Lightweight searchable encryption scheme based on certificateless cryptosystem

Xiaodong YANG Guilan CHEN Meiding WANG Xizhen PEI
Department of Computer Science and Engineering
Northwest Normal University
Lanzhou, China
y200888@163.com chenalan95@163.com 775631303@qq.com 704546511@qq.com

***Abstract*—Searchable encryption technology can guarantee the confidentiality of cloud data and the searchability of ciphertext data, which has a very broad application prospect in cloud storage environments. However, most existing searchable encryption schemes have problems, such as excessive computational overhead and low security. In order to solve these problems, a lightweight searchable encryption scheme based on certificateless cryptosystem is proposed. The user's final private key consists of partial private key and secret value, which effectively solves the certificate management problem of the traditional cryptosystem and the key escrow problem of identity-based cryptosystem. At the same time, the introduction of third-party manager has significantly reduced the burden in the cloud server and achieved lightweight multi-user ciphertext retrieval. In addition, the data owner stores the file index in the third-party manager, while the file ciphertext is stored in the cloud server. This ensures that the file index is not known by the cloud server. The analysis results show that the scheme satisfies trapdoor indistinguishability and can resist keyword guessing attacks. Compared with similar certificateless encryption schemes, it has higher computational performance in key generation, keyword encryption, trapdoor generation and keyword search.**

Keywords- cloud storage; searchable encryption; certificateless cryptosystem; lightweight

I. INTRODUCTION

The rapid development of cloud computing has enabled a growing number of organizations and individuals to migrate data to cloud servers in order to save local resources [1]. However, once the data owner outsources sensitive data to the cloud server, it will lose complete control over the data, causing unauthorized users and cloud service provider (CSP) to access or maliciously steal sensitive data from data owner. Therefore, the security of data is one of the key issues that cloud computing needs to solve [2]. In order to protect the privacy of users' data, data owners typically encrypt data before it is outsourced. In this way, the availability of outsourced data is limited, and encrypted data can not be retrieved effectively [3,4]. In order to solve the above problems, Song *et al.* [5] first proposed a searchable encryption technology. Searchable encryption has the function of keyword search of ciphertext, which can guarantee the security and privacy of encrypted data, and save a lot of expenses for users. Since it was proposed, many researchers have devoted themselves to secure keyword search through encrypted data. Searchable encryption technology has become an important research hotspot in the field of cloud computing security in recent years [6-8].

After Boneh *et al.* [9] proposed the idea of public key searchable encryption, some searchable encryption schemes with special properties were proposed. Guo *et al.* [10] proposed a certificate-based searchable encryption scheme. This solution improved security, but only met the chosen of plaintext security. Xu *et al.* [11] proposed a new certificate-based encryption scheme, which improved the efficiency of keyword search, but faced complex certificate management overhead. Ni *et al.* [12] proposed an identity-based searchable encryption scheme, which not only solved the traditional public key certificate management problem, but also deleted the specified identity file. However, this scheme only satisfied the indistinguishability under the choose plaintext attack. Zhu *et al.* [13] proposed a new identity-based searchable encryption scheme to achieve sharing of search permissions in the form of agents. But this scheme required a trusted key generation center (KGC) to generate private key. In order to solve this problem, Peng *et al.* [14] proposed a searchable encryption scheme based on the certificateless cryptosystem, which effectively solved the key escrow problem, but only applied to single-user ciphertext data retrieval. At the same time, these searchable encryption schemes greatly increased the workload of the cloud server. Verma *et al.* [15] proposed a lightweight searchable encryption scheme, which introducing third-party manager to reduce the burden of cloud computing. However, this solution faced complex certificate management overhead problems and had certain limitations in practical applications.

Aiming at the problems of large computation overhead and low security in most existing searchable encryption schemes, a new lightweight searchable encryption scheme is proposed in this paper. The scheme has the following characteristics.(1) We use certificateless cryptosystem to encrypt keywords, avoiding the use of public key certificates and supporting ciphertext retrieval by multiple users. (2) We introduce a third-party manager who took on most of the work of the cloud server while ensuring that managers stayed away from sensitive data. (3) We store the file index in a third-party manager, and the file ciphertext is stored in the cloud server, ensuring that the file index is not known by the cloud server from the source. (4) We introduce random numbers which is linked with encrypted files to ensure that the cloud server quickly returns the correct file ciphertext. The analysis results show that the scheme has high computing performance and reduces the load of the cloud server.

978-1-7281-4689-8/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMCCE48743.2019.00155

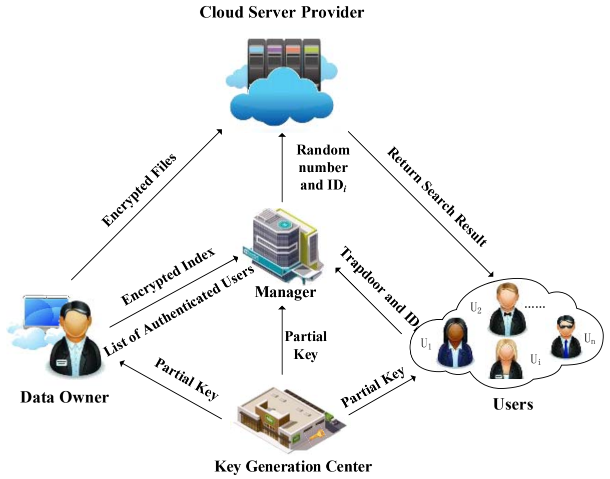

Figure 1. System model.

II. OUR LIGHTWEIGHT SEARCHABLE ENCRYPTION SCHEME

A. System Model

The system model of the solution in this paper is shown in Fig. 1. It includes KGC, Data Owner (DO), *n* access users {U$_1$, U$_2$,⋯, U$_n$}, CSP, and third-party manager (Mng). KGC is primarily responsible for generating system parameters and partial private keys for each entity. DO is primarily responsible for generating random numbers, file ciphertext, file indexes, and authenticated user lists. Access user generates a keyword trapdoor that is sent to Mng at the same time and decrypts the file ciphertext returned by the CSP. Mng is mainly responsible for storing file index and authentication user list, responding to the user's search request, and sending the document random number and the access user identity to the CSP. The CSP is primarily responsible for storing the file ciphertext and returning the search results of the access user.

B. Scheme Construction

1) **Setup** : KGC selects two multiplicative cyclic groups G_1 and G_2 with prime p , a generator g of a group G_1, a bilinear map $e: G_1 \times G_1 \to G_2$ and an anti-collision hash function $H:\{0,1\}^* \to G_1$. It randomly selects $x \in Z_p^*$, secretly saves the master key $msk = x$, calculates $P = g^x$ and publishes the system public parameter $param = (g, G_1, G_2, e, p, H, P)$.

2) **PatialKeyGen** : KGC calculates a partial private key $psk_{id_o} = H(id_o)^x$ for the received DO identity id_o , and sends a partial private key psk_{id_o} to the accessing user through the secure channel. Similarly, the Mng with the identity id_m obtains the partial private key $psk_{id_m} = H(id_m)^x$.

3) **KeyGen** : DO randomly selects $s_o \in Z_p^*$, calculates the public key $pk_{id_o} = g^{s_o}$, and sets its own final private key $sk_{id_o} = (s_o, psk_{id_o}) = (s_o, H(id_o)^x)$. Similarly, Mng randomly selects $s_m \in Z_p^*$ and sets its own final private key $sk_{id_m} = (s_m, psk_{id_m}) = (s_m, H(id_m)^x)$ and public key $pk_{id_m} = g^{s_m}$.

4) **Enc** : First, DO extracts the keyword $w_j \in \psi (1 \le j \le m)$ from the data file F , where ψ is the set of all keywords. Then it randomly selects the random number $r_j \in Z_p^*$ and a random element $nonce \in Z_p^*$ which is linked with the document F , and calculates the encrypted ciphertext $C_j = (C_{j1}, C_{j2}, C_{j3})$ of the keyword w_j , where $C_{j1} = g^{r_j}$, $C_{j2} = H(w_j)^{s_o} \cdot H(pk_{id_m})^{r_j} \cdot H(id_o)^x$ and $C_{j3} = H(id_o) \cdot H(id_m)$ .It uses a symmetric encryption algorithm (such as AES algorithm) to encrypt the file F to obtain the corresponding encrypted file *CT*. Finally, it sends the encrypted file *CT* and its linked random number *nonce* to the CSP. The encrypted file index $C_F = (C_1, C_2, \ldots, C_m)$, the file random number *nonce* and the authenticated user list L_F are sent to Mng, where L_F holds the identity of the authenticated user.

5) **Trapdoor** : First, the access user U$_i$ selects the search keyword $w' \in \psi$. Then it randomly selects $r' \in Z_p^*$, calculates $T_1 = (pk_{id_m})^{r'} = g^{s_m r'}$, $T_2 = H(e(pk_{id_o}, H(w')^{r'}))$ and $T_3 = P^{r'}$. Finally, it sends the identity id_i and search trap $T_{w'} = (T_1, T_2, T_3)$ to the CSP.

6) **Search** : After receiving the search request from the access user U$_i$, Mng first determines whether the user identity id_i exists in L_F . If so, it calculates $C_4 = C_{j2} \cdot H(id_m)^x / H(C_{j1}{}^{s_m})$ and $\sigma = e(T_1, C_4{}^{1/s_m}) / e(T_3, C_{j3})$. Then, it verifies whether the equation $H(\sigma) = T_2$ is true. If the equation is true, the keyword matching is successful, and the matching document random number *nonce* and the access user id_i are sent to the CSP. Otherwise, the matching failure information is sent to the U$_i$. Finally, the CSP queries all encrypted file according to *nonce* , and returns the corresponding file to U$_i$.

III. SECURITY ANALYSIS AND PERFORMANCE EVALUATION

A. Security Analysis

Theorem 1. *Under the DBDH assumption, the scheme of this paper satisfies the trapdoor indistinguishability under the random oracle model.*

Proof. Suppose that the polynomial time attacker $\mathcal{A}$ can break the trapdoor indistinguishability of the scheme with a non-negligible probability ε after performing the most q_T times trapdoor queries (q_T is a positive integer), there exists an algorithm $\mathcal{C}$ that can solve the DBDH problem with a non-negligible probability ε' . Given $(g, g^a, g^b, g^c) \in G_1^4$ and

$Z \in G_2$, $\mathcal{C}$ performs the following interactive game with $\mathcal{A}$ in order to distinguish whether Z is $e(g,g)^{abc}$ or an element in G_2.

Key generation: $\mathcal{C}$ randomly selects $x,k,s_o \in Z_p^*$, and sets the master key $msk = x$. It calculates the public key $pk_{id_o} = u_1^{s_o}$ of the DO, the public key $pk_{id_m} = u_2$ of Mng, and send pk_{id_m} to $\mathcal{A}$. Specifically, the private key $sk_{id_o} = (s_o\alpha, H_1(id_o)^x)$ of the DO and the private key $sk_{id_m} = (\beta, H_1(id_m)^x)$ of the Mng are implied here.

Hash Query: $\mathcal{A}$ accesses the random oracle H to obtain the corresponding hash value.

$\mathcal{C}$ Creates $H-list$ list of elements of the form (w_j, h_j, a_j, c_j), initialized to null. When $\mathcal{A}$ initiates an query about the keyword $w_j \in \{0,1\}^*$, $\mathcal{C}$ responds as follows.

- If there is a corresponding item (w_j, h_j, a_j, c_j) of w_j in $H-list$, $\mathcal{C}$ returns $H(w_j) = h_i$.
- Otherwise, $\mathcal{C}$ randomly selects $a_j \in Z_p^*$, $c_j \in \{0,1\}$, and sets $\Pr[c_j = 0] = 1/(q_T + 1)$. If c_j=0, it calculates h_j = $u_3 \cdot g^{a_j}$ .if c_j=1, it calculates $h_j = u_1^{a_j}$ .It returns h_j to $\mathcal{A}$, and stores (w_j, h_j, a_j, c_j) in the table.

Trapdoor Query 1: When the query keyword w_j selected by $\mathcal{A}$ is received, $\mathcal{C}$ generates a corresponding search trapdoor in the following manner.

- $\mathcal{C}$ asks hash query about w_j to get the corresponding item (w_j, h_j, a_j, c_j).
- $\mathcal{C}$ judges the value of c_j. If c_j=0, the game terminates and returns the failure information. Otherwise, it returns h_j, randomly selects $r_j' \in Z_p^*$, and calculates trapdoors $T_1 = u_2^{r_j'}$, $T_2 = H(e(u_1^{s_o}, h_j^{r_i'})) = H(e(u_1^{s_o}, u_1^{a_j r_j'}))$ and T_3 $= P^{r_j'}$. $\mathcal{C}$ sends $T_{w'} = (T_1, T_2, T_3)$ to $\mathcal{A}$.

Challenge: After receiving the challenge keywords W_0 and W_1 selected by $\mathcal{A}$, $\mathcal{C}$ generates the corresponding challenge traps in the following manner.

- Hash query is made for W_0 and W_1, $\mathcal{A}$ obtains the corresponding items (W_0, h_0, a_0, c_0) and (W_1, h_1, a_1, c_1).
- $\mathcal{C}$ judges the value of c_j, if c_0=1 and c_1=1, the game terminates and returns a failure message. If c_0=0 and c_1=0, $\mathcal{C}$ randomly select $b \in \{0,1\}$, let c_b=0. Otherwise, At least one of c_0 and c_1 is 0. $\mathcal{C}$ randomly selects $r_b \in Z_p^*$, lets $s_o r_b = \beta$, calculates $T_1^* = (pk_{id_m})^{r_b}$=$g^{s_m r_b}$, $T_2^* = H(e(pk_{id_o}, H(w')^{r_b})) = H(e(u_1^{s_o}, (u_3 \cdot g^{a_j})^{r_b})) = H(e$ $(g,\mathrm{g})^{\alpha\beta(\gamma + a_j)})$ and $T_3^* = P^{r_b} = g^{x r_b}$.$\mathcal{C}$ defines $T_2^* = H(Z \cdot$ $e(g,\mathrm{g})^{\alpha\beta a_j})$, and sends the challenge trap $T_{w'}^* = (T_1^*, T_2^*,$ $T_3^*)$ to $\mathcal{A}$.

Trapdoor Query 2: $\mathcal{A}$ continues to query search trapdoor about w_j, and the process is basically the same as the trapdoor query 1, which requiring $w_j \neq W_0$ and $w_j \neq W_1$.

Output: $\mathcal{A}$ output a guess $b' \in \{0,1\}$ to b. If b'=b, $\mathcal{A}$ attacks successfully, and $\mathcal{C}$ successfully distinguishes whether Z is an element of G_2 or $e(g,g)^{abc}$.

It can be seen from the above analysis that the game has a termination condition in the trapdoor query phase and the challenge phase, where ε_1 and ε_2 are respectively used to indicate that the game is not terminated in the trapdoor query phase and the challenge phase. In the trapdoor query phase, $\mathcal{A}$ dose not repeat the query keyword, and the game is terminated when $c_i = 0$. According to the hash query phase, the probability of $c_i = 0$ is $1/(q_T + 1)$. $\mathcal{A}$ can perform up to q_T trapdoor queries, and the probability that the game is not terminated in the trapdoor query phase is $\Pr[\varepsilon_1] \geq (1 - (1/(q_T + 1)))^{q_T} \geq 1/e$. In the challenge phase, $\mathcal{A}$ chooses to challenge keywords W_0 and W_1, and the game terminates when $c_0 = 1$ and $c_1 = 1$. According to the hash query phase, $\Pr[c_i = 0] = 1/(q_T + 1)(i = 0,1)$ can be known. c_0 and c_1 are independent of each other, so $\mathcal{A}$ has a game termination probability of $\Pr[c_0 = c_1 = 1] = (1 - 1/(q_T + 1))^2 \leq 1 - 1/q_T$ in the challenge phase. The probability that the game is not terminated in the challenge phase is $\Pr[\varepsilon_2] \geq 1/q_T$. Since ε_1 and ε_2 are independent of each other, the probability that ε_1 and ε_2 occur simultaneously is $\Pr[\varepsilon_1 \wedge \varepsilon_2] = \Pr[\varepsilon_1] \cdot \Pr[\varepsilon_2]$ $= (1/e) * (1/q_T) = 1/eq_T$. The probability that the game is not terminated during the simulation is $1/eq_T$. Therefore, the probability that $\mathcal{C}$ does not terminate and $\mathcal{A}$ successfully break the scheme during the game is $\varepsilon' = \varepsilon / eq_T$. The DBDH assumption is a well-recognized problem, so ε' is a negligible function. Therefore, the probability $\varepsilon = \varepsilon' eq_T$ of $\mathcal{A}$ breaking this scheme should also be a negligible function. There is no adversary can break through the solution in a polynomial time with a non-negligible advantage.

B. *Performance Evaluation*

In the certificateless encryption scheme, the user's private key consists of two parts: a partial private key and a secret value. The partial private key of the user is generated by a trusted KGC, and the user independently generates his own secret value and corresponding public key. Since KGC cannot obtain the user's secret value, the certificateless encryption scheme can solve the key escrow problem in the identity-based encryption scheme. Both Wu's scheme [15] and the scheme of this paper adopt a certificateless encryption scheme, which effectively reduces the complexity of certificate maintenance costs and solves the key escrow problem. Verma’s scheme [15] and the proposed scheme introduce a third-party manager to effectively reduce the cloud server overhead and ensure that the CSP quickly returns the correct encrypted file.

The following is a performance analysis of the proposed scheme, mainly considering the time-consuming bilinear pair operation and exponential operation. For ease of description, let P and E denote a bilinear pair operation and an exponential operation, respectively. Table 1 gives a comparison of the performance of several searchable encryption schemes [15,16]. The literature [15] faces the problem of the management overhead of certificates. The literature [16] only supports single users and does not support lightweight ciphertext retrieval. The proposed scheme addresses these shortcomings, and the search phase of the proposed scheme is completed on Mng. The server only needs to query the random number linked with the file and return the corresponding encrypted file. Therefore, the scheme of this paper has high computational efficiency and security.

IV. CONCLUSION

Based on the certificateless cryptosystem, we propose a new lightweight public key searchable encryption scheme whose security depends on the DBDH hypothesis. The solution introduces a third-party manager, which reduces cloud server workload and supports multi-user ciphertext retrieval. At the same time, the proposed scheme introduces random numbers which is linked with encrypted files to ensure that the cloud server can quickly return the correct encrypted file. The analysis results show that the scheme has high computational performance. It is suitable for multi-user cloud ciphertext retrieval environment. However, this scheme resists keyword guessing attacks under the random oracle model. Therefore, the next phase will study lightweight ciphertext retrieval scheme under the standard model.

ACKNOWLEDGMENT

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

REFERENCES

[1] Wang Y, Li J, and Wang H H, “Cluster and cloud computing framework for scientific metrology in flow control,” Cluster Computing. vol. 22, 2019, pp. 1189-1198.

[2] Karajeh H, Maqableh M, Masa’deh R. “Privacy and security issues of cloud computing environment,” Proceedings of the 23rd IBIMA Conference Vision. 2020, pp. 1-15.

[3] Altowaijri S M. “An architecture to improve the security of cloud computing in the healthcare sector,” Smart Infrastructure and Applications. Springer, Cham, 2020, pp. 249-266.

[4] Zhang Y, Xiang Y, Zhang L Y. “Cloud computing security,” Secure Compressive Sensing in Multimedia Data. Cloud Computing and IoT. Springer, Singapore, 2019, pp. 63-82.

[5] Song D X, Wagner D, Perrig A. “Practical techniques for searches on encrypted data,” Proceedings of 2000 IEEE Symposium on Security and Privacy. 2000, pp. 44-55.

[6] Bost R, Fouque P A. “Security-efficiency tradeoffs in searchable encryption,” Privacy Enhancing Technologies. vol. 4, 2019, pp. 132-151.

[7] Deng X, Cheng H, Sun B, et al. “ID-based deletion searchable encryption scheme,” IOP Conference Series: Earth and Environmental Science. vol. 234. IOP Publishing, 2019.

[8] Chen L, Lee W K, Chang C C, et al. “Blockchain based searchable encryption for electronic health record sharing,” Future Generation Computer Systems. vol. 95. 2019, pp. 420-429.

[9] Boneh D, Di Crescenzo G, Ostrovsky R, et al. “Public key encryption with keyword search,” Proceedings of International conference on the theory and applications of cryptographic techniques. Berlin: Springer, 2004: pp. 506-522.

[10] Guo Yuyan, Jiang Mingming, Song Wanqian. “Proof of security-based elastic leakage based on certificate encryption scheme,” Journal of Huaibei Normal University. vol. 40, 2019, pp. 19-25.

[11] Xu Hailin, Li Yang. “Certificate-based encryption scheme with keyword search for efficient unparalleled linear pairs,” Journal of Computer Applications. vol. 38, 2018, pp. 379-385.

[12] Ni Lulin, Xu Chungen. “Identity-based dynamic searchable encryption scheme,” Computer Engineering. vol. 45, 2019, pp. 136-140.

[13] Zhu Minhui, Chen Yanxi, Hu Yuanyuan. “Identity-based searchable encryption scheme supporting proxy re-encryption,” Computer Engineering. vol. 45, 2019, pp. 129-135.

[14] Peng Yanguo, Cui Jiangtao, Peng Changgen, Ying Zuofu. “Certificateless public key keyword searchable encryption ,” China Communications. vol. 11, 2014, pp. 100-113.

[15] Verma S. “A new lightweight approach for multiuser searchable encryption in the cloud,” Proceedings of Communication, Networks and Computing. vol. 839, 2019, pp. 49-63

[16] Wu Qiying, Ma Jianfeng, Li Hui, Miao Yinbin. “Ciphertext retrieval without certificate connection keyword,” Journal of Xidian University. vol. 44, 2017, pp. 55-60.

TABLE I. PERFORMANCE COMPARISON OF SEVERAL TYPES OF SEARCHABLE ENCRYPTION SCHEMES

schemes	**Key generation**	**Keyword encryption**	**Trapdoor generation**	**Search verification**	**Certificateless**	**Lightweight**	**Multi-user**
Verma scheme [15]	$3E$	$3E$	P+2E	P+2E	×	✓	✓
Wu scheme [16]	$9E$	$8E$	$8E$	$4P$	✓	×	×
Our scheme	$5E$	$3E$	P+3E	$2P$+2E	✓	✓	✓