1}^n PK_i^{H(s_i)} \right)^{r_1},$$

$$C_2 = Y^{r_2},$$

$$C_3 = u^{r_3},$$

$$C_4 = g^{r_4}.$$ 

Then the data owner outputs the ciphertext corresponding to the access policy $S$ as $C_{ph} = (I_{w}, CT)$ such that $CT = (C_1, C_2)$ and $I_{w} = (I_{w_j})_{j \in [1, k]}$. $I_{w}$ is the value of keyword $j$ and $d$ is the number of the interested keywords, the user secret key $SK$, and the public parameters $PP$ as inputs, the data user selects $s \in \mathbb{Z}_p$ randomly, and computes

$$Tok_i = f_j^s,$$

$$Tok_2 = (\prod_{j=1}^d f_j^{f_j^s})^{r_2},$$

$$Tok_3 = u^{r_3},$$

$$Tok_4 = dr.$$ 

Then the data user outputs the search token as $T_w = (Tok_1, Tok_2, Tok_3, Tok_4, d)$.

- **Search(T_w, I_w, d, PP) $\rightarrow 0$ or 1.** Taking the search token $T_w$ corresponding to the data user’s attribute set $A$, the ciphertext component $I_w$ corresponding to the access policy $S = \{s_1, s_2, ..., s_n\}$, the number of the interested keywords $d$, and the public parameters $PP$ as inputs, the cloud server checks whether the following formula holds or not by comparing the access policy $S$ with the attribute set $A$ ( i.e. $s_i = x_i$, for $i = 1, ..., n$) to achieve
the purpose of fine-grained search.

\[ e(\prod_{j=1}^{h} C_{3,j} \cdot e_{4}^{T_{0} \cdot k_{1}}, T_{0} k_{1}) \]

\[ \overset{\text{def}}{=} e(T_{0} k_{2}, C_{5}) \cdot e(C_{6}^{d}, T_{0} k_{3}). \]

Output 1 if it holds. Otherwise, it outputs 0.

- **Decrypt** \((CT, SK) \rightarrow M\). Taking the ciphertext component \(CT\) and the user secret key \(SK\) as inputs, the data user computes the following formula to obtain the message file \(M\).

\[ M = \frac{C}{e(f_{1}, C_{1}, C_{2})}. \]

- **PKUpd** \((x_{i}, AK_{i}, PP) \rightarrow (RL_{x_{i}}, \overline{AK}_{i}, \overline{PK}_{i}), \) for \(i \in [1, n]\). Taking the revoked user attribute \(x_{j}\), the private parameter of the non-revoked user \(AK_{i}\), and the public parameters \(PP\) as inputs, the attribute authority adds a user's id whose attribute \(x_{j}\) has been revoked to \(RL_{x_{i}}\). For \(i = 1, n\), if \(i = j\), it selects \(\overline{AK}_{i} \in \mathbb{Z}_{p}\) randomly such that \(\overline{AK}_{i} \neq AK_{i}\); otherwise \(\overline{AK}_{i} = AK_{i}\). Then it computes \(\overline{PK}_{i} = g^{\overline{AK}_{i}}, i \in [1, n]\) and outputs the user revocation list \(RL_{x_{i}}\), the updated key \(\overline{PK}_{i, i} \in [1, n]\).

- **SKUpd** \((SK, RL_{x_{i}}, \overline{AK}_{i} \in [1, n], PP) \rightarrow \overline{SK}\). Taking the non-revoked user secret key \(SK\) whose id is not in the user revocation list \(RL_{x_{i}}\), the updated secret number \(\overline{AK}_{i}\), for \(i \in [1, n]\), and the public parameters \(PP\) as inputs, the attribute authority computes

\[ f_{1} = g^{\sum_{i=1}^{h} H_{i}(x_{i})/\overline{AK}_{i}}, \]

\[ f_{2} = h^{\sum_{i=1}^{h} H_{i}(x_{i})/\overline{AK}_{i}}, \]

\[ f_{3} = u^{\sum_{i=1}^{h} H_{i}(x_{i})/\overline{AK}_{i}}, \]

and outputs the updated non-revoked user secret key \(\overline{SK} = (f_{1}, f_{2}, f_{3})\).

**B. Correctness**

The correctness can be demonstrated as follows:

1. The correctness of keyword search:

\[ e(\prod_{j=1}^{h} C_{3,j} \cdot e_{4}^{T_{0} \cdot k_{1}}, T_{0} k_{1}) = e(\prod_{j=1}^{h} g^{h H_{i}(x_{j})} f_{1}, \prod_{i=1}^{n} P_{K_{i}}^{H_{i}(x_{i})}) \]

\[ = e(\prod_{j=1}^{h} g^{h H_{i}(x_{j})} f_{1}, g^{\prod_{i=1}^{n} P_{K_{i}}^{H_{i}(x_{i})}}) \]

\[ = e(\prod_{j=1}^{h} g^{h H_{i}(x_{j})} f_{1}, \prod_{i=1}^{n} P_{K_{i}}^{H_{i}(x_{i})} g^{b \cdot r \cdot d}) \]

\[ = e(\prod_{j=1}^{h} g^{h H_{i}(x_{j})} f_{1}, \prod_{i=1}^{n} P_{K_{i}}^{H_{i}(x_{i})} g^{b \cdot r \cdot d}) \]

\[ = e(T_{0} k_{2}, C_{5}) \cdot e(C_{6}^{d}, T_{0} k_{3}). \]

2. The correctness of decryption:

\[ C \]

\[ = M \cdot e(g, g)^{f_{1}} \]

\[ = e(g^{\sum_{i=1}^{h} H_{i}(x_{i})/\overline{AK}_{i}}, g^{\prod_{i=1}^{n} P_{K_{i}}^{H_{i}(x_{i})}})^{f_{1}} \cdot e(g^{b \cdot r \cdot d}, g) \]

\[ = M \cdot e(g, g)^{f_{1}} \]

\[ = e(g^{\sum_{i=1}^{h} H_{i}(x_{i})/\overline{AK}_{i}}, g^{\prod_{i=1}^{n} P_{K_{i}}^{H_{i}(x_{i})}})^{f_{1}} \]

\[ = M. \]

**IV. Comparisons**

In this section, we compare the properties and performance of our scheme with those of [16][18][17] in Table [III] and Table [V]. Table [V] shows that our scheme achieves fine-grained multi-keyword search, user revocation, attributes independency and constant-size secret key, simultaneously. That is, a data user is allowed to use multiple keywords to search for the data efficiently with their attribute sets conformed to the access policy.

In order to simplify the case and evaluate the performance, we have to make some assumptions. Based on [18], we have the assumptions shown in TABLE [II] and set \(|G| = |\mathbb{G}_{2}| = |\mathbb{Z}_{p}| = 256\) bits on the environment with Ubuntu 10.04 LTS OS, 2.6GHz Intel Celeron 64 bits PC, and 1 GB RAM. In addition, we make the assumption that the number of encrypted keywords is equal to the number of search keywords and set the number of attributes used in the Search algorithm as \(|I| = 10\). Also, we set \(k\) as the number of keywords related to a ciphertext and compare the cost for search/decryption as shown in Table [III] especially for the single keyword and multi-keywords when searching with the condition \(k = 1\) and \(k = 10\). Besides, we assume \(\ell = \tau = 20\) as the number of attributes related to a secret key or a ciphertext to further compute the size of ciphertext/secret key as shown in Table [IV]. Table [III] and Table [IV] show that the proposed scheme has the better performance when the number of keywords is equal to 10. Moreover, the proposed scheme owns more properties as shown in Table [IV]. Overall, the proposed scheme is more practical and suitable for real-world environments.

**V. Conclusion**

With the rapid development of cloud computing, people start to upload their data files to the cloud server for the ones who have the rights to access, which makes cloud servers play a very important role in today’s society. Because the data files are mostly confidential, the security and privacy of data become important issues. Therefore, the data owners should encrypt the data files before outsourcing them to the cloud server. Besides, encryption mechanisms have to enable data users to use keywords to search efficiently for the information they need through a large number of encrypted files. Although the searchable encryption mechanism can assist the data users on searching for encrypted files, the schemes nowadays cannot simultaneously satisfy human needs for fine-grained multi-keyword search, user revocation, and constant size secret key so as to flexibly implement in the real world.
In view of this, we propose a revocable attribute-based encryption scheme with multi-keyword search based on a traditional identity-based encryption scheme. In the proposed scheme, we achieve the advantages including multi-keyword search which is that the proposed scheme supports multi-keyword search so that the data user can effectively find required information within huge data files, fine-grained search which is that the proposed scheme supports multi-keyword search so that the data user can effectively find required information, user revocation at an attribute level to support the possible frequent change of the data user’s attributes, user attributes independency which is that the length of the ciphertext and the length of the secret key in the proposed scheme are independent of the number of user attributes and do not increase linearly, and low computation cost which is that the decryption in this scheme needs only one pairing.

To the best of our knowledge, the proposed scheme is the first that simultaneously satisfies these advantages. With those advantages, our scheme is more suitable for the clouds environment. In the future, how to prevent the revoked users from decrypting the past ciphertext and achieve a more customized multi-keyword search with a flexible access policy will be our future works.

### References


TABLE IV
Comparison of Properties

<table>
<thead>
<tr>
<th></th>
<th>Li et al. [16]</th>
<th>Sun et al. [8]</th>
<th>Wang et al. [17]</th>
<th>Our Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Keyword Search</strong></td>
<td>Single</td>
<td>Multiple</td>
<td>Single</td>
<td>Multiple</td>
</tr>
<tr>
<td><strong>Access Structure</strong></td>
<td>Tree-based</td>
<td>AND Gates</td>
<td>Linear Secret</td>
<td>AND Gates</td>
</tr>
<tr>
<td><strong>Fine-Grained Search</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>User Revocation</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Constant-Size Secret Key</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Multi-Value Independency</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Size of Secret Key</strong></td>
<td>(\approx 10240) bits</td>
<td>(\approx 10752) bits</td>
<td>(\approx 6400) bits</td>
<td>(\approx 1024) bits</td>
</tr>
<tr>
<td><strong>Size of Ciphertext</strong></td>
<td>(\approx 6656) bits</td>
<td>(\approx 5888) bits</td>
<td>(\approx 11008) bits</td>
<td>(\approx 1792) bits</td>
</tr>
</tbody>
</table>

Chun-I Fan received the M.S. degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1993, and the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1998. From 1999 to 2003, he was an Associate Researcher and a Project Leader with Telecommunication Laboratories, Chunghwa Telecom Company, Ltd., Taoyuan, Taiwan. In 2003, he joined as a faculty with the Department of Computer Science and Engineering, National Sun Yat-sen University (NSYSU), Kaohsiung, Taiwan. He has been a Full Professor since 2010 and a Distinguished Professor since 2019. He also is the Dean of College of Engineering and the Director of Information Security Research Center at NSYSU, and he was the CEO of “Aim for the Top University Plan” Office, NSYSU. And he is currently an outstanding faculty in Academic Research in NSYSU. His current research interests include applied cryptology, information security, and communication security. He received the Best Student Paper Awards from the National Conference on Information Security in 1998, the Dragon Ph.D. Thesis Award from the Institute of Information and Computing Machinery in 1999, and the Y. Z. Hsu Science Paper Award (Information and Communication Science and Technology Category) in 2020. He won the Engineering Professors Award from Chinese Institute of Engineers — Kaohsiung Chapter in 2016, and the Outstanding Technical Achievement Award from IEEE Tainan Section in 2020. He is the Chairman of Chinese Cryptology and Information Security Association, and was the Chief Executive Officer of Telecom Technology Center in Taiwan.

Si-Jing Wu is now an MS student in computer science and engineering from National Sun Yat-sen University, Kaohsiung, Taiwan. Her research interests include cloud computing and cloud storage, network and communication security, and applied cryptography.
Yi-Fan Tseng was born in Kaohsiung, Taiwan. He received the Ph.D. degree and MS degree in computer science and engineering from National Sun Yat-sen University, Taiwan, in 2014 and 2018, respectively. From 2018 to 2019, as a postdoctoral researcher, he joined the research group of Taiwan Information Security Center at National Sun Yat-sen University (TWISC@NSYSU). In 2019, he has joined the faculty of the Department of Computer Science, National Chengchi University, Taipei, Taiwan. His research interests include cloud computing and security, network and communication security, information security, cryptographic protocols, and applied cryptography.