On the Security of Key-Aggregate Searchable Encryption

Jing Wen[1], Haifeng Li[2,3,*] and Caihui Lan[1]

[1]*School of Information Engineering, Lanzhou City University, Lanzhou, 730070, China*
[2]*School of Computer and Information Science, Qinghai University of Science and Technology, Xining, 810016, China*
[3]*Department of Information Technology, Qinghai University, Xining 810016, China*
E-mail: lihaifengdlut@163.com
[*]*Corresponding Author*

Received 28 April 2023; Accepted 29 January 2024;
Publication 09 April 2024

Abstract

The sharing of encrypted data in cloud computing is an essential functionality with countless applications in our everyday life. However, the issue of how to securely, efficiently and flexibly share encrypted data in multi-user settings has not been well solved. As a promising and elegant technique, the key-aggregate searchable encryption (KASE) scheme can efficiently support selective sharing of a large number of documents with a set of users using only a single, constant-size authorization key (i.e., the aggregated key). However, by conducting cryptanalysis on existing KASE schemes, we classify the attack methods into two types: offline keyword guessing attacks and authorization abuse. For the former attacks, we first employ the known keyword guessing attack methods to cryptanalyze several existing KASE schemes. Furthermore, we propose two novel keyword guessing attack methods, namely (1) Keyword guessing attack by modifying ciphertext and (2) Keyword guessing attack by constructing verification equation. For the

Journal of Cyber Security and Mobility, Vol. 13_3, 565–584.
doi: 10.13052/jcsm2245-1439.13310

latter attacks, we first utilized the known authorization abuse attack methods to cryptanalyze several existing KASE schemes. Furthermore, we develop a novel attack method in which the attacker can independently upgrade their own authorization and gain enhanced search privileges without colluding with multiple authorized users.

Keywords: Searchable encryption, key-aggregate keyword searchable encryption, offline keyword guessing attack, authorization abuse.

1 Introduction

Nowadays, with the rapid development of information technology, cloud computing [1, 9] has been widely used in different fields, providing users various benefits. Cloud storage [6] is a new storage mode based on cloud computing technology, which has become an important service of cloud computing. Since cloud servers are not completely credible, it is necessary to use encryption technology to protect data privacy. However, keyword search technology based on plaintext data is no longer suitable. Fortunately, Boneh et al. have proposed a novel technology called "public key encryption with keyword search (PEKS)" [2] to achieve searchability in a ciphertext environment. Since the pioneering work of Boneh et al., numerous researchers have conducted further investigations [3, 8, 10].

To maximize the benefits of cloud storage, data owners should be able to share their data with the intended users. Effective access control is an important requirement in addition to data security and privacy of shared data. Key-Aggregate Cryptosystem (KAC) [4, 7, 20, 23] provides a solution for a flexible access control, which was first proposed by Chu et al. [4]. In order to resolve the data searchable issue in data sharing, Cui et al. [5] proposed the Key-Aggregate Searchable Encryption (KASE) scheme on the basis of Chu et al. [4]. In their scheme, the data owner adopts the notion of KAC [4] to generate keyword ciphertext and aggregated key, i.e., the authorization key. It should be noted that different files of search authorizations can be aggregated into a single authorization key instead of generating different authorization keys for different files. A data user can generate a constant-size trapdoor through the authorization key, and send it to the cloud server. The cloud server performs keyword search according to the trapdoor, and returns search results. In this scheme, the input of the encryption algorithm is no longer the user's public key, but the index corresponding to the file and the public key of the data owner.

In 2016, Kiayias et al. [12] pointed out that Cui et al.'s scheme [5] has keyword guessing attacks and proposed a new KASE scheme. Zhou et al. [25] analyzed Cui et al.'s [5] scheme and proposed a new keyword guessing attack approach under the assumption that the attacker is an internal malicious user, that improves the speed of keyword guessing, and they also proposed a scheme to overcome the security flaw of Cui et al.'s scheme [5]. In 2019, Wang et al. [21] also analyzed Cui et al.'s scheme [5], and found that their scheme cannot withstand collusion attacks, i.e., multiple authorized users can collude with each other to obtain higher permissions, and proposed a scheme to address the issue. To ensure the verifiability of the returned search results, several verifiable key aggregation searchable encryption schemes [13–15, 17] were proposed. In 2020, Kamimura et al. [11] investigated a definition, security and application of KASE and proposed a new KASE scheme. By taking the advantages of KASE, many scholars and researchers have designed a number of KASE schemes [16, 18, 19, 22, 24]. However, under the premise that the attacker is an internal malicious user, it is found that the solutions mentioned above have some security vulnerabilities, such as offline keyword guessing attacks, uncontrollability of authorization, i.e., authorized users can generate authorizations with higher privileges independently.

While KASE can support a one-to-many search mode, it should be noted that KASE also has some limitations. One significant shortcoming is that KASE schemes typically rely on a single aggregate key to search across multiple encrypted documents. If this key is compromised, it could potentially reveal information about the searched keywords, even if the underlying document content remains secure. In some KASE schemes, it might be possible for malicious users to create forged search queries that could retrieve unauthorized documents without possessing the correct keys. This could lead to data leakage or privacy breaches. Moreover, because many individual keys are aggregated into a single aggregated key, KASE schemes often lack fine-grained access control mechanisms, providing only limited access control. Additionally, in some KASE schemes, revoking the access privileges of a specific user may require regenerating and redistributing the aggregation key, which may be cumbersome and inefficient. Furthermore, in some KASE scenarios, revoking access from a specific user may necessitate regenerating and redistributing aggregate keys, which can be cumbersome and inefficient. The encryption, decryption, and search processes in the KASE scheme may be less efficient than traditional searchable encryption because it involves using an aggregate key to decrypt multiple indexes or ciphertexts. This can result in longer search times, especially for large datasets. The KASE scheme

may also struggle to handle frequent updates or additions to the encrypted dataset efficiently, as these operations may require regenerating aggregate keys or updating search indexes, which can be resource-intensive.

1.1 Our Works

Followed Zhou et al.'s [25] work, assume that the attacker is an internal malicious user, that is, the attacker not only has some keyword/trapdoor pairs of the attacked target, but also has some authorization keys of the non-attacked target. Under this assumption, we examine existing aggregated searchable encryption schemes and find that some schemes suffer from authorization abuse and offline keyword guessing attacks, which can be summarized and categorized as follows.

- We have examined the cryptographic analysis methods of Cui et al.'s scheme [5] by Kiayias et al.'s scheme [12], Zhou et al.'s scheme [25], and Wang et al.'s scheme [21]. Additionally, we have utilized their approaches to investigate the security flaws of several other schemes. More specifically, we utilize the cryptanalysis of ciphertext by Kiayias et al. [12] to analyze the scheme in [16], and apply the cryptanalysis of trapdoors by Kiayias et al. [12] to analyze the schemes in [13,14,17–19], and employ a cryptanalysis of Zhou et al.'s scheme [25] to examine the schemes in [21, 25], and leverage the cryptanalysis of Wang et al. [21] to analyze the schemes in [4, 5, 11, 17, 18, 22, 25].
- We have proposed two novel approaches for keyword guessing attacks: one is that the attacker can launch the keyword guessing attack by modifying the ciphertext of the attacked target, the other is that the attacker can initiate the keyword guessing attack by constructing a new verification equation. Furthermore, we conducted a concrete cryptographic analysis of the relevant schemes [12–15, 17–19, 22], where the schemes in [13–15, 17, 19, 22] are found to be vulnerable to the new keyword guessing attacks by modifying ciphertext, while the schemes in [12, 18] are discovered to be vulnerable to the new keyword guessing attacks by constructing verification equation.
- We develop a novel attack method in which the attacker can independently upgrade their own authorization and gain enhanced search privileges without colluding with multiple authorized users, and present the concrete instances of cryptanalysis of the scheme [24] for the novel authorization abuse attack method.

1.2 Organization

The remainder of the paper is organized as follows. In Section 2, we briefly review some preliminary knowledge, including the definition of bilinear pairing, system definition of KASE and security requirement of KASE. In Section 3, we revisit existing KASE schemes and provide comprehensive cryptanalysis. Finally, we draw the conclusion of the whole paper in Section 4.

2 Preliminary Knowledge

2.1 Bilinear Pairing

Suppose that G and G_T are both cyclic multiplicative groups of order p is a big prime. The bilinear pairing e between two groups: $G \times G \rightarrow G_T$ is a mapping satisfying the following conditions :

- Bilinearity: $\forall a, b \in Z_p, \forall g_1, g_2 \in G, e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$;
- Non-degeneracy: $\exists g_1, g_2 \in G, e(g_1, g_2) \neq 1$, here 1 is the unit element of G_T;
- Computability : for $\forall g_1, g_2 \in G$, there is an efficient algorithm to calculate $e(g_1, g_2)$.

2.2 System Definition of KASE

A key-aggregate keyword searchable encryption framework is shown in Figure 1. There are three types of entity (Data Owner, Data Consumer and Cloud Server). The details are as follows.

Data Owner: The data owner performs encryption operation and generates search authorization for Data User.

Data User: The data users have search requirements, but it needs to obtain searchable authorization from the Data Owner to generate the search trapdoor.

Cloud Server :It stores the ciphertext and performs the specific search task after getting trapdoor, and returns the search result to authorized users.

Definition 1. A key-aggregate keyword searchable encryption scheme (KASE) consists of the following seven algorithms:

- $Setup(\lambda)$: The algorithm inputs security parameter λ and outputs system parameter pp.

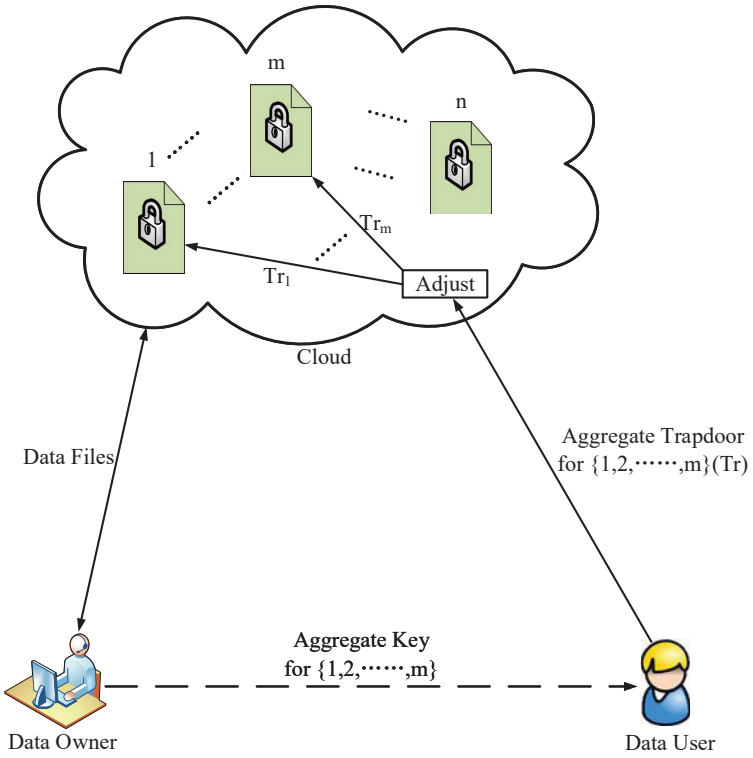

Figure 1 Key-aggregate searchable encryption.

- $KeyGen(pp)$: The algorithm is performed by the data owner to outputs a private key sk and a public key pk.
- $Authorize(pp, pk, sk, S)$: Data owner inputs private key sk and a authorized set $S \subseteq \{1, \ldots, n\}$ represents the files that can be search, and runs this algorithm to generate an authorized key AK_S for a data user (authorized user).
- $Encrypt(pp, pk, sk, F_i, w_l)$: The algorithm inputs $i \in \{1, 2, \ldots, n\}$, keyword w_l, the private key sk, and outputs the ciphertext CT. At the same time, CT is sent to the cloud server.
- $Trapdoor(pp, pk, AK_S, w_l)$: The algorithm inputs authorized key AK_S and a keyword w_l, and outputs the trapdoor Tr and sends it to the cloud server.
- $Adjust(pp, pk, i, S, Tr)$: The cloud server calculates the trapdoor Tr_i for each file according to (Tr, S).

- $Test(pp, pk, Tr_i, pp, C_i)$: The cloud server performs the search task according to the verification equation. If the equation holds, the keywords in C_i are the same as those in the trapdoor. Otherwise, the keywords in C_i are different from those in the trapdoor.

2.3 Security Requirement of KASE

In the practical application scenario of data sharing with KASE, user's data will face the security risk of data privacy leakage and illegal access. For instance, an attacker, who may be one or more internal malicious users, may launch attack through the information such as ciphertext, trapdoor, and authorization to obtain data information that does not belong to their own access rights or try to obtain rights beyond the scope of their own access rights, such as obtaining keyword information for ciphertext without search rights, obtaining authorization from other users, or generating new trapdoor or authorization.

Consequently, the basic semantic security of KASE scheme needs to satisfy the following three requirements.

1) Ciphertexts security: Keyword ciphertexts do not leak any information about relevant keywords to unauthorized attackers.
2) Trapdoors security: Trapdoors do not leak any information about relevant keywords to unauthorized attackers.
3) Authorization unforgeability: In KASE, the data owner should be the solely legitimate authorization center, any form of authorization bypassing the granting of data owner should be prohibited, that is, the attacker cannot generate a new authorization bypassing the granting of data owner.

3 Cryptanalysis for Existing KASE Schemes

In this subsection, the attacker is assumed to be an internal malicious user, which means that the attacker has obtained some authorization keys but does not know any authorization keys of the attacked target. In general, it can be assumed that the attacker's authority is less than that of the attacked target. In the KASE security model, such as [11, 17, 25], it is allowed to make authorization query and trapdoor query, that is to say, the attacker is allowed to obtain some keyword/trapdoor pairs and some authorization information according to the security model. When there is an overlap in authorization between the attacker and the target, and the attacker intercepts the target's

trapdoor, the attacker can use both the intercepted trapdoor and the trapdoor generated from their own authorization to perform searches, respectively. Then, by comparing the results returned by the two searches, the attacker can determine the keywords contained in the intercepted trapdoor. If the input of the encryption algorithm only contains the public key and keywords, but does not contain other secret information, the attacker can determine the keywords in Tr_* by generating the ciphertext by himself. Therefore, it is reasonable to assume that the attacker has some keyword / trapdoor pairs and some authorization keys.

Next, we discuss two common types of attacks: offline keyword guessing attacks and authorization abuse. Among them, keyword guessing attacks are conducted a comprehensive cryptanalysis of some state of art schemes from both the perspectives of ciphertext cryptanalysis and trapdoor cryptanalysis. An overview of the attacks is presented in Table 1, where "✓" indicates that the specified scheme (row) is insecure under a certain attack (column), and "–" indicates that the attack is not considered in this work.

Table 1 Security analysis for existing schemes

Schemes	Offline Keyword Guessing		Authorized Abuse
	Cryptanalysis of Ciphertext	Cryptanalysis of Trapdoor	
Kamimura et al. [11]	✓	–	✓
Kiayias et al. [12]	✓	–	–
Lee et al. [13]	✓	✓	–
Li et al. [14]	✓	✓	✓
Li et al. [15]	✓	–	✓
Liu et al. [16]	✓	–	–
Liu et al. [17]	✓	✓	✓
Lou et al. [18]	✓	✓	–
Oh et al. [19]	–	✓	–
Wang et al. [21]	–	✓	–
Wang et al. [22]	✓	–	✓
Yao et al. [24]	–	–	✓
Zhou et al. [25]	–	✓	✓

3.1 The Cryptanalysis of Offline Keyword Guessing Attacks

3.1.1 The existing keyword guessing attack methods

In this section, we have reviewed the cryptanalysis of Cui et al.'s scheme [5] by Kiayias et al.'s scheme [12] and Zhou et al.'s scheme [25]. Furthermore, we have leveraged their methodology to investigate the security issues of some other scheme.

A. The cryptanalysis of Kiayias et al.'s scheme [12]

A. 1 The cryptanalysis of ciphertext by Kiayias et al. [12]

Considering that Cui et al.'s method [5] employs the same random number as the key to encrypt different keywords under the same category of file , Kiayias et al. [12] found that the ciphertexts of two different keywords in the same category of file can be subjected to keyword guessing attacks by using cross pairing. The cross-pairing method proposed by Kiayias et al. [12] can also be utilized for cryptanalyzing other schemes. For instance, Liu et al. [16] put forward a KASE scheme that supports multi-keyword cascaded queries and applied it to secure data sharing in a cloud computing environment. However, their scheme has a similar flaw as Cui et al.'s scheme [5], which is using the same random number to encapsulate different keywords. Consequently, their scheme can reveal the keyword information of the targeted ciphertext under keyword guessing attack by Kiayias et al.' scheme [12]. The specific attack process is as follows:

1) The adversary intercepts ciphertexts

$$C_i = (c_1 = g^t, c_2 = e(v_2, g_i)^t, c_3 = e(g_1, g_n)^t, c_4 = g^{\frac{t}{\beta}}, c_5 = e(g, g)^{rt},$$
$$CW_i = \{c_{i,j}\}_{j=1}^{m} = \{pk_s^{H(w_{i,j})+\beta r}\}_{j=1}^{m},$$

where the pk_s is the public key of the cloud server.

2) The adversary selects two keywords w_1^* and w_2^* as the guessed values of ciphertext $c_{i,k}$and $c_{i,l}$ $(1 \leq k \neq l \leq m)$ respectively, and then verifies whether the equation $\frac{c_{i,k}}{pk_s^{H(w_1^*)}} = \frac{c_{i,l}}{pk_s^{H(w_2^*)}}$ is true.

3) If the above equation holds, it indicates that the ciphertexts $c_{i,k}$ and $c_{i,l}$ contain the keywords w_1^* and w_2^* respectively; Otherwise, repeat step 2).

A.2 The cryptanalysis of trapdoors by Kiayias et al. [12]

Kiayias et al. [12] also conducted a trapdoor analysis on Cui et al.'s scheme [5], which is similar to the ciphertext analysis method above. More specifically, they assumed that the attacker (i.e., the server) intercepted two different search trapdoors $Tr_1 = k_{agg}H(w_1)$ and $Tr_2 = k_{agg}H(w_2)$ from the same

user. Similarly, w_1^* and w_2^* can be used as the guessed values for Tr_1 and Tr_2, respectively, and then verify whether the equation $\frac{Tr_1}{H(w_1^*)} = \frac{Tr_2}{H(w_2^*)}$ holds. If the equation is valid, it means that the guess is correct, and then the authorization information of the data user can be further recovered by $\frac{Tr_1}{H(w_1^*)}$ or $\frac{Tr_2}{H(w_2^*)}$.

Since the trapdoor construction in scheme [13,14,17–19] is similar to Cui et al.'s scheme [5], authorization can be recovered using the above approach proposed by Kiayias et al. [12].

B. The cryptanalysis proposed by Zhou et al. [25]

Kiayias et al. [12] pointed out that Cui et al. [5] scheme could recover the user's authorization under keyword guessing attacks. Later, Zhou et al. [25] assumed that the attacker is an insider attacker (authorized user) and took advantage of the fact that there is an intersection between their own authorization and the attacked trapdoor to provide a more effective method for keyword guessing attacks and authorization recovery. Unfortunately, applying the cryptanalysis method proposed by Zhou et al. to analyze both their own proposed Fc-MKA-KSE scheme and the scheme in [21], it can be found that there exist similar security issues.

Example 1. Cryptanalysis of Zhou et al.'s scheme [25]

Let $\mathscr{B}$ be an authorized user who has been compromised by an attack, and let S_B be his authorized file set and k_{agg^*} be his private key. Let $\mathscr{A}_1$ be an insider attacker, and let S_A be his authorization set. We also assume $S_A \neq S_B, S_A \cap S_B \neq \emptyset$, and $S_A \subseteq S_B$. To simplify the notation, we denote the common files as $F_l = S_A \cap S_B$, and we assume that F_l contains the keywords w_1, w_2, w_3.

Similarly, we assume the authorized user $\mathscr{B}$ sends the searchable trapdoor $Tr_* = (Tr_{*1} = k_{agg^*}^{H(w_*)} v^x,\ Tr_{*2} = u^x)$ to the cloud server and receives the search results that contain F_l, and we suppose that this process is intercepted by the attacker $\mathscr{A}_1$. Here, $u = g^{\beta_1}$ is the public key of the server, and $v = g^{\beta_2}$ is the public key of the data owner, and x is a random number. Obviously, there is $w_* \in \{w_1, w_2, w_3\}$.

The attacker $\mathscr{A}_1$ can calculate a new trapdoor of other keyword w' as follows according to Tr_*.

1) Compute the following three groups of trapdoor items.

$$\begin{aligned} Tr_{*,w_1} &= \left((Tr_{*1})^{\frac{H(w_3)}{H(w_1)}}, (Tr_{*2})^{\frac{H(w_3)}{H(w_1)}}\right) \\ &= \left(k_{agg^*}^{\frac{H(w_*)H(w_3)}{H(w_1)}} v^{\frac{H(w_3)}{H(w_1)}x}, u^{\frac{H(w_3)}{H(w_1)}x}\right) \end{aligned}$$

$$Tr_{*,w_2} = ((Tr_{*1})^{\frac{H(w_1)}{H(w_2)}}, (Tr_{*2})^{\frac{H(w_1)}{H(w_2)}})$$
$$= (k_{agg^*}^{\frac{H(w_*)H(w_1)}{H(w_2)}} v^{\frac{H(w_1)}{H(w_2)}x}, u^{\frac{H(w_1)}{H(w_2)}x})$$
$$Tr_{*,w_3} = ((Tr_{*1})^{\frac{H(w_2)}{H(w_3)}}, (Tr_{*2})^{\frac{H(w_2)}{H(w_3)}})$$
$$= (k_{agg^*}^{\frac{H(w_*)H(w_2)}{H(w_3)}} v^{\frac{H(w_2)}{H(w_3)}x}, u^{\frac{H(w_2)}{H(w_3)}x})$$

Obviously, one of the three trapdoor items above is a valid trapdoor.

2) Initiate the search request to the cloud server using these three groups of trapdoor items, respectively. If the search results of Tr_{*,w_k} contain F_l, it means that the intercepted Tr_* belongs to the keyword w_k. Here, $k \in 1, 2, 3$.
3) The attacker $\mathscr{A}_1$ can generate searchable trapdoor of other keyword w' by

$$Tr_{w'} = ((K_{agg^*}^{H(w_*)} v^x)^{\frac{H(w')}{H(w_k)}}, (u^x)^{\frac{H(w')}{H(w_k)}})$$
$$= (k_{agg^*}^{H(w')} v^{\frac{H(w')}{H(w_k)}x}, u^{\frac{H(w')}{H(w_k)}x}).$$

Further, the attacker can use the new trapdoor $Tr_{w'}$ to perform ciphertext search to obtain keyword.

Based on the above cryptanalysis, it can be concluded that both Zhou et al.'s scheme [25] and Cui et al.'s scheme [5] suffer from the same attack by internal malicious users.

Example 2. Cryptanalysis of Wang et al.'s scheme [21]
Assume that the attacker has obtained two trapdoors of the attacked target, and obtains the keywords corresponding to them through the above attack method. Without loss of generality, we may assume that the attacker has two keyword/trapdoor pairs $(w_1, T_{w_1 1} = D_1 D_3^{H_1(w_1)}, T_{w_1 2} = D_2 D_4^{H_1(w_1)})$, $(w_2, T_{w_2 1} = D_1 D_3^{H_1(w_2)}, T_{w_2 2} = D_2 D_4^{H_1(w_2)})$ for the same attacked target $\mathscr{B}$. Then, the attacker $\mathscr{A}_1$ can derive the aggregate key D_1, D_2, D_3, D_4 of $\mathscr{B}$ by the following steps.

1) Compute $D_3 = \frac{T_{w_1 1}}{T_{w_2 1}}^{(H_1(w_1)-H_1(w_2))^{-1}}$;
2) Compute $D_4 = \frac{T_{w_1 2}}{T_{w_2 2}}^{(H_1(w_1)-H_1(w_2))^{-1}}$;
3) Compute $D_1 = T_{w_1 1} D_3^{-H_1(w_1)}$,$D_2 = T_{w_1 2} D_4^{-H_1(w_1)}$.

The attacker $\mathscr{A}_1$ can generate search trapdoor using the aggregate key D_1, D_2, D_3, D_4, further the attacker can obtain the keyword of ciphertext by searching.

3.1.2 The novel keyword guessing attack methods

In this section, we further investigate the security of the state-of-the-art KASE schemes and put forward three new attack methods: (1) Keyword guessing attack by modifying ciphertext; (2) Keyword guessing attack by constructing verification equation.

A. To launch keyword guessing attack by modifying the intercepted ciphertext

Assuming that the attacker $\mathscr{A}_1$ obtains any keyword w_* and the corresponding trapdoor Tr_*, the attacker can first select a keyword w_g from all keyword set as the guessed value of the keyword w_l that is contained in the attacked ciphertext C; then, the attacker modifies the ciphertext C to another ciphertext C' and uses the trapdoor Tr_* to perform adjustment and search algorithms to verify his guesses. The following is illustrated by cryptanalysis of Kamimura et al.'s scheme [11].

1) The attacker $\mathscr{A}_1$ intercepts a ciphertext $C = (c_{1,i,l}, c_{2,i,l}, c_{3,i,l})$ that $\mathscr{B}$ can access, where $c_{1,i,l} = g^{t_{i,l}}$,$c_{2,i,l} = \left(g^{\beta} g_i\right)^{t_{i,l}}$, $c_{3,i,l} = \frac{e(H(w_l),g)^{t_{i,l}}}{e(g_1,g_n)^{t_{i,l}}}$ and $t_{i,l}$ is a random number chosen for encryption.
2) The attacker $\mathscr{A}_1$ first guesses w_g to be the contained keyword of C, then modifies C to $C^{'} = (c_{1,i,l}, c_{2,i,l}, c^{'}_{3,i,l} = e(H(w_*)H(w_g)^{-1}, c_{1,i,l})c_{3,i,l})$. Obviously, when $w_g = w_l$, $C^{'}$ is a valid ciphertext of w_*.
3) Input trapdoor Tr_*, the attacker $\mathscr{A}_1$ executes the $Adjust$ algorithm to get the values Tr_i and pub.
4) If $\frac{e(Tr_i,c_{1,i,l})}{e(c_{2,i,l},pub)} = c'_{3,i,l}$ holds, the attacker can determine that the current guessed value w_g contains the ciphertext C.

The above attack assumes that the attacker knows a trapdoor and its corresponding keyword. However, in practice, when the ciphertext and the trapdoor's keyword are both unknown, the attacker can use a cross-pairing method to guess both the ciphertext and the trapdoor's keyword simultaneously. Suppose the attacker intercepts a ciphertext containing the keyword w_1^* and a trapdoor containing the keyword w_2^*, he first chooses a keyword w_1 from the keyword space as the guessed value for w_1^*, and then chooses another keyword w_2 from the keyword space as the guessed value for w_2^*. By modifying the ciphertext in the above way, and then he uses the search

trapdoor for validation. Obviously, the attacker can let w_1 and w_2 traverse the keyword space, and there will always be a case where the search verification equation $w_1 = w_1^*$ and $w_2 = w_2^*$ holds. When the verification equation is valid, it means that the guess is successful.

Similarly, the type of attack also exists in the literature [13–15, 17, 22].

B. To initiate Keyword guessing attack by constructing verification equation.

In the literature [12, 18], there is another equation that can be used to guess any ciphertext's keyword with only the attacker $\mathscr{A}_1$'s authorization. Thus, the keyword guessing attacks can be performed by any user with the attacker's authorization.

Example 1. Cryptanalysis of Lou et al.'s scheme [18]

1) The attacker (an authorized user) can obtain the authorization information as below. $d_1 = v^{t_1 t_2}$, $d_2 = g^{t_1 t_2}$, $d_1^{'} = u^{-t_2}$, $d_2^{'} = u^{-t_1}$, $d_1^{''} = h^{-t_2}$, $d_2^{''} = h^{-t_1}$, $d_3 = \Pi_{j \in S} g_{n+1-j}^{\gamma}$, where t_1, t_2, γ are the private key of data owner, and $\{g_i\}_{i \in [1,n] \cup [n+2,2n]}$ is the public parameters.
2) Assume that the ciphertext $CT_w = (C_1, C_2, C_3, C_4, C_5,$ $C_6) = (e(g_1, g_n)^s, g^s, (vg_i)^s, v^{-s}(u^w h)^{z_1}, T_1^{z_1 - s_1}, T_2^{s_1})$ is the attacked target, where r, s_1, z_1 are random numbers, w is a keyword, and $T_1 = g^{t_1}$,$T_2 = g^{t_2}$,$v = g^{\gamma}$ are the public key.
3) Construct the following equation

$$e(C_4, d_2) = e(d_1, C_2^{-1}) e((d_1^{'})^{-w}, C_5) e((d_2^{'})^{-w}, C_6) \\ e((d_1^{''})^{-1}, C_5) e((d_2^{''})^{-1}, C_6)).$$

It is obvious that the above equation only involves $d_1, d_2, d_1^{'}, d_2^{'}, d_1^{''}, d_2^{''}$ and C_2, C_4, C_5, C_6, but not contains d_3 and C_3 related to search permission control. In addition, any authorized user possesses a copy of $d_1, d_2, d_1^{'}, d_2^{'}, d_1^{''}, d_2^{''}$. Hence, the attacker (an authorized user) can perform a successful attack by finding a keyword that satisfies the above equation.

The correctness of the above equation can be guaranteed as follows.

$$(1)\ e(C_4, d_2) = e(v^{t_1 t_2}, C_2^{-1}) e(u^w, g^{t_1 t_2 z_1}) e(h, g^{t_1 t_2 z_1})$$
$$(2)\ e(v^{t_1 t_2}, C_2^{-1}) = e(d_1, C_2^{-1})$$
$$(3)\ e(u^w, g^{t_1 t_2 z_1}) = e(u^{t_2 w}, g^{t_1 z_1 - t_1 s_1}) e(u^{t_1 w}, g^{t_2 S_1}) \\ = e((d_1^{'})^{-w}, C_5) e((d_2^{'})^{-w}, C_6)$$

$$(4)\ \ e(h, g^{t_1 t_2 z_1}) = e(h^{t_2}, g^{t_1 z_1 - t_1 s_1}) e(h^{t_1}, g^{t_2 s_1})$$
$$= e((d_1^{''})^{-1}, C_5) e((d_2^{''})^{-1}, C_6)$$

Example 2. Cryptanalysis of Kiayias et al.'s scheme [12]
There is also the same issue as in Lou et al.'s [18], that is, their scheme also has another equation that can guess the keywords of ciphertext. Let $sk_{Do} = (\gamma, \beta, a_1, b_1, a_2, b_2)$ to represent the private key of the data owner, and $pk = (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, v = g^{\gamma}, v^{'} = g^{\beta}, h_{0,1}^{a_1}, h_{1,1}^{a_1}, h_{0,1}^{b_1}, h_{1,1}^{b_1}, h_{0,2}^{a_2}, h_{1,2}^{a_2}, h_{0,2}^{b_2}, h_{1,2}^{b_2})$ be the public key, where $g, h_{0,1}, h_{0,2}, h_{1,1}, h_{1,2} \in G$. The description is as follows.

1) The attacker (user i) can obtain the secret key as below.
   $d_{i,1} = g_i^{\gamma}, d_{i,2} = g^{a_1 \rho_{i1}}, d_{i,3} = g^{a_1 \rho_{i1}^{'}}$
   $d_{i,4} = g^{a_2 \rho_{i2}}, d_{i,5} = g^{a_2 \rho_{i2}^{'}}, d_{i,6} = g^{b_1 \rho_{i1}}$
   $d_{i,7} = g^{b_1 \rho_{i1}^{'}}, d_{i,8} = g^{b_2 \rho_{i2}}, d_{i,9} = g^{b_2 \rho_{i2}^{'}}$
   $d_{i,10} = h_{0,1}^{a_1 b_1 \rho_{i1}} h_{0,2}^{a_2 b_2 \rho_{i2}}, d_{i,11} = h_{0,1}^{a_1 b_1 \rho_{i1}^{'}} h_{0,2}^{a_2 b_2 \rho_{i2}^{'}}$
   $d_{i,12} = h_{1,1}^{a_1 b_1 \rho_{i1}} h_{1,2}^{a_2 b_2 \rho_{i2}}, d_{i,13} = h_{1,1}^{a_1 b_1 \rho_{i1}^{'}} h_{1,2}^{a_2 b_2 \rho_{i2}^{'}}$
   $d_{i,14} = g_i^{\beta}$
   where $\rho_{i1}, \rho_{i2}, \rho_{i1}^{'}, \rho_{i2}^{'}$ are the random number.
2) Assuage that the ciphertext $CT_k = (hdr_{k,1}, \ldots, hdr_{k,10})$ is the attacked target,and the ciphertext CT_k allows users included in S_k $(i \notin S_k)$ to search and access.
   $hdr_{k,1} = (h_{0,1}^{a_1}(h_{1,1}^{a_1})^w)^{t_1}, hdr_{k,2} = (h_{0,1}^{b_1}(h_{1,1}^{b_1})^w)^{t-t_1},$
   $hdr_{k,3} = (h_{0,2}^{a_2}(h_{1,2}^{a_2})^w)^{t_2}, hdr_{k,4} = (h_{0,2}^{b_2}(h_{1,2}^{b_2})^w)^{t-t_2},$
   $hdr_{k,5} = g^t, hdr_{k,6} = (v \Pi_{j \in S_k} g_{n+1-j})^t, hdr_{k,7} = K,$
   $hdr_{k,8} = g^{t^{'}}, hdr_{k,9} = (v^{'} \Pi_{j \in S_k} g_{n+1-j})^{t^{'}},$
   $hdr_{k,10} = K^{'} M_k.$
   where $t, t^{'}, t_1, t_2$ are random numbers, w is a keyword, and $K = e(g_{n+1}, g)^t$, $K^{'} = e(g_{n+1}, g)^{t^{'}}$.
3) Construct the following equation
   $e(d_{i,6}, hdr_{k,1}) e(d_{i,2}, hdr_{k,2}) e(d_{i,8}, hdr_{k,3}) e(d_{i,4}, hdr_{k,4})$
   $= e(d_{i,10}, hdr_{k,5}) e(d_{i,12}, hdr_{k,5})^w$

Obviously, the above equation only includes $d_{i,2}, d_{i,4}, d_{i,6}, d_{i,8}, d_{i,12}$ and $hdr_{k,1}, hdr_{k,2}, hdr_{k,3}, hdr_{k,4}, hdr_{k,5}$, but not contains $d_{i,14}$ and $hdr_{k,6}$, $hdr_{k,9}$ related to search permission control. In addition, any authorized user

has a copy of $d_{i,2}, d_{i,4}, d_{i,6}, d_{i,8}, d_{i,12}$. Therefore, the attacker (an authorized user) can be achieved by guessing the keyword that makes the above equation hold.

The correctness of the above equation can be guaranteed by the following equations.

$$
\begin{aligned}
(1)\ & e(d_{i,6}, hdr_{k,1})e(d_{i,2}, hdr_{k,2}) \\
&= e(g^{b_1\rho_{i1}}, (h_{0,1}^{a_1}(h_{1,1}^{a_1})^w)^{t_1})e(g^{a_1\rho_{i1}}, (h_{0,1}^{b_1}(h_{1,1}^{b_1})^w)^{t-t_1}) \\
&= e(g^{a_1\rho_{i1}}, (h_{0,1}^{b_1}(h_{1,1}^{b_1})^w)^{t}) \\
(2)\ & e(d_{i,8}, hdr_{k,3})e(d_{i,4}, hdr_{k,4}) \\
&= e(g^{b_2\rho_{i2}}, (h_{0,2}^{a_2}(h_{1,2}^{a_2})^w)^{t_2})e(g^{a_2\rho_{i2}}, (h_{0,2}^{b_2}(h_{1,2}^{b_2})^w)^{t-t_2}) \\
&= e(g^{a_2\rho_{i2}}, (h_{0,2}^{b_2}(h_{1,2}^{b_2})^w)^{t}) \\
(3)\ & e(d_{i,10}, hdr_{k,5})e(d_{i,12}, hdr_{k,5})^w \\
&= e(h_{0,1}^{a_1b_1\rho_{i1}}h_{0,2}^{a_2b_2\rho_{i2}}, g^t)e(h_{1,1}^{a_1b_1\rho_{i1}}h_{1,2}^{a_2b_2\rho_{i2}}, g^t) \\
&= e(h_{0,1}^{a_1b_1\rho_{i1}}, g^t)e(h_{0,2}^{a_2b_2\rho_{i2}}, g^t)e(h_{1,1}^{a_1b_1\rho_{i1}w}, g^t)e(h_{1,2}^{a_2b_2\rho_{i2}w}, g^t) \\
&= e(h_{0,1}^{a_1b_1\rho_{i1}}h_{1,1}^{a_1b_1\rho_{i1}w}, g^t)e(h_{0,2}^{a_2b_2\rho_{i2}}h_{1,2}^{a_2b_2\rho_{i2}w}, g^t) \\
&= e(g^{a_1\rho_{i1}}, (h_{0,1}^{b_1}(h_{1,1}^{b_1})^w)^t)e(g^{a_2\rho_{i2}}, (h_{0,2}^{b_2}(h_{1,2}^{b_2})^w)^t)
\end{aligned}
$$

3.2 The Cryptanalysis of Authorization Abuse

3.2.1 The existing authorization abuse attack methods

Wang et al. [21] also conducted a security analysis of the authorization mechanism in Cui et al.'s scheme [5]. They found that their scheme is vulnerable to authorization abuse. That is, the data users can collude with each other to obtain higher authorizations without involving the data owner.

Assuming that $\mathscr{A}_1$ is an attacker and has associated with multiple authorized users. For a better description, it is assumed that there are n compromised authorized users, and the corresponding authorized subsets are $\{1\}, \{2\}, \ldots, \{n\}$ respectively. The following is illustrated by the literature [17].

1) Firstly, the attacker $\mathscr{A}_1$ collects the aggregate keys $k_{agg_1} = {g_n}^{\gamma}, k_{agg_2} = g_{n-1}^{\gamma}, \ldots, k_{agg_n} = g_1^{\gamma}$. Here γ is private key of data owner, and $\{g_i\}$ $(i = 1, 2, \ldots, n)$ are the system parameters.
2) Then, the attacker can generate authorization keys $k_{agg}^{'} = \Pi_{j\in S'} g_{n+1-j}^{\gamma}$ $= \Pi_{j\in S'} k_{agg_j}$ for new users of any subset $S' \subseteq \{1, 2, \ldots, n\}$.

Since the authorization generation in schemes [11, 14, 15, 17, 22, 25] is similar to Cui et al.'s scheme [5], their authorization can be recovered using the aforementioned approach proposed by Wang et al. [21], that is, that is, by colluding to generate authorization with higher privileges.

3.2.2 The novel authorization abuse attack methods

In the reference [24], the attacker (an authorized user) can derive another aggregate key via the aggregate key extracted by the data owner. For the sake of simplicity, suppose the attacker has obtained the aggregate key of $S = \{1\}$, then, the aggregate key of $S' = \{1, 2\}$ can be obtained as follows.

1) The attacker obtains K_S by the algorithm $Extract$, this is $K_S = BasisDel(A, R_S, T_A, s_S)$, where A is the public key and T_A is the private key of the data owner, and satisfies $AT_A mod q = 0$,$R_S = G(H(1))$. Let $F_S = AG(H(1))^{-1}$, then $F_S K_S mod q = 0$ holds.
2) The attacker computes $R = G(H(2))$, and obtains the aggregate key $K_{S'} = BasisDel(F_S, R, K_S, s_{S'})$. Obviously, $F_S R^{-1} K_{S'} mod\ q = 0$ holds, that is, $A(G(H(2))G(H(1)))^{-1} K_{S'} mod\ q = 0$ holds. Thus, $K_{S'}$ can also be regarded as obtaining the aggregate key through $Extract$ in the original scheme.

Obviously, the authorization scope of $K_{S'}$ is lager than that of K_S.

4 Conclusion

In this paper, we have cryptanalyzed the existing KASE schemes and shown that they have two types of security vulnerabilities, i.e., offline keyword guessing attacks and authorization abuse. For the former attacks, we first employ the known keyword guessing attack methods to cryptanalyze several existing KASE schemes. Furthermore, we propose two novel keyword guessing attack methods, namely (1) Keyword guessing attack by modifying ciphertext and (2) Keyword guessing attack by constructing verification equation. Then, we provide the concrete instances of cryptanalysis for the novel two new keywords guessing attack methods. For the latter attacks, we first utilized the known authorization abuse attack methods to cryptanalyze several existing KASE schemes. Furthermore, we develop a novel attack method in which the attacker can independently upgrade their own authorization and gain enhanced search privileges without colluding with multiple authorized users. Then, we also present the concrete instances of cryptanalysis for the novel authorization abuse attack method. We hope that our analysis can help cryptography researchers to design more secure KASE schemes in the future.

Acknowledgments

The authors are very grateful to the anonymous referees for their valuable comments and constructive suggestions to improve the quality of our paper.

This work is supported in part by "Kunlun Elite" Talent Recruitment Research Project under Grant No. 2023-QLGKLYCZX-028, and New Faculty (Ph.D.) Extended Research and Cultivation Program under Grant No.202302lwys018.

References

[1] Y Bao, W Qiu, and X Cheng. Secure and lightweight fine-grained searchable data sharing for iot-oriented and cloud-assisted smart healthcare system. *IEEE Internet of Things Journal*, 9(4):2513–2526, 2022.

[2] D Boneh, G Crescenzo, R Ostrovsky, and G Persiano. Public key encryption with keyword search. international cconf. on the theory and applications of cryptographic techniques, interlaken, switzerland, 2–6 May, 2004.

[3] P. Chaudhari and M. L. Das. Keysea: Keyword-based search with receiver anonymity in attribute-based searchable encryption. *IEEE Transactions on Services Computing*, 15(2):1036–1044, 2022.

[4] Cheng-Kang Chu, Sherman SM Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H Deng. Key-aggregate cryptosystem for scalable data sharing in cloud storage. *IEEE transactions on parallel and distributed systems*, 25(2):468–477, 2013.

[5] B Cui, Z Liu, and L Wang. Key-aggregate searchable encryption (kase) for group data sharing via cloud storage. *IEEE Transactions on computers*, 65(8):2374–2385, 2015.

[6] K. Dhal, S. C. Rai, P. K. Pattnaik, and S. Tripathy. Cemar: a fine grained access control with revocation mechanism for centralized multi-authority cloud storage. *Journal of supercomputing*, 78(1):987–1009, 2022.

[7] C Guo, N Luo, M.Z.A Bhuiyan, and et al. Key-aggregate authentication cryptosystem for data sharing in dynamic cloud storage. *Future Generation Computer Systems*, 2017.

[8] Q Huang and H Li. An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks. *Information Sciences*, s 403–404:1–14, 2017.

[9] W. Hussain, J. M. Merigo, H. Gao, and et al. Integrated ahp-iowa, powa framework for ideal cloud provider selection and optimum resource management. *IEEE Transactions on Services Computing*, 2022.

[10] Ik Rae Jeong, Jeong Ok Kwon, Dowon Hong, and Dong Hoon Lee. Constructing peks schemes secure against keyword guessing attacks is possible? *Computer communications*, 32(2):394–396, 2009.

[11] M. Kamimura, N. Yanai, S Okamura, and et al. Key-aggregate searchable encryption, revisited: Formal foundations for cloud applications, and their implementation. *IEEE Access*, pages 1–17, 2020.

[12] Aggelos Kiayias, Ozgur Oksuz, Alexander Russell, Qiang Tang, and Bing Wang. Efficient encrypted keyword search for multi-user data sharing. In *European symposium on research in computer security*, pages 173–195. Springer, 2016.

[13] JoonYoung Lee, MyeongHyun Kim, JiHyeon Oh, YoungHo Park, KiSung Park, and Sungkee Noh. A secure key aggregate searchable encryption with multi delegation in cloud data sharing service. *Applied Sciences*, 11(19), 2021.

[14] T Li, Z Liu, C Jia, and et al. key-aggregate searchable encryption under multi-owner setting for group data sharing in the cloud. *Int. J. Web and Grid Services*, 14(1):21–43, 2018.

[15] Tong Li, Zheli Liu, Ping Li, Chunfu Jia, Zoe L Jiang, and Jin Li. Verifiable searchable encryption with aggregate keys for data sharing in outsourcing storage. In *Australasian inproceedings on Information Security and Privacy*, pages 153–169. Springer, 2016.

[16] Jinlu Liu, Bo Zhao, Jing Qin, Xinyi Hou, and Jixin Ma. Key-aggregate searchable encryption supporting conjunctive queries for flexible data sharing in the cloud. *Information Sciences*, 645:119336, 2023.

[17] Zheli Liu, Tong Li, Ping Li, Chunfu Jia, and Jin Li. Verifiable searchable encryption with aggregate keys for data sharing system. *Future Generation Computer Systems*, 78:778–788, 2018.

[18] C Lou, M Cao, Y Lou, and et al. A secure key-aggregate keyword retrieval scheme over encrypted data in cloud computing. *IEEE Access*, pages 1–12, 2020.

[19] Jihyeon Oh, JoonYoung Lee, MyeongHyun Kim, Youngho Park, KiSung Park, and SungKee Noh. A secure data sharing based on key aggregate searchable encryption in fog-enabled iot environment. *IEEE Transactions on Network Science and Engineering*, 9(6):4468–4481, 2022.

[20] S Patranabis, Y Shrivastava, and D Mukhopadhyay. Provably secure key-aggregate cryptosystems with broadcast aggregate keys for online data sharing on the cloud. *IEEE Transactions on Computers*, 66(5): 891–904, 2017.

[21] H Wang, X Dong, Z Cao, and et al. Secure key-aggregation authorized searchable encryption. *Science China, Information Sciences*, 62: 039111:–039111:3, 2019.

[22] Xuqi Wang, Yu Xie, Xiangguo Cheng, and Zhengtao Jiang. An efficient key-aggregate keyword searchable encryption for data sharing in cloud storage. In *IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2019.

[23] Z Wang. Provably secure key-aggregate cryptosystems with auxiliary inputs for data sharing on the cloud. *Future generation computer systems*, 939(APR):770–776, 2019.

[24] Y Yao, Z Zhai, J Liu, and et al. lattice-based key-aggregate (searchable) encryption in cloud storage. *IEEE Access*, 2019.

[25] Rang Zhou, Xiaosong Zhang, Xiaojiang Du, Xiaofen Wang, Guowu Yang, and Mohsen Guizani. File-centric multi-key aggregate keyword searchable encryption for industrial internet of things. *IEEE Transactions on Industrial Informatics*, 14(8):3648–3658, 2018.

Biographies

Jing Wen received the M.S. degree in computer science and technology from the Chang'an University in 2009. She is currently a Lecturer with the School of Electronics and Information Engineering, Lanzhou City University. Her research interests include network security and cryptography.

Haifeng Li received the B.S. degree in computer science from Hebei University and the M.S. degree in computer science from Northwest Normal University and the Ph.D. degree with the School of Software, Dalian University of Technology. Currently, he is an associate with the School of Computer and Information Science, Qinghai University of Science and Technology. His research interests include applied cryptography, network security, cloud computing security, and big data security.

Caihui Lan received the Ph.D. degree in basic mathematics from the School of Mathematics and Statistics, Northwest Normal University, Lanzhou, China, in 2013. He is currently an Associate Professor with the School of Electronics and Information Engineering, Lanzhou City University. His main research interests include cryptography and information security, in particular, cryptographic protocols.