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 on-line 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-Keyword 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.

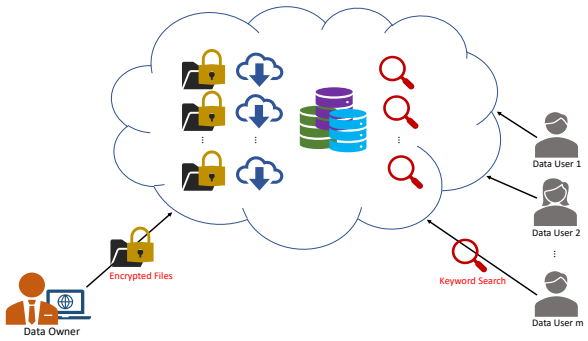

Fig. 1. The Scenario about the File Sharing in Clouds.

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 pre-cursory 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

Chun-I Fan is with the Department of Computer Science and Engineering and the Information Security Research Center, National Sun Yat-sen University, Kaohsiung 804, Taiwan, and also with the Intelligent Electronic Commerce Research Center, National Sun Yat-sen University, Kaohsiung 804, Taiwan, e-mail: cifan@mail.cse.nsysu.edu.tw.

Si-Jing Wu is with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan, e-mail: jim5566556@gmail.com.

Yi-Fan Tseng is with the Department of Computer Science, National Chengchi University, Taipei 11605, Taiwan, e-mail: yftseng@cs.nccu.edu.tw(*The corresponding author).

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 about the access policy. For the sake of protecting the classified information in the ciphertext, Li *et al.* [11] in 2017 presented an ABKS scheme with partially hidden access structures. Subsequently, Huang *et al.* [12] presented an ABKS which can be implemented in a multi-server environment and secured against the adaptive chosen keyword attack. However, the size of a ciphertext and a secret key is proportional to the number of the attributes. To improve the efficiency when searching in a large database, Wang *et al.*[13] in 2018 proposed a multi-keyword search scheme which supports attribute revocation. Nonetheless, it still exists the issue of high overhead of the searching time.

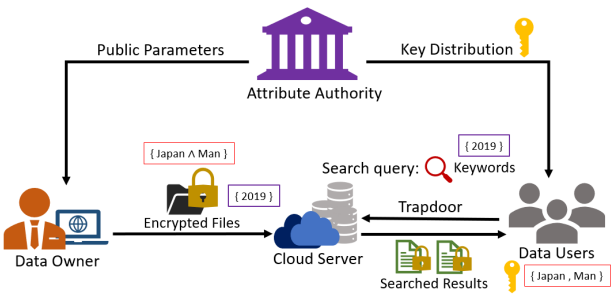

Fig. 2. The Scenario about the ABKS Scheme.

To mitigate the aforementioned security aspects, we propose a revocable attribute-based encryption with multi-keyword search scheme. The length of ciphertext and secret key in our scheme does not grow linearly with the number 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.

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.

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 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ 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 a 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 $U = (u_1, u_2, u_3)$ = ("Department", "Position", "Gender"). The possible values of u_1 ="Deparment" would be ("CS", "EE", ...), and the possible values of u_2 ="Position" would be ("Professor", "Student", ...). An access structure would be like S =("Department"="CS", "Position"="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. "Deparment" 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

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 $\mathcal{A}$ has an advantage ϵ in solving the generalized decisional Diffie-Hellman problem if

$$\left|\Pr[\mathcal{C}(g, g^a, g^b, g^c, Q = g^{abc})] - \Pr[\mathcal{C}(g, g^a, g^b, g^c, Q = R)]\right| \geq \epsilon.$$

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

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

In the game, we define the advantage of $\mathcal{A}$ in winning the game as follows:

$$Adv_{\mathcal{A}}^{\text{IND-sID-CPA}} = \left|\Pr\left[\rho' = \rho\right] - \frac{1}{2}\right|.$$

If $Adv_{\mathcal{A}}^{\text{IND-sID-CPA}}$ is negligible for every polynomial-time $\mathcal{A}$, 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**$(1^\lambda, U)$: The algorithm takes as inputs a security parameter 1^λ and a universe set of attributes U. The attribute authority runs it to outputs the public parameters PP, and the secret parameters MSK.
- **KeyGen**(MSK, A, PP): The algorithm takes as inputs the secret parameters MSK, a set of user attributes A, and the public parameters PP. The data user runs it to outputs the secret key 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_{ph}, 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 $i \in [1, ..., n]$.
- **SKUpd**$(SK, RL_{x_j}, \overline{AK_i})$: 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 $\mathcal{A}$ outputs a target access policy S^* to the challenger $\mathcal{C}$.
2) *Setup:* $\mathcal{C}$ runs **Setup** algorithm to produce public parameters and a master secret key, while sending the public parameters to $\mathcal{A}$.
3) *Phase 1:* $\mathcal{A}$ is able to issue polynomially a number of queries to $\mathcal{C}$ for private keys by issuing (A, id). If the user attribute A doesn't satisfy the access policy S^*, $\mathcal{C}$ runs **KeyGen** algorithm to gain the private key SK and sends it to $\mathcal{A}$.
4) *Challenge:* $\mathcal{A}$ commits two equal-length messages M_0 and M_1 with the keyword set W^*. Now, $\mathcal{C}$ randomly selects $\rho \in \{0, 1\}$, and runs **Encrypt** algorithm with M_ρ to obtain the overall ciphertext $C^*_{ph} = (I^*_w, CT^*)$, where I^*_w is the index of keyword set and CT^* is the ciphertext component, then sends C^*_{ph} to $\mathcal{A}$.
5) *Phase 2:* The queries here are similar to the ones in phase 1. $\mathcal{A}$ continues to query with the restriction that $\mathcal{A}$ can't query the same access policy as S^*.
6) *Guess:* $\mathcal{A}$ outputs a guess as $\rho' \in \{0, 1\}$. $\mathcal{A}$ wins the game if $\rho' = \rho$.

In the game, we define the advantage of $\mathcal{A}$ in winning the game as follows:

$$Adv^{\text{IND-CPA}}_{\mathcal{A}} = \left| \Pr\left[\rho' = \rho\right] - \frac{1}{2} \right|.$$

If $Adv^{\text{IND-CPA}}_{\mathcal{A}}$ is negligible for every polynomial-time $\mathcal{A}$, 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, PKUpd, SKUpd.*

A. The Proposed Scheme

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

- **Setup**$(1^\lambda, U) \to (PP, MSK)$. Taking a security parameter 1^λ, a universe set of attributes $U = \{u_1, u_2, ..., u_n\}$ as inputs, the attribute authority let $\mathbb{G}$ and $\mathbb{G}_T$ be the multiplicative cyclic groups of prime order p, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. Then it chooses three generators g, h, u from $\mathbb{G}$ and select one collision-resistant hash function: $H : \{0, 1\}^* \to \mathbb{Z}_p$, $\alpha, \beta \in \mathbb{Z}_p$ randomly. Compute $X = g^\alpha, Y = g^\beta$ and select $AK_i \in \mathbb{Z}_p$ randomly for each attribute $u_i, i \in [1, \ldots, n]$. 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**$(MSK, A, PP) \to SK$. Taking the master secret key MSK, and a set of a user's attributes $A = \{x_1, \ldots, x_n\}$, where $x_i \in u_i$ for $i = 1$ to n as inputs, the attribute authority select $r \in \mathbb{Z}_p$ randomly. It then computes

$$f_1 = g^{\frac{1}{\alpha + \beta r + \sum_{i=1}^{n} H(x_i) AK_i}},$$
$$f_2 = h^{\frac{1}{\alpha + \beta r + \sum_{i=1}^{n} H(x_i) AK_i}},$$
$$f_3 = u^{\frac{1}{\alpha + \beta r + \sum_{i=1}^{n} H(x_i) AK_i}}.$$

  It outputs the secret key as $SK = (r, f_1, f_2, f_3)$ which is related to the attribute set A.
- **Encrypt**$(PP, M, W, S) \to C_{ph}$. 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 $S = \{s_1, \ldots, s_n\}$, 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

$$\begin{gathered} C = M \cdot e(g, g)^t, \\ C_1 = \left(X \prod_{i=1}^{n} PK_i^{H(s_i)}\right)^t, \\ C_2 = Y^t, \\ C_{3,j} = \left(g h^{H(w_j)}\right)^{r_1} \left(X \prod_{i=1}^{n} PK_i^{H(s_i)}\right)^{r_2}, j \in [1, k], \\ C_4 = Y^{r_2}, \\ C_5 = u^{r_1}, \\ C_6 = g^{r_2}. \end{gathered}$$

  Then the data owner outputs the ciphertext corresponding to the access policy S as $C_{ph} = (I_w, CT)$ such that $CT = (C, C_1, C_2)$ and $I_w = (\{C_{3,j}\}_{j\in[1,k]}, C_4, C_5, C_6)$.
- **TokenGen**$(W', SK, PP, d) \to T_w$. Taking a set of interested keyword values $W' = \{w'_1, ...w'_d\}$, where w'_j 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

$$\begin{gathered} Tok_1 = f_3^s, \\ Tok_2 = (\prod_{j=1}^{d} f_1 f_2^{H(w'_j)})^s, \\ Tok_3 = u^s, \\ Tok_4 = dr. \end{gathered}$$

  Then the data user outputs the search token as

$$T_w = (Tok_1, Tok_2, Tok_3, Tok_4, d).$$

- **Search**$(T_w, I_w, d, PP) \to 0$ 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.

$$e(\prod_{j=1}^{k} C_{3,j} \cdot C_4^{Tok_4}, Tok_1) \stackrel{?}{=} e(Tok_2, C_5) \cdot e(C_6^d, Tok_3).$$

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.

$$M = \frac{C}{e(f_1, C_1 \cdot C_2^r)}.$$

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

$$\overline{f_1} = g^{\frac{1}{\overline{\alpha+\beta r+\sum_{i=1}^{n} H(x_i) AK_i}}},$$
$$\overline{f_2} = h^{\frac{1}{\overline{\alpha+\beta r+\sum_{i=1}^{n} H(x_i) AK_i}}},$$
$$\overline{f_3} = u^{\frac{1}{\overline{\alpha+\beta r+\sum_{i=1}^{n} H(x_i) AK_i}}}.$$

and outputs the updated non-revoked user secret key $\overline{SK} = (r, \overline{f_1}, \overline{f_2}, \overline{f_3})$.

B. Correctness

The correctness can be demonstrated as follows:

1. The correctness of keyword search:

$$\begin{aligned}
& e(\prod_{j=1}^{k} C_{3,j} \cdot C_4^{Tok_4}, Tok_1) \\
= \; & e(\prod_{j=1}^{k} \left(g h^{H(w_j)}\right)^{r_1} \left(g^{\alpha} \prod_{i=1}^{n} PK_i^{H(s_i)}\right)^{r_2} \\
& \cdot g^{\beta r_2 r d}, u^{\frac{s}{\overline{\alpha+\beta r+\sum_{i=1}^{n} H(x_i) AK_i}}}) \\
= \; & e(\prod_{j=1}^{k} \left(g h^{H(w_j)}\right)^{r_1}, u^{\frac{s}{\overline{\alpha+\beta r+\sum_{i=1}^{n} H(x_i) AK_i}}}) \\
& \cdot e((g^{\alpha} g^{\beta r} \prod_{i=1}^{n} PK_i^{H(s_i)})^{dr_2}, u^{\frac{s}{\overline{\alpha+\beta r+\sum_{i=1}^{n} H(x_i) AK_i}}}) \\
= \; & e(\prod_{j=1}^{k} \left(g h^{H(w_j)}\right)^{\frac{s}{\overline{\alpha+\beta r+\sum_{i=1}^{n} H(x_i) AK_i}}}, u^{r_1}) \cdot e(g^{dr_2}, u^s) \\
= \; & e(Tok_2, C_5) \cdot e(C_6^d, Tok_3).
\end{aligned}$$

2. The correctness of decryption:

$$\begin{aligned}
& \frac{C}{e(f_1, C_1 \cdot C_2^r)} \\
= \; & \frac{M \cdot e(g, g)^t}{e(g^{\frac{1}{\overline{\alpha+\beta r+\sum_{i=1}^{n} H(x_i) AK_i}}}, \left(g^{\alpha} \prod_{i=1}^{n} PK_i^{H(s_i)}\right)^t \cdot g^{\beta t r})} \\
= \; & \frac{M \cdot e(g, g)^t}{e(g^{\frac{1}{\overline{\alpha+\beta r+\sum_{i=1}^{n} H(x_i) AK_i}}}, \left(g^{\alpha+\beta r+\sum_{i=1}^{n} H(s_i) AK_i}\right)^t)} \\
= \; & M.
\end{aligned}$$

IV. Comparisons

In this section, we compare the properties and performance of our scheme with those of [16][8][17] in Table III and Table IV. Table IV 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 $|\mathbb{G}| = |\mathbb{G}_{\mathbb{T}}| = |\mathbb{Z}_P| = 256$ 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.

TABLE I
The Notations

Notation	Meaning
k	the number of keywords related to a ciphertext
ℓ	the number of attributes related to a secret key
τ	the number of attributes related to a ciphertext
$\|I\|$	the number of attributes used in the Search algorithm

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.

TABLE II
The Computation Costs of Cryptographic Primitives

Notation	Meaning	Cost
T_h	the cost of a hash operation	0.136 ms
T_b	the cost of a scalar multiplication in an additive group or an exponentiation in a multiplicative $\mathbb{G}$	0.348 ms
T_e	the cost of a scalar multiplication in an additive group or an exponentiation in a multiplicative $\mathbb{G}_T$	3.944 ms
T_s	the cost of a an addition in an additive group or a multiplication in a multiplicative group $\mathbb{G}$	0.55 ms
T_g	the cost of a an addition in an additive group or a multiplication in a multiplicative group $\mathbb{G}_T$	5.16 ms
T_p	the cost of a pairing operation	5.05 ms
T_m	the cost of a modular multiplication in $\mathbb{Z}_P$	0.029 ms

TABLE III
Comparison of Performance

	Search ($k = 1$)	Search ($k = 10$)	Decrypt
Li *et al.* [16]	$4T_p + 2T_e + 2\|I\|T_m + 2T_g + T_h \approx 131.424$ ms	≈ 2628.48 ms	$3T_s + 2T_p \approx 11.75$ ms
Sun *et al.* [8]	$T_b + T_s \approx 0.898$ ms	≈ 17.96 ms	NA
Wang *et al.* [17]	$T_h + T_p + T_b \approx 5.534$ ms	≈ 110.68 ms	$T_p + T_m + T_g \approx 10.239$ ms
Our Scheme	$T_p + T_b + kT_s \approx 5.948$ ms	≈ 16.398 ms	$T_p + T_s + T_g + T_b \approx 11.108$ ms

NA: no such operator in the literature

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 which is that the proposed scheme supports the 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

[1] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000*. IEEE, 2000, pp. 44–55. I

[2] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2004, pp. 506–522. I

[3] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–5. I

[4] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Transactions on parallel and distributed systems*, vol. 25, no. 1, pp. 222–233, 2014. I

[5] Q. Zheng, X. Li, and A. Azgin, "CLKS: Certificateless keyword search on encrypted data," in *International Conference on Network and System Security*. Springer, 2015, pp. 239–253. I

[6] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2005, pp. 457–473. I

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*. Acm, 2006, pp. 89–98. I

[8] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 226–234. I, IV, III, IV

[9] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: verifiable attribute-based keyword search over outsourced encrypted data," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 522–530. I

[10] 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," *Journal of medical systems*, vol. 40, no. 11, p. 246, 2016. I

[11] J. Li, Y. Shi, and Y. Zhang, "Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage," *International Journal of Communication Systems*, vol. 30, no. 1, p. e2942, 2017. I

[12] H. Haiping, D. Jianpeng, D. Hua, and W. Ruchuan, "Multi-sever multi-keyword searchable encryption scheme based on cloud storage," *Journal*

TABLE IV
Comparison of Properties

	Li *et al.* [16]	Sun *et al.* [8]	Wang *et al.* [17]	Our Scheme
Keyword Search	Single	Multiple	Single	Multiple
Access Structure	Tree-based	AND Gates	Linear Secret Sharing Scheme	AND Gates
Fine-Grained Search	Yes	Yes	No	Yes
User Revocation	No	Yes	Yes	Yes
Constant-Size Secret Key	No	No	No	Yes
Multi-Value Independency	No	No	No	Yes
Size of Secret Key	$2\ell\|\mathbb{G}\|$ $\approx$ 10240 bits	$2\|\mathbb{Z}_p\| + (2\ell+1)\|\mathbb{G}\|$ $\approx$ 10752 bits	$\|\mathbb{Z}_p\| + (4+\ell)\|\mathbb{G}\|$ $\approx$ 6400 bits	$3\|\mathbb{G}\| + \|\mathbb{Z}_p\|$ $\approx$ 1024 bits
Size of Ciphertext	$(\tau+4)\|\mathbb{G}\| + 2\|G_T\|$ $\approx$ 6656 bits	$\|\mathbb{Z}_p\| + (2+\tau)\|\mathbb{G}\|$ $\approx$ 5888 bits	$(2\tau+3)\|\mathbb{G}\|$ $\approx$ 11008 bits	$\|\mathbb{G}_T\| + 6\|\mathbb{G}\|$ $\approx$ 1792 bits

of Electronics & Information Technology, vol. 39, no. 2, pp. 389–396, 2017. I

[13] S. Wang, L. Yao, and Y. Zhang, "Attribute-based encryption scheme with multi-keyword search and supporting attribute revocation in cloud storage," *PloS one*, vol. 13, no. 10, p. e0205675, 2018. I

[14] D. Boneh and X. Boyen, "Efficient selective-ID secure identity-based encryption without random oracles," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2004, pp. 223–238. I-A, II-D

[15] E. Bresson, O. Chevassut, and D. Pointcheval, "The group diffie-hellman problems," in *International Workshop on Selected Areas in Cryptography*. Springer, 2002, pp. 325–338. II-C

[16] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 715–725, 2016. IV, III, IV

[17] S. Wang, D. Zhang, Y. Zhang, and L. Liu, "Efficiently revocable and searchable attribute-based encryption scheme for mobile cloud storage," *IEEE Access*, vol. 6, pp. 30 444–30 457, 2018. IV, III, IV

[18] A. Guillevic, "Comparing the pairing efficiency over composite-order and prime-order elliptic curves," in *International Conference on Applied Cryptography and Network Security*. Springer, 2013, pp. 357–372. IV

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 fac- ulty 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 Acer Foundation, the Best 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..