Multi-user Fuzzy Keyword Searchable Encryption Scheme Based on Certificateless Cryptosystem

Xiaodong Yang*, Jiaqi Wang, Tian Tian, and Xiuxiu Wang

Department of Computer Science and Engineering, University of Northwest Normal University, Teaching Building No. 9, Room 9C102, 967 Anning East Road, Anning District, Lanzhou, China

Email: y200888@163.com; hhwjdyxhh@163.com

Abstract. Nowadays, the application prospect of cloud storage is more and more extensive, and the security problem of cloud storage has become one of the hotspots of research. Searchable encryption can make full use of the huge computing resources of cloud servers and solve the problem of secure search after data encryption and upload. However, the existing searchable encryption schemes have the problems of high computational cost, low security, and insufficient flexibility in the way of searching keywords. In order to solve these problems, we propose a new certificateless multi-user fuzzy keyword searchable encryption scheme, which introduces the key distribution center, effectively reduces the computational cost, solves the problem of certificate management and key escrow, and uses the optimized wildcard technology to realize the fuzzy search of multiple users, which is not only brings convenience to users, but also realizes flexible search methods. Finally, the analysis results show that our scheme has indistinguishable security under keyword guessing attacks. Simulation experiments and performance analysis results show that our scheme has higher computational efficiency.

1. Introduction

Uploading a large amount of data to cloud servers through outsourcing has become the choice of more and more people or companies, but because cloud data is out of people’s control, the ensuing data leakage problem has aroused everyone’s attention and discussion [1]. The traditional search based on plaintext data is no longer suitable for cloud storage. Therefore, the security of cloud data has become a problem to be solved [2]. It is necessary to design a search scheme suitable for cloud storage to deal with this problem.

Song et al. [3] first proposed the idea of searchable encryption, Boneh et al. [4] subsequently proposed the asymmetric searchable encryption scheme is the first public key encryption scheme using keyword search, which solves the problem of untrustworthy server routing, but trapdoor transmission relies on the secure channel, Chang et al. [5] realized a more efficient index-based searchable encryption scheme by establishing a hash table corresponding to keywords and data files, Tian [6] realized identity-based search encryption without a secure channel, although the calculation overhead of the public key is reduced and the CA certification link is omitted, there is a problem of key escrow, Peng [6] et al. putted forward the concept of certificateless public key encryption, which solved the problems of certificate management and key escrow, but it did not support multi-user search and cannot support the sharing of encrypted files, Yang et al. [7] added the nature of multi-user on the basis of certificateless, but it does not support inaccurate keyword search by users. Li [8] used
wildcard technology to realize searchable encryption of fuzzy keywords for the first time, but this scheme returns too many redundant ciphertexts and does not support multi-user search.

In view of the high computational cost and low security of existing searchable encryption schemes and inspired by research ideas on emerging computer-related topics [9],[10],[11],[12] we propose a new certificateless fuzzy keyword searchable encryption scheme. Our scheme has the following characteristics. (1) Introducing KGC to generate partial keys, avoiding the use of public key certificates. (2) Use the identity of the visiting user to realize multi-user search (3) Combine fuzzy keywords set and tree index structure to realize users' fuzzy search. (4) Random values are added in the ciphertext index and trapdoor generation stage to ensure that cloud server provider cannot tamper with the data.

2. Preliminaries
2.1. Bilinear mapping
Give two multiplicative cyclic groups $G_1$ and $G_2$ with the same order as large prime numbers $p$, a bilinear map $e: G_1 \times G_1 \rightarrow G_2$ meet the following properties:

- **Bilinear.** For any generators $g_1 \in G_1$, $g_2 \in G_2$, $a, b \in Z_p^*$, $e(g_1^a, g_2^b) = e(g_1^a, g_2^b)^{ab}$.
- **Non-degenerate.** For any generators $g \in G_1$, $e(g, g) \neq 1$.
- **Computability.** For any generators $g_1, g_2 \in G_1$, Can make $e(g_1, g_2)$ computational.

2.2. Difficult Problem Hypothesis
Give a four-tuple $(g, a, b, c, g) , a, b, c \in Z_p^*$, $g \in G_1$, Calculating $e(g, g)^{abc}$ is a Bilinear Diffie-Hellman (BDH) difficult problem.

**Definition.** If there is no polynomial algorithm adversary (PPT) that can solve BDH with a non-negligible probability, BDH is called a difficult problem.

3. System model
The system model diagram of this scheme as shown in Figure 1, including four entities: Cloud server Provider (CSP) $id_C$, Key Generation Center (KGC), users $id_{uid}$ and data owner $id_{oid}$. CSP is responsible for receiving and storing the ciphertext file and fuzzy keywords index, and returning the ciphertext to. KGC is mainly responsible for generating system parameters and distributing partial keys. uploads trapdoor to CSP and receives ciphertext. is responsible for uploading the ciphertext file and indexing to the CSP.

![Figure 1. System model diagram.](image-url)
4. Scheme Construction and Proof Of Correctness

4.1. Scheme Structure

**Setup**: Enter system parameter $\lambda$, and the Key Generation Center (KGC) selects two multiplicative cyclic groups $G_i$ and $G_j$ with a large prime number $p$, where satisfies the bilinear mapping relationship, and $g$ is the generator of $G_1$, and $m \in \mathbb{Z}_p^*$ is randomly selected as a master key to save and calculate $V = g^m$, select two anti-collision hash functions $H : [0,1]^* \rightarrow G_i$, $H_i : [0,1]^* \rightarrow G_i$, KGC to disclose system parameters.

**Param**: $\text{KGC}$ calculates private keys $psk_a = H(id_a)^m$, $psk_c = H(id_c)^m$ for each access users and cloud server (CSP) $id_a$, and sends them to and $id_c$ through a secure channel.

**KeyGen**: randomly select $x_a \in \mathbb{Z}_p^*$, set the corresponding complete private key pair $sk_a = (x_a, psk_a) = (x_a, H(id_a)^m)$, calculate the public key value $pk_a = g^{x_a}$, similar CSP randomly chooses $x_c \in \mathbb{Z}_p^*$, the corresponding private key pair $sk_c = (x_c, psk_c) = (x_c, H(id_c)^m)$ and public key $pk_c = g^{x_c}$ are respectively.

**Enclnd**: The data owner extracts the key set from the data file $W = \{w_i \in \Delta | 1 \leq i \leq n\}$ that needs to be encrypted, of which $\Delta$ is the key subset initialized. $n$ is the maximum number of keywords, constructs the one-to-one index, and inputs the distance parameter $dis$, and uses the optimized wildcard algorithm and the primary keyword to generate the fuzzy keyword set $\psi_{w_i, dis}$, which can be expressed as (where $w_{i0}$ is the main keyword):

$$\psi_{w_i, dis} = \{w_{i0}, w_{i1}, w_{i2}, \ldots, w_i\}$$

Specifically, if the edit distance $dis$ is 1, the fuzzy keyword set of the keyword apple is:

$$\psi_{\text{apple}, 1} = \{\text{apple}, \text{app}*, \text{a}*, \text{ple}, \text{apple}, \text{*apple}, \text{apple*}\}$$

Compared with the existing fuzzy searchable encryption scheme [13], this paper optimizes the wildcard algorithm, sets the three keywords after the main keyword as common spelling errors, and removes redundant generated keywords. Note that the data owner may not generate fuzzy keyword sets for sensitive keywords. The plaintext index structure of the set keywords is shown in Figure 2.

![Figure 2. Plaintext index structure of fuzzy keywords.](image)

Then randomly generate $t, k \in \mathbb{Z}_p^*$, where $k$ is the symmetric key, and calculate the encrypted ciphertext $CF = C_i = (C_{i1}, C_{i2}, C_{i3})$ of the keyword $w_i$, among them $C_{i1} = e(H_i(w_i)^{y_k}, pk_c), \quad C_{i2} = e(H_i(w_i)^{y_k}, pk_c), \quad C_{i3} = e(H_i(w_i)^{y_k}, pk_c), \quad$, The ciphertext file $CT$ is generated using a symmetric encryption algorithm (AES algorithm, etc.), and finally the data owner sends the ciphertext file $CF$ and $CT$ to the CSP.

**Trapdoor**: User selects the keyword $w' \in \Delta$, randomly selects $z \in \mathbb{Z}_p^*$ and calculates $T_{w'} = (T_1, T_2, T_3)$, where $T_1 = H_i(w')$, $T_2 = g^{z_k} \cdot pk_c$, and $T_3 = pk_c \cdot g^{z_k} \cdot psk_a$.
Search: After receiving the search trapdoor of \( \sigma \), CSP calculates \( \sigma_1 = e(T_{1x}, C_{1x}) \cdot e(H(id_u), C_{1x}) \), \( \sigma_2 = e(T_3, g) \). And through the verification equation \( \sigma_1/\sigma_2 = C_{1x} \) to determine whether the keyword matching is successful, and return the value of the master node corresponding to the ciphertext to \( id_u \).

4.2. Correctness Analysis
CSP calculates after receiving the keyword ciphertext \( CF \) and the trapdoor \( T_x \) of the user:
\[
\begin{align*}
\sigma_1 &= e(T_{1x}, C_{1x}) \cdot e(H(id_u), C_{1x}) = e(H_1(w')^x, g^{\alpha x}) \cdot e(H(id_u), V^m) \\
\sigma_2 &= e(T_3, g) = e(pk_i \cdot g^{\alpha x} \cdot H(id_u)^m, g) \\
&= e(H_1(w')^x, g^{\alpha x}) \cdot e(H(id_u), V^m) = e(H_1(w'), g^{\alpha x}) \\
&= e(H_1(w'^i), pk_i)
\end{align*}
\]
If \( w_1 = w' \) then the equation holds \( \sigma_1/\sigma_2 = C_{1x} \), it means that the CSP successfully matched the ciphertext through the trapdoor uploaded by the user.

5. Safety Analysis
Here we introduce two types of adversaries to play a security game with a challenger, where A1 is defined as an adversary who does not know the master key but can replace the public key of any user. On the contrary, adversary A2 knows the master. The key, but it cannot replace the public key of any user. The two parties involved in the game are opponents, A1 or A2 and challenger X.

Theorem 1. Under the random oracle model, based on the difficult problem of BDH, our scheme satisfies the indistinguishable security under the keyword guessing attack.

Prove: Below we will prove theorem 1 through two security games:

Game 1. Participants are A1 and challenger X. It is stipulated that A1 can make up to \( q_r \) trapdoor queries and create user queries \( q_r \) times. X in order to breaks BDH difficult problems plays game with A1 as follows:

Setup. X select an identity \( id^* \), set \( V = g^x \), and use the steps in the Setup phase to generate system parameters.

Hash Query. X Create a query list \( H \)-list with \( (id, q, h) \) elements. When A1 asks \( id_i \), X answers and updates the list through the following steps:

- If there is a query value \( id_i \) in \( H \)-list, then X returns \( q_i \).
- If \( id_i = id^* \), then X set \( q_i = g^x \). After that, X stores \( (id^*, q, h) \) in \( H \)-list.
- Otherwise, X randomly selects \( h_i \in Z_p^* \) and calculates \( q_i = (g \cdot g^h)^{h_i} \).

Hash1 Query. X Create a query list with \( (id, pk_i, psk_i, x_i) \) elements. When A1 asks \( id_i \), X answers \( pk_i \) and updates \( H_\text{-list} \).

PartialKey Query. After receiving the identity \( id_i \), if \( id_i = id^* \), X terminate the game. (We use SPE1 to indicate this situation.)

SecretValue Query. After receiving the identity \( id_i \), X finds \( q_i \) and sends it to A1. Note that if the corresponding \( pk_i \) is replaced by A1, X returns \( H_\text{-list} \).

PublicKey Query. After receiving the identity \( id_i \), X finds \( pk_i \) and sends it to A1.
ReplacePublicKey Query. After receiving $(id_o, pk_i')$, X replace $x_i$ in $Luser-list$.

Trapdoor Query 1. After receiving the query keyword $w_i$, data owner identity $id_o$ and user identity $id_u$ selected by A1, X sends trapdoor $T_i = h_{w_i}$, and to A1 through the interactive process.

$T_i = h_{w_i} \equiv g^{\epsilon} \cdot psk$

Challenge. After receiving the data owner identity $id_o$, user identity $id_u$, challenge keyword $W_o$ and $W_i$, X generates the corresponding keyword challenge cipher text in the following way.

- If $id_o \neq id_u$, X terminate the game and return a failure message. (SPE3)
- Randomly select $W_i \in \{W_o, W_i\}$ , $t, k \in Z_p^*$, set $t = b \cdot x_i$.
- Calculate $C_{ti} = e(h_{w_i}^{t}, pk_c)$,
- Send the challenge ciphertext $C_t \equiv g^{C_{ti}} \cdot C_{w_i}$ to X.

Trapdoor Query 2. Continue to search for trapdoor of $w_i$, which is similar to trapdoor query 1, except that $w_i$ cannot be a challenge keyword.

Guess. A1 outputs the guess $l' \in \{0, 1\}$ of $l$. If $l' = l$, it means A1 attack is successful.

At this time, $Z = \frac{C_{ti}}{e(g^{\alpha}, g^{\alpha})}$ is calculated by X. If $C_{ti} = e((g^\alpha g^{\gamma})^{\alpha \cdot k}, g^{\alpha \cdot u})$, then $Z = e(g, g)^{\alpha \cdot k}$ can be obtained.

According to SEP1, SEP2, SEP3 we can get $Pr[-SEP3] = 1/q_c$, then the probability of the game continuing in the trapdoor stage is $Pr[-SEP2] \geq (1 - 1/q_c)^\gamma \cdot (1 - (1/q_c + 1))^{\gamma} \geq (1 - 1/q_c)^\gamma / e$, So the probability of X ending the game is $Pr[-SEP1 \cap -SEP2 \cap -SEP3] \geq (1/q_c)(1 - 1/q_c)^\gamma / e = (1 - 1/q_c)^\gamma / e q_c$, So the probability of successfully breaking the BDH problem is at least $\epsilon \geq (1 - 1/q_c)^\gamma / e q_c$. From this equation, we can see that this probability is negligible.

Game 2. Participants are A2 and challenger X, the probability of A2 breaking this scheme is $\zeta'$. We get $\zeta' = (1 - 1/q_c)^\gamma / e q_c$ by analyzing SEP1, SEP2, SEP3, from this equation, we can see that $\zeta'$ is negligible.

In summary, Theorem 1 holds.

6. Performance Analysis

| Schemes | Certificateless | Multi-user | Fuzzy search | EncInd | Trapdoor | Search |
|---------|----------------|------------|--------------|--------|----------|--------|
| [7]     | √              | √          | ×            | 2E+P+2h| 4E+3h    | 2E+4P+h|
| [14]    | ×              | √          | √            | E+P+3h | E+h      | 5E+P+h |
| Ours    | √              | √          | √            | 3E+P+h | 2E+h     | E+3P+h |

In this section, we evaluate the related schemes [7,14] from the perspectives of computing performance and functional characteristics. Among them, we have realized this scheme and scheme by simulation [7], function and calculation cost comparison are shown in Table 1. Where √ means satisfaction, × means dissatisfaction, we define h as the execution time of a hash function, E is the execution time of a power operation, and P is the execution time of a bilinear pairing operation. It can be seen that our scheme has advantages.

The simulation experiment was run on a laptop computer with a 2.7GHZ Intel Core i7-5700HQ processor, 8GB memory and 128GB hard disk space. The simulation program is carried out in the software environment of the 64-bit Windows 10 operating system and the
The comparison of the various stages of the scheme is shown in Figure 3. After adding multiple users, the comparison chart is shown in Figure 4. The experiment shows that our scheme has certain advantages.

![Figure 3. Comparison of the various stages of the scheme.](image1)

![Figure 4. Comparison after joining multiple users.](image2)

7. Conclusion
Certificateless searchable encryption is one of the hotspots in recent years. We propose a new certificateless multi-user fuzzy keyword searchable encryption scheme. Our scheme can be realized fuzzy keyword search requests from multiple users, while avoiding certificate management and key escrow issues, reducing computing overhead and network bandwidth, are very suitable for cloud storage environments. Security analysis shows that our scheme meets the indistinguishability under keyword guessing, and performance analysis shows that our scheme has better performance.

Acknowledgments
This work was supported by the National Natural Science Foundation of China (61662069, 61562077), China Postdoctoral Science Foundation (2017M610817), Lanzhou Science and Technology Project (2013-4-22) and the Foundation of Northwest Normal University (NWNU-LKQN-14-7).

References
[1] Fu Y X, Luo S M and Su J W 2013 Research on Data Security Storage Technology under Cloud Computing Journal of Computer Research and Development 50 136-45
[2] Wu X S and Ma L 2020 Research on Data Security Storage Technology under Cloud Computing Telecom Power Technologies 37 181-3
[3] Song D X, Wagner D and Perrig A 2000 Practical techniques for searches on encrypted data Proceedings of 2000 IEEE Symposium on Security and Privacy pp 44-55
[4] Boneh D, Gentry C, Lynn B, et al 2003 Aggregate and verifiably encrypted signatures from bilinear maps International Conference on the Theory and Applications of Cryptographic Techniques pp 416-432
[5] Chang Y, Mitzenmacher M 2005 Privacy preserving key-word searches on remote encrypted data *International Conference on Applied Cryptography and Network Security* pp 416-432

[6] Peng Y G, Cui J T, Peng C G and Ying Z F 2014 Certificateless public key keyword searchable encryption *China Communications* **11** 100-13

[7] Yang X D, Chen G L, Li T, Liu Y and Zhao X B 2020 Multi-user ciphertext retrieval scheme based on certificateless cryptosystem *Computer Engineering* **46** 129-135

[8] Li J, Wang Q, Wang C, et al 2014 Fuzzy keyword search over encrypted data in cloud computing *International Journal of Engineering Research and Applications* **4** 441-5

[9] Wen S P, Dong M H, Yang Y, Zhou P, Huang T W, and Chen Y R 2019 End-to-end detection-segmentation network for face labeling *IEEE Transactions on Emerging Topics in Computational Intelligence*. doi: 10.1109/TETCI.2019.2947319

[10] Wen S P, Liu W W, Yang Y, Zhou P, Yan Z, Guo Z Y, Chen Y R, and Huang T W 2020 Multi-label image classification via feature/label co-projection *IEEE Transactions on Systems, Man and Cybernetics: Systems*. doi: 10.1109/TSMC.2020.2967071

[11] Wen S P, Liu W W, Yang Y, Zeng Z G, and Huang T W 2019 Generating realistic videos from keyframes with concatenated GANs *IEEE Transactions on Circuits and Systems for Video Technology* **29**(8) 2337-48

[12] Wen S P, Wei H Q, Huang T W, and Zeng Z G 2018 Memristive fully convolutional networks: an accurate hardware image-segmentor in deep learning *IEEE Transactions on Emerging Topics in Computational Intelligence* **2**(5) 324-34

[13] Jiang J, Cai L Q, Wei P C and Li L 2020 Support verifiable ciphertext fuzzy keyword retrieval scheme *Computer Engineering and Applications* **56** 74-80

[14] Li S C and Wang X Y 2016 Multi-user verifying completeness and integrity encrypted keyword query in cloud storage *Computer Engineering and Applications* **52** 132-8