SCIENCE CHINA
Information Sciences

• LETTER •

March 2019, Vol. 62 039111:1–039111:3
https://doi.org/10.1007/s11432-017-9508-3

Secure key-aggregation authorized searchable encryption

Haijiang WANG[1,2], Xiaolei DONG[3*], Zhenfu CAO[3*], Dongmei LI[2] & Nanyuan CAO[2]

[1] *School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou* 310023*, China;*
[2] *Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240*, China;*
[3] *Shanghai Key Lab for Trustworthy Computing, East China Normal University, Shanghai* 200062*, China*

Received 16 December 2017/Revised 12 April 2018/Accepted 4 May 2018/Published online 12 February 2019

Citation Wang H J, Dong X L, Cao Z F, et al. Secure key-aggregation authorized searchable encryption. Sci China Inf Sci, 2019, 62(3): 039111, https://doi.org/10.1007/s11432-017-9508-3

Dear editor,
Selective sharing of encrypted documents with other users makes the cloud storage system more flexible and effective [1]. For safety reasons, data owners use different keys to encrypt different documents. However, data owners need to distribute all keys to the query users for data sharing. Similarly, the query users must submit the same number of trapdoors in order to complete search queries over the entire encrypted documents. Key-aggregation searchable encryption (KASE) cryptosystem was proposed to solve the above problem. A KASE scheme enables a data owner to share encrypted documents with other query users by distributing only a single aggregated searchable secret key, and the authorized query user only needs to submit a single trapdoor to the cloud server to perform keyword search over the shared encrypted documents.

To facilitate encrypted data sharing, Bao et al. proposed a multiple users searchable scheme in [2]. In this scheme, each user possesses a distinct secret key and can search through all the encrypted documents. Dong et al. [3] proposed a data sharing scheme supporting keyword search based on proxy cryptography. However, in these schemes, a trusted party is assumed to be responsible for managing the legitimate user's keys. To enable authorized keyword search, Li et al. [4] built an authorized private-keyword search scheme by employing hierarchical predicate encryption. In the proposed scheme, some trusted authorities stay online to generate trapdoors for query users. Based on ABE, Shi et al. [5] presented a more general construction. In the scheme, the data owner could identify the authorized query user by embedding an access policy into the encrypted data, and the query user can submit keyword search independently. By using the technique of broadcast encryption [6], Cui et al. [7] proposed the primitive key-aggregation searchable encryption to solve the problem. However, we found some security issues in scheme [7]. By proposing the "collusion attack", we show that scheme [7] is not completely secure.

Collusion attack. First, we review the construction of scheme [7]. Suppose a data owner has n documents. The data owner publishes the system public parameters

$$\mathrm{PP} = (g, g_1, g_2, \ldots, g_n, g_{n+2}, \ldots, g_{2n}).$$

For subset $S \subseteq \{1, 2, \ldots, n\}$, the data owner generates an aggregated searchable secret key by computing

$$K_{\mathrm{agg}} = \prod_{j \in S} g_j^a.$$

This construction treats the aggregated key as a simple combination of record-independent keys. Obviously, it cannot resist the following collusion attack:

(1) Suppose user A gets an aggregated searchable secret key $K_{\mathrm{agg}_1} = (g_1 \cdot g_2)^a$ of documents sub-

[*] Corresponding author (email: dongxiaolei@sei.ecnu.edu.cn, zfcao@sei.ecnu.edu.cn)

Table 1 Functionality comparison

	[4]	[5]	[7]	Our scheme
Key-aggregation authorization	×	×	√	√
Autonomic query	×	×	√	√
Collusion attack resistant	√	√	×	√
Secret-key-extraction attack resistant	√	√	×	√

set $\{1,2\}$, and user B gets the aggregated searchable secret key $K_{\mathrm{agg}_2} = (g_3 \cdot g_4)^a$ of documents subset $\{3,4\}$.

(2) A and B can combine their aggregated searchable secret keys to get the right to operate on documents set $S = \{1,2,3,4\}$ by computing

$$K_{\mathrm{agg}_{1\wedge 2}} = K_{\mathrm{agg}_1} \cdot K_{\mathrm{agg}_2} = \prod_{j\in S} g_j^a.$$

As shown above, the authorized query user can conspire to obtain additional rights.

Our contribution. We propose a secure searchable encryption scheme which supports key-aggregation authorization. With a traditional method, to authorize keyword search right to a user for documents encrypted by different keys, data owner must allocate the same number of keys, which leads to high communication costs. However, in our scheme, we achieve authorization by allocating an aggregated searchable secret key. Our scheme supports autonomic query by distributing searchable secret keys to query users, so that the query users could generate trapdoors for any keyword by themselves. Our scheme is collusion attack resistant, that is, query users cannot conspire to obtain additional rights. Furthermore, our scheme is secret-key-extraction attack resistant, that is, an adversary cannot extract the secret key from a valid trapdoor. Using the “aggregation” technology, the sizes of searchable secret key and trapdoor are constant in scheme [7] and our scheme. Compared with [7], our new system only sacrifices tiny size of searchable secret key to resist collusion attack. Besides, the size of the aggregated searchable secret key is independent of the number of encrypted records. Table 1 gives the comparison between our scheme and some other related studies.

Our construction.

(1) Setup. On inputting security parameter 1^λ and the number of documents n, the setup algorithm prepares the parameters for the system as follows:

- Choose a bilinear pairing group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with the same order p and let $\dot{g}$ be a generator of $\mathbb{G}_1$ and $\ddot{g}$ be a generator of $\mathbb{G}_2$.
- Randomly select $\alpha \in \mathbb{Z}_p$ and compute $\mu_i, \nu_i, i = 1,2,\ldots,n$ as $\mu_i = \dot{g}^{\alpha^i}, \nu_i = \ddot{g}^{\alpha^i}$.
- Pick 4 collision-resistant Hash functions: $H_2 : \{0,1\}^* \rightarrow \{0,1\}^{l_\theta}$[1], $H_3 : \{0,1\}^* \rightarrow \{0,1\}^{l_m}$[2], $H_1, H_4 : \{0,1\}^* \rightarrow \mathbb{Z}_p$.
- Randomly select $a \in \mathbb{Z}_p$, and set the parameters as $\mathrm{PP} = (\mu_i, \nu_i, h, e(\dot{g}, \ddot{g}), H_1, H_2, H_3, H_4)$, $\mathrm{MSK} = (a, \alpha, \dot{g})$.

(2) Encrypt. Data owners use this algorithm to encrypt their documents. On inputting each document message M, keyword w, public parameters PP, master secret key MSK and identity ID_i, data owners encrypt their documents by computing:

- Randomly select $\theta \in \{0,1\}^{l_\theta}$ and compute $t = H_4(M, \theta)$. For each document, let $\mathrm{ID}_i = (y_1, y_2, \ldots, y_n)$ be the identity.
- Compute an $(n-1)$-degree polynomial $\pi(x, \mathrm{ID})$ as $\pi(x, \mathrm{ID}) = \prod_{i=1}^{n} (x+i)^{1-y_i}$.
- Compute C_1 as $C_1 = (\ddot{g}^{\pi(\alpha,\mathrm{ID})})^t = (\ddot{g}^{\pi_0} \prod_{i=1}^{n-1} \nu_i^{\pi_i})^t$.
- Compute $C_{2,i}$ $(i = 0,1,2,\ldots,n)$ as $C_{2,i} = \mu_i^t$ and set $C_{2,0} = C_{2,1}^{(a+H_1(w))}$.
- Compute $e(\dot{g}, \ddot{g})^t$ and C_0, C_3, C_4 as $C_0 = e(\dot{g}, \ddot{g})^{t(a+H_1(w))}, C_3 = H_2(e(\dot{g}, \ddot{g})^t) \oplus \theta, C_4 = H_3(\theta) \oplus M$.
- Set index ciphertext I_w as $(C_0, C_1, \{C_{2,i}$ $(i = 0,1,2,\ldots,n)\})$ and set document ciphertext CT as $(\{C_{2,i}$ $(i = 0,1,2,\ldots,n)\}, C_3, C_4)$.

(3) KeyGen. By distributing an aggregated searchable secret key, the data owner can authorize keyword search right of any subset $S \subseteq \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$ to some query users. On inputting master secret key MSK and subset S, the aggregated key SK is output as follows:

- Let $\mathbb{S} = (x_1, x_2, \ldots, x_n)$, $x_i \in \{0,1\}$ be the aggregation string, where $x_i = 1$ if and only if $\mathrm{ID}_i \in S$. Compute an $(n-|S|)$-degree polynomial $\pi(x, \mathbb{S})$ as $\pi(x, \mathbb{S}) = \prod_{i=1}^{n} (x+i)^{1-x_i}$.
- Randomly select $s \in \mathbb{Z}_p$ and generate the aggregated key for $\mathbb{A}$ as $D_1 = \dot{g}^{\frac{as}{\pi(\alpha,\mathbb{S})}}, D_2 = \ddot{g}^{\frac{a(s-1)}{\alpha}}, D_3 = \dot{g}^{\frac{s}{\pi(\alpha,\mathbb{S})}}, D_4 = \ddot{g}^{\frac{s-1}{\alpha}}$.

(4) Trapdoor. For all the documents related to the aggregated key, the query user generates a single trapdoor by computing $t_w = (S, T_1 = D_1 \cdot D_3^{H_1(w)}, T_2 = D_2 \cdot D_4^{H_1(w)})$, and then submits

[1] l_θ the length of a random string under system security parameter.

[2] l_m denotes the length of message.

the trapdoor to cloud server.

(5) Test. Cloud server can search over the entire encrypted data I_w with the submitted trapdoor t_w:

- If the $\mathrm{ID}_i \notin S$, then abort. Otherwise, compute z_i $(i = 1, 2, \ldots, n)$ as $z_i = x_i - y_i$.
- Compute U_1, V_1, W_1[3] as

$$
\begin{aligned}
U_1 &= e\left(C_{2,0}, \prod_{i=1}^{n-1} \nu_{i-1}^{\Phi_i}\right) \\
&= e(\dot{g}, \ddot{g})^{(t\Phi(\alpha) - t\Phi_0)\cdot(a + H_1(w))}, \\
V_1 &= e\left(\prod_{i=1}^{n} C_{2,i}^{\Phi_{i-1}}, T_2\right) \\
&= e(\dot{g}, \ddot{g})^{(ts\Phi(\alpha) - t\Phi(\alpha))\cdot(a + H_1(w))}, \\
W_1 &= e(T_1, C_1) = e(\dot{g}, \ddot{g})^{ts\Phi(\alpha)\cdot(a + H_1(w))}.
\end{aligned}
$$

The algorithm outputs 0 or 1 by judging

$$\left(\frac{W_1}{U_1 \cdot V_1}\right)^{\frac{1}{\Phi_0}} \overset{?}{=} C_0.$$

(6) Decrypt. The decryption algorithm works as follows:

- Compute $z_i, i = (1, 2, \ldots, n)$ as $z_i = x_i - y_i$.
- Compute U_2, V_2, W_2 as

$$
\begin{aligned}
U_2 &= e\left(C_{2,1}, \prod_{i=1}^{n-1} \nu_{i-1}^{\Phi_i}\right) = e(\dot{g}, \ddot{g})^{(t\Phi(\alpha) - t\Phi_0)}, \\
V_2 &= e\left(\prod_{i=1}^{n} C_{2,i}^{\Phi_{i-1}}, D_4\right) = e(\dot{g}, \ddot{g})^{(ts\Phi(\alpha) - t\Phi(\alpha))}, \\
W_2 &= e(D_3, C_1) = e(\dot{g}, \ddot{g})^{ts\Phi(\alpha)}.
\end{aligned}
$$

- Compute $e(\dot{g}, \ddot{g})^t = (\frac{W_2}{U_2 \cdot V_2})^{\frac{1}{\Phi_0}}$.
- Compute the random number θ by $\theta = H_2(e(\dot{g}, \ddot{g})^t) \oplus C_3$. The algorithm outputs M by computing

$$M = H_3(\theta) \oplus C_4.$$

We prove our proposed key-aggregation authorized searchable encryption scheme is secure in the following theorems.

Theorem 1. IND-CPA. The security of our scheme is based on the hardness of aMSE-DDH problem: an adversary cannot know any plaintext information of the encrypted data.

Theorem 2. The keyword-trapdoor indistinguishability security of our scheme is based on the assumption of pseudo-random generator: an adversary cannot to distinguish the output of a pseudo-random generator from real random strings.

Theorem 3. Collusion attack resistant: given searchable secret key K_{agg_1} for subset S_1 and searchable secret key K_{agg_2} for subset S_2, a malicious user cannot forge a new valid searchable secret key $K_{\mathrm{agg}_{1\wedge 2}}$.

Theorem 4. Trapdoor forgery attack resistant: given some trapdoors, a malicious user is unable to forge a new valid trapdoor.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61632012, 61672239), in part by Shanghai High-Tech Field Project (Grant No. 16511101400), and in part by Fundamental Research Funds for the Central Universities.

References

1 Liang J, Han W L, Guo Z Q, et al. DESC: enabling secure data exchange based on smart contracts. Sci China Inf Sci, 2018, 61: 049102

2 Bao F, Deng R H, Ding X H, et al. Private query on encrypted data in multi-user settings. In: Proceedings of International Conference on Information Security Practice and Experience. Berlin: Springer, 2008. 71–85

3 Dong C Y, Russello G, Dulay N. Shared and searchable encrypted data for untrusted servers. J Comput Secur, 2011, 19: 367–397

4 Li M, Yu S C, Cao N, et al. Authorized private keyword search over encrypted data in cloud computing. In: Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS), 2011. 383–392

5 Shi J, Lai J Z, Li Y J, et al. Authorized keyword search on encrypted data. In: Proceedings of European Symposium on Research in Computer Security. Berlin: Springer, 2014. 419–435

6 Boneh D, Gentry C, Waters B. Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Advances in Cryptology—CRYPTO. Berlin: Springer, 2005. 3621: 258–275

7 Cui B J, Liu Z L, Wang L Y. Key-aggregate searchable encryption (KASE) for group data sharing via cloud storage. IEEE Trans Comput, 2016, 65: 2374–2385

[3] $\Phi(x, \mathrm{ID}, \mathbb{S})$ is the $(n - |S| - 1)$-degree polynomial defined as $\Phi(x, \mathrm{ID}, \mathbb{S}) = \prod_{i=1}^{n}(x + i)^{z_i}$. Φ_i is the coefficient of x^i, and $\Phi_0 \neq 0$.