ITC 1/47
Journal of Information Technology and Control
Vol. 47 / No. 1 / 2018
pp. 56-62
DOI 10.5755/j01.itc.47.1.16137

On Security of a Secure Channel Free Public Key Encryption with Conjunctive Field Keyword Search Scheme

Received 2017/08/28 | Accepted after revision 2018/01/08

http://dx.doi.org/10.5755/j01.itc.47.1.16137

On Security of a Secure Channel Free Public Key Encryption with Conjunctive Field Keyword Search Scheme

Yang Lu, Gang Wang, Jiguo Li
College of Computer and Information, Hohai University, No.8, Focheng Xi Road, Jiangning District, Nanjing, China
e-mails: luyangnsd@163.com, wg15951977612@163.com, lijiguo@hhu.edu.cn

Corresponding author: luyangnsd@163.com

Public key encryption with keyword search is a practical cryptographic paradigm that enables one to search for the encrypted keyword without compromising the security of the original data. Recently, Hwang *et al.* proposed a secure channel free public key encryption with conjunctive field keyword search (SCF-PECKS) scheme and claimed that their scheme can withstand the keyword guessing attack and does not need the secure channel. In this paper, by presenting three concrete attacks, we demonstrate that Hwang *et al.*'s SCF-PECKS scheme fails to achieve the security against keyword guessing attacks by either outsider attackers or malicious insider servers. The presented attacks show that Hwang *et al.*'s scheme is vulnerable to the keyword guessing attack regardless of whether the trapdoors are sent via public channel or secure channel. Therefore, devising a secure SCF-PECKS scheme remains an unsolved problem until now.

KEYWORDS: public key encryption, conjunctive field keyword search, secure channel free, keyword guessing attack.

1. Introduction

With the development of cloud computing technology, the amount of sensitive data to be stored on the cloud servers is rapidly increasing. By using the cloud storage, the users can get the convenient service and greatly reduce the cost of local data management. To protect the confidentiality of the data, more and more companies and individuals choose to encrypt their data before outsourcing them to the cloud storage servers. By using the encryption techniques, the sensitive data are transformed into the random strings

and are not readable to anyone except the holder of the corresponding decryption key. Thus, the data confidentiality is guaranteed. However, the application of traditional encryption techniques makes it difficult for the cloud storage server to selectively retrieve the encrypted data. A common solution is to allow the user to download all his/her encrypted data from the cloud storage server and then search for them whenever he wants to retrieve the data. In this way, the user has to download and decrypt all encrypted data, regardless of what data he/she is searching for. Obviously, this solution is extremely inefficient due to the high cost of network transmission and the heavy overhead at the user devices.

In order to solve the ciphertext retrieval problem, the notion of keyword search over encrypted data (*i.e.*, searchable encryption) was introduced [26, 2]. In [26], Song *et al.* proposed the first searchable encryption scheme in symmetric cryptography. Searchable encryption enables the user to search encrypted data without revealing the underlying plaintexts and the searched keywords. Therefore, it provides a promising solution to address the ciphertext retrieval in clouds [9-11, 29]. Public key encryption with keyword search (PEKS) was proposed by Boneh *et al.* [2] in 2004. In Boneh *et al.*'s proposal, a PEKS scheme contains three entities: sender, receiver and server (also called tester). The sender encrypts the data with a standard public key encryption (PKE) scheme and then appends a PEKS ciphertext of a keyword to the encrypted data. To retrieve the encrypted data with keyword ciphertext on the server, the receiver sends a trapdoor of the searched keyword to the server. After receiving the trapdoor, the server can test whether the keyword associated with the encrypted data is the same as the keyword encoded in the trapdoor without revealing any information about the searched keyword and the original data. Finally, the server returns the encrypted data containing the searched keyword to the receiver.

Following Boneh *et al.*'s work [2], searchable PKE has attracted great attention from the research community and lots of PEKS schemes [8, 12, 14, 18, 19, 28] have been presented. To convey the trapdoors securely, PEKS needs a secure channel between the server and each receiver. However, it is well known that building a secure channel is very expensive. To solve the problem, Baek *et al.* [1] designed a new framework that removes the secure channel between the server and the receiver, which is called secure channel free public key encryption with keyword search (SCF-PEKS) or designed server public key encryption with keyword search (dPEKS). The basic idea of SCF-PEKS is to make the server generate and keep its own public key and private key. When the sender wants to produce a keyword ciphertext, he/she must take the server's public key as input in the keyword encryption algorithm. Thus, only the designed server is able to execute the testing algorithm by using its private key. After Baek *et al.*'s first construction of SCF-PEKS, a number of SCF-PEKS schemes have been proposed [6, 13, 15, 22, 24, 25, 31]. Searchable encryption was also introduced into attribute-based encryption so as to associate the keyword ciphertexts and the trapdoors with sets of attributes [20, 21, 27]. To provide multi-keyword search, Park *et al.* [23] first proposed a public key encryption with conjunctive field keyword search (PECKS) scheme. Subsequently, several PECKS schemes were proposed, *e.g.* [3, 4, 7, 17, 30]. Recently, Hwang *et al.* [16] showed that two previous PECKS schemes [7, 30] are insecure under the keyword guessing attack from the outsider attacker. To remove the requirement of secure channel in PECKS, Hwang *et al.* [16] extended the framework of PECKS and proposed a secure channel free PECKS (SCF-PECKS) scheme. They asserted that their scheme resists the keyword guessing attack and does not require conveying the trapdoors via secure channel.

In [5], Byun *et al.* first observed the keyword guessing attacks on some PEKS schemes. By performing this attack, an attacker may reveal the keyword encoded in a keyword ciphertext or a trapdoor. The keyword guessing attack exploits the low-entropy property of the commonly-used keywords. In practice, the users usually select some keywords from a small keyword space (such as an English dictionary) to generate the keyword ciphertexts and the trapdoors. Therefore, the attacker is able to traverse the keyword space to guess a correct keyword in an acceptably short time. As introduced in [5], the latest edition of *Merriam-Webster's Collegiate Dictionary* includes approximately 225 000 entries. Thus, to guess a correct keyword, the probability is about $1/2^{18}$. For another example, the current *Oxford English Dictionary* contains about 600 000 entries and thus the probability to guess the correct keyword is about $1/2^{19}$. Actually, this pro-

bability will be higher if the attack applies to some particular application, *e.g.* email application, which usually has a small keyword space containing some commonly-used keywords such as {urgent, high, normal, low, ...}. Undoubtedly, the keyword guessing attack has become the most devastating attack on the keyword search encryption schemes, since it may lead to the disclosure of the information pertaining to the encrypted data.

In this paper, we demonstrate that Hwang *et al.*'s SCF-PECKS scheme [16] is vulnerable to the keyword guessing attack. In [16], Hwang *et al.* asserted that their scheme guards against the keyword guessing attack and has the property of no secure channel. However, by giving three concrete keyword guessing attacks, we show that either the outsider attacker or the insider server is able to launch the keyword guessing attack on their scheme to disclose the keywords encoded in the encrypted data or the trapdoor. The presented attacks indicate that Hwang *et al.*'s SCF-PECKS scheme is insecure under the keyword guessing attack regardless of whether the trapdoors are sent via public channel or secure channel.

The rest of our paper is organized as follows. In Section 2, we briefly review some background preliminaries. In Section 3, we present three keyword guessing attacks on Hwang *et al.*'s SCF-PECKS scheme. Finally, we draw our conclusions in Section 4.

2. Preliminaries

2.1. Bilinear Pairing

Assume that q is a big prime number, G_1 and G_2 are two cyclic groups of the same order q. The bilinear pairing is a map e: $G_1 \times G_1 \rightarrow G_2$ which satisfies the following three attributes:

1. Bilinearity: $e(xU, yV) = e(U, V)^{xy}$ for all $U, V \in G_1$ and $x, y \in Z_q$.
2. Non-degeneracy: There exists $U, V \in G_1$ such that $e(U, V) \neq 1$.
3. Computability: There exists an efficient algorithm to compute $e(U, V)$ for all $U, V \in G_1$.

2.2 The Definition of a SCF-PECKS Scheme

A SCF-PECKS scheme consists of the following algorithms [16]:

1. **Setup(λ):** This algorithm takes a security parameter λ as input and generates a set of global parameters $\mathcal{GP}$.
2. **KenGen$_{Server}$($\mathcal{GP}$):** This algorithm takes the global parameters $\mathcal{GP}$ as input and generates a public/private key pair (pk_S, sk_S) for a server S.
3. **KeyGen$_{Receiver}$($\mathcal{GP}$):** This algorithm takes the global parameters $\mathcal{GP}$ as input and generates a public/private key pair (pk_R, sk_R) for a receiver R.
4. **dPECKS($\mathcal{GP}$, pk_S, pk_R, D):** This algorithm takes the global parameters $\mathcal{GP}$, a server's public key pk_S, a receiver's public key pk_R and a keyword set D as input and outputs a dPECKS cihpertext C.
5. **dTrapdoor($\mathcal{GP}$, pk_S, sk_R, Q):** This algorithm takes the global parameters $\mathcal{GP}$, a server's public key pk_S, a receiver's private key sk_R and a query Q as input and outputs a trapdoor $T_{w'}$.
6. **dTest($\mathcal{GP}$, sk_S, C, $T_{w'}$):** This algorithm takes the global parameters $\mathcal{GP}$, a server's private key sk_S, a dPECKS ciphertext C and a trapdoor $T_{w'}$ as input, returns "yes" if $w_{I_1} = w'_{I_1}, w_{I_2} = w'_{I_2}, ..., w_{I_t} = w'_{I_t}$ or "no" otherwise.

2.3. Adversary Types

As introduced in [5], two kinds of adversaries are considered in our keyword guessing attacks. They are outsider attacker and malicious insider server.

1. *Outsider attacker*: This adversary is able to eavesdrop on the public channel and obtain the ciphertexts and the trapdoors that are conveyed over the public channel. However, this adversary cannot perform the **dTest** algorithm, because the **dTest** algorithm requires the server's private key as input.
2. *Malicious insider server*: This adversary is able to receive the encrypted data and keyword ciphertexts from the senders. It can also receive the trapdoors from the receivers. Most importantly, it is able to execute the **dTest** algorithm to test whether a keyword ciphertext and a trapdoor correspond to a same keyword set by using its private key.

3. Keyword Guessing Attacks on Hwang *et al.*'s SCF-PECKS Scheme

In this section, we first briefly review Hwang *et al.*'s SCF-PECKS scheme [16] and then present three keyword guessing attacks on it. The presented attacks

show that Hwang *et al.*'s scheme is insecure against keyword guessing attacks by either outsider attackers or insider servers.

3.1. A Review of Hwang *et al.*'s Scheme

Hwang *et al.*'s SCF-PECKS scheme works as follows:

1. **Setup(λ):** Let G_1 be an additive group of prime order p with a generator g and G_2 be a multiplicative group of same order p. Let $e: G_1 \times G_1 \rightarrow G_2$ be a bilinear pairing and $H: \{0,1\}^* \rightarrow Z_p^*$ be a secure one-way hash function. Let KS_w denote the keyword space. This algorithm outputs the global parameters $\mathcal{GP} = \{G_1, G_2, e, g, H, KS_w\}$.
2. **KeyGen$_{Server}$($\mathcal{GP}$):** This algorithm selects a random value $\alpha \in Z_p^*$ and sets $sk_S = \alpha$. Then, it computes $pk_S = (pk_{S1}, pk_{S2}) = (e(g,g)^{\alpha}, g^{\alpha})$. Finally, it outputs the server's public/private key pair (pk_S, sk_S).
3. ***KeyGen*$_{Receiver}$($\mathcal{GP}$):** This algorithm selects a random value $\beta \in Z_p^*$ and sets $sk_R = \beta$. Then, it computes $pk_R = (pk_{R1}, pk_{R2}) = (e(g,g)^{\beta}, g^{\beta})$. Finally, it outputs the receiver's public/private key pair (pk_R, sk_R).
4. **dPECKS($\mathcal{GP}$, pk_S, pk_R, D):** In this algorithm, the sender encrypts m keywords with the server's public key $pk_S = (pk_{S1}, pk_{S2})$ and the receiver's public key $pk_R = (pk_{R1}, pk_{R2})$. This algorithm first chooses a random value $r \in Z_p^*$ and computes $C_1 = H(w_1) pk_{S1}^r$, $C_2 = H(w_2) pk_{S1}^r, \ldots, C_m = H(w_m) pk_{S1}^r$, $C_{m+1} = rg$ and $C_{m+2} = pk_{R2} \cdot pk_{S1}^r$. It then outputs the keyword ciphertext $C = (C_1, C_2, \ldots, C_m, C_{m+1}, C_{m+2})$.
5. **dTrapdoor($\mathcal{GP}$, pk_S, sk_R, Q):** In this algorithm, an authorized receiver produces a trapdoor for the keywords $w'_{I_1}, w'_{I_2}, \ldots, w'_{I_t}$ in the query Q. This algorithm first selects a random value $k \in Z_p^*$ and computes $V = \sum_{i=1}^{t} H(w'_{I_i})$. Then, it computes $T_1 = kg$ and $T_2 = (sk_R + V)^{-1} \cdot (pk_{S1})^{kt}$. Finally, it outputs the trapdoor $T_{w'} = (T_1, T_2, I_1, I_2, \ldots, I_t)$.
6. **dTest($\mathcal{GP}$, sk_S, C, $T_{w'}$):** Once receiving a trapdoor $T_{w'}$ from the receiver, the server executes this algorithm to verify whether the trapdoor $T_{w'}$ matches with the keyword ciphertext C by using its private key sk_S. This algorithm first computes $(u, \tilde{u})$ from the keyword ciphertext C as follows:

$$u = (C_{I_1} \times C_{I_2} \times \ldots \times C_{I_t}) / e(C_{m+1}, g)^{sk_S \cdot t} = \prod_{i=1}^{t} H(w_{I_i}),$$

$$\tilde{u} = (C_{m+2}) / e(C_{m+1}, g)^{sk_S} = pk_{R2}.$$

Then, it computes $Z = \dfrac{T_2}{e(T_1, g)^{sk_S \cdot t}} = (\beta + V)^{-1}$ and checks if $e(\tilde{u} \cdot g^u, g^Z) = e(g, g)$. If so, the output "yes" and "no" otherwise.

3.2. Keyword Guessing Attacks by Outsider Attackers

In Hwang *et al.*'s SCF-PECKS scheme, the keyword ciphertext is generated with the server's and the receiver's public keys while the trapdoor is generated with the server's public key and the receiver's private key. It seems that the scheme is secure against keyword guessing attacks by outsider attackers, since the execution of the **dTest** algorithm requires the input of the server's private key. However, an outsider attacker can launch a keyword guessing attack on the scheme through the following steps:

Step 1: The attacker chooses a target receiver R. Then, it uses the target receiver R's public key pk_R and the server S's public key pk_S to produce the keyword ciphertexts $C_{W_1}, C_{W_2}, \ldots, C_{W_n}$ for all candidate keyword sets W_1, W_2, ..., W_n of its choice, where $W_i = \{w_{i1}, w_{i2}, \ldots, w_{im}\}$ for each $i \in \{1, 2, \ldots, n\}$.

Step 2: The attacker simulates a sender by sending $EncDoc_1 || C_{W_1}$, $EncDoc_2 || C_{W_2}$, ..., $EncDoc_n || C_{W_n}$ to the target receiver R, where $EncDoc_1$, $EncDoc_2$, ..., $EncDoc_n$ denote the crafted encrypted documents corresponding to the keyword sets W_1, W_2, ..., W_n, respectively. After that, the encrypted data $EncDoc_1 || C_{W_1}$, $EncDoc_2 || C_{W_2}$, ..., $EncDoc_n || C_{W_n}$ will be transmitted to and stored on the server S.

Step 3: When receiving a trapdoor T_w from the receiver R, the server S returns all matched encrypted documents to the receiver R based on the result of the **dTest** algorithm. The returned documents may contain some encrypted documents faked by the attacker previously.

Step 4: Since a SCF-PECKS scheme does not assume a secure channel, the outsider attacker is able to wiretap the communication between the server S and the receiver R. Therefore, it knows which trapdoor has been sent by the receiver R. Upon observing the returned encrypted documents, including one of its crafted encrypted documents (*e.g.* $EncDoc_i$), the attacker can determine that the encoded keyword set is W_i, which implies a correct guess

In the above attack, although the outsider attacker cannot execute the **dTest** algorithm, it is able to make

use of the server as a testing oracle to get the testing results. This attack is feasible as it does not require a compromise of the server by just eavesdropping the communication between the server and the receiver.

3.3. Keyword Guessing Attacks by Insider Servers

The above attack can be avoided if the keyword trapdoors are sent to the designated server via secure channel. However, Hwang *et al.*'s SCF-PECKS scheme is still insecure against keyword guessing attacks by insider servers.

3.3.1. Attack I

A malicious insider server S can launch a keyword guessing attack on Hwang *et al.*'s scheme through the following steps:

Step 1: It obtains a valid trapdoor T_w from any receiver R.

Step 2: It guesses a candidate keyword set W' and then produces a keyword ciphertext C for the keyword set W' with the receiver R's public key pk_R and its public key pk_S.

Step 3: Using its private key sk_S, it executes the **dTest** algorithm to verify whether the keyword ciphertext C matches with the trapdoor T_w. If it does, then W' is a correct keyword set. Otherwise, it goes to Step 2 and continues its guessing.

3.3.2. Attack II

A malicious insider server S also can execute a keyword guessing attack on Hwang *et al.*'s scheme through the following steps:

Step 1: It obtains a valid trapdoor $T_w = (T_1, T_2, I_1, I_2, ..., I_t)$ from any receiver R;

Step 2: It guesses a candidate keyword set $W' = \{w'_{I_1}, w'_{I_2}, ..., w'_{I_t}\}$ and then computes the hash value $H_3(w'_{I_i})$ of the each keyword w'_{I_i}. After that, it calculates $V^* = \prod_{i=1}^{t} H(w'_{I_i})$;

Step 3: Using its private key sk_S and the value V^*, it verifies whether the trapdoor $T_w = (T_1, T_2, I_1, I_2, ..., I_t)$ satisfies the equation $g^{\left(\frac{T_2}{e(T_1,g)^{sk_S \cdot t}}\right)^{-1}} = pk_{R2} \cdot g^{V^*}$. If the equation holds, then W' is a correct keyword set. Otherwise, it goes to Step 2 and continues its guessing.

Obviously, an insider server is able to reveal the keyword set encoded in a trapdoor by the above attack. What is worse, is that after guessing the correct keyword set, the insider server can further run the **dTest** algorithm to determine which encrypted document contains the keyword set.

4. Conclusions

In this paper, we have shown that Hwang *et al.*'s SCF-PECKS scheme [16] is insecure against keyword guessing attacks. Our first attack on Hwang *et al.*'s scheme indicates that an outsider attacker is able to make use of the server as a testing oracle to verify the correctness of its guesses. This attack is feasible due to the communication between the receiver and server is over a public channel. Therefore, a secure channel is still required for a SCF-PECKS scheme to guard against the keyword guessing attacks by outsider attackers. However, our other two attacks show that this is still not enough to protect a SCF-PECKS scheme from the keyword guessing attacks by insider servers. It seems that it is impossible to devise a scheme against keyword guessing attacks under the current framework of SCF-PECKS. Therefore, it is necessary to design a new framework for SCF-PECKS.

Acknowledgments

We would like to present our thanks to the anonymous reviewers for their helpful comments. This work is supported by the National Natural Science Foundation of China under Grant Nos. 61772009 and U1736112, the Natural Science Foundation of Jiangsu Province under Grant No. BK20161511, the Fundamental Research Funds for the Central Universities under Grant Nos. 2016B10114 and 2017B17014.

References

1. Baek, J., Safavi-Naini, R., Susilo, W. Public Key Encryption with Keyword Search Revisited. Proceedings of 2008 International Conference on Computational Science and Its Applications (ICCSA 2008), Perugia, Italy, June 30 – July 3, 2008, 1249-1259. https://doi.org/10.1007/978-3-540-69839-5_96.

2. Boneh, D., Crescenzo, G., Ostrovsky, R., Rersiano, G. Public Key Encryption with Keyword Search. Proceedings of 23rd International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2004), Interlaken, Switzerland, May 2-6, 2004, 506-522. https://doi.org/10.1007/978-3-540-24676-3_30

3. Boneh, D., Waters, B. Conjunctive, Subset, and Range Queries on Encrypted Data. Proceedings of 4th Theory of Cryptography Conference (TCC 2007), Amsterdam, Netherlands, February 21-24, 2007, 535-554. https://doi.org/10.1007/978-3-540-70936-7_29

4. Byun, J., Lee, D., Lim, J. Efficient Conjunctive Keyword Search on Encrypted Data Storage System. Proceedings of 3rd European Public Key Infrastructure Workshop: Theory and Practice (EUROPKI 2006), Turin, Italy, June 19-20, 2006, 184-196. https://doi.org/10.1007/11774716_15

5. Byun, J., Rhee, H., Park, H., Lee, D. Off-Line Keyword Guessing Attacks on Recent Keyword Search Schemes over Encrypted Data. Proceedings of 3rd VLDB Workshop on Secure Data Management, (SDM2006), Seoul, Korea, September 10-11, 2006, 75-83. https://doi.org/10.1007/11844662_6

6. Chen, Y. SPEKS: Secure Server-Designation Public Key Encryption with Keyword Search Against Keyword Guessing Attacks. The Computer Journal, 2015, 58(4), 922-933. https://doi.org/10.1093/comjnl/bxu013

7. Chen, Y., Horng, G. Timestamped Conjunctive Keyword-Searchable Public Key Encryption. Proceedings of 4th International Conference on Innovation Computing Information and Control, (ICICIC2009), Kaohsiung, Taiwan, December 7-9, 2009, 729-732. https://doi.org/10.1109/ICICIC.2009.369

8. Chen, R., Mu, Y., Yang, G., Guo, F., Wang, X. A New General Framework for Secure Public Key Encryption with Keyword Search. Proceedings of 20th Australasian Conference on Information Security and Privacy (ACISP 2015), Brisbane, QLD, Australia, June 29 – July 1, 2015, 59-76. https://doi.org/10.1007/978-3-319-19962-7_4

9. Fu, Z., Ren, K., Shu, J., Sun, X., Huang, F. Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(9), 2546-2559. https://doi.org/10.1109/TPDS.2015.2506573

10. Fu, Z., Sun, X., Liu, Q., Zhou, L., Shu, J. Achieving Efficient Cloud Search Services: Multi-Keyword Ranked Search over Encrypted Cloud Data Supporting Parallel Computing. IEICE Transactions on Communications, 2015, 98(1), 190-200. https://doi.org/10.1587/transcom.E98.B.190

11. Fu, Z., Wu, X., Guan, C., Sun, X., Ren, K. Towards Efficient Multi-Keyword Fuzzy Search over Encrypted Outsourced Data with Accuracy Improvement. IEEE Transactions on Information Forensics and Security, 2016, 11(12), 2706-2716. https://doi.org/10.1109/TIFS.2016.2596138

12. Gu, C., Zhu, Y., Pan, H. Efficient Public Key Encryption with Keyword Search Schemes from Pairings. Proceedings of 3rd SKLOIS Conference on Information Security and Cryptology (INSCRYPT2007), Xining, China, August 31 – September 5, 2007, 372-383. https://doi.org/10.1007/978-3-540-79499-8_29

13. Guo, L., Yau, W. Efficient Secure-Channel Free Public Key Encryption with Keyword Search for EMRs in Cloud Storage. Journal of Medical Systems, 2015, 39(2), 1-11. https://doi.org/10.1007/s10916-014-0178-y

14. Hu, C., He, P., Liu, P. Public Key Encryption with Multi-Keyword Search. Proceedings of 2nd International Conference on Network Computing and Information Security (NCIS 2012), Shanghai, China, December 7-9, 2012, 568-576. https://doi.org/10.1007/978-3-642-35211-9_72

15. Hu, C., Liu, P. An Enhanced Searchable Public Key Encryption Scheme with a Designated Tester and Its Extensions. Journal of Computers, 2012, 7(3), 131-136. https://doi.org/10.4304/jcp.7.3.716-723

16. Hwang, M., Hsu, S., Lee, C. A New Public Key Encryption with Conjunctive Field Keyword Search Scheme. Information Technology and Control, 2014, 43(3), 277-288. https://doi.org/10.5755/j01.itc.43.3.6429

17. Hwang, Y., Lee, P. Public Key Encryption with Conjunctive Keyword Search and Its Extension to a Multi-User System. Proceedings of 1st International Conference on Pairing-Based Cryptography (Pairing 2007), Tokyo, Japan, July 2-4, 2007, 2-22. https://doi.org/10.1007/978-3-540-73489-5_2

18. Jiang, P., Mu, Y., Guo, F., Wen, Q. Public Key Encryption with Authorized Keyword Search. Proceedings of 21th Australasian Conference on Information Security and Privacy (ACISP 2016), Melbourne, VIC, Australia, July 4-6, 2016, 170-186. https://doi.org/10.1007/978-3-319-40367-0_11

19. Khader, D. Public Key Encryption with Keyword Search Based on K-Resilient IBE. Proceedings of 2006 International Conference on Computational Science and Its Applications (ICCSA 2006), Glasgow, UK, May 8-11, 2006, 298-308. https://doi.org/10.1007/11751595_33

20. Li, J., Lin, X., Zhang, Y., Han, J. KSF-OABE: Outsourced Attribute-Based Encryption with Keyword Search Function for Cloud Storage. IEEE Transactions on

Services Computing, 2016, 10(5), 715-725. https://doi.org/10.1109/TSC.2016.2542813.

21. Li, J., Shi, Y., Zhang, Y. Searchable Ciphertext-Policy Attribute-Based Encryption with Revocation in Cloud Storage. International Journal of Communication Systems, 2015, 30(1), e2942. https://doi.org/10.1002/dac.2942.
22. Lu, Y., Wang, G., Li, J., Shen, J. Efficient Designated Server Identity-Based Encryption with Conjunctive Keywords Search. Annals of Telecommunications, 2017, 72(5-6), 359-370. https://doi.org/10.1007/s12243-017-0574-7
23. Park, D., Kim, K., Lee, P. Public Key Encryption with Conjunctive Field Keyword Search. Proceedings of 5th International Workshop on Information Security Applications (WISA 2004), Jeju Island, Korea, August 23-25, 2004, 73-86. https://doi.org/10.1007/978-3-540-31815-6_7
24. Rhee, H.-S., Park, J., Susilo, W., Dong, H. Trapdoor Security in a Searchable Public Key Encryption Scheme with a Designated Tester. Journal of Systems and Software, 2010, 83(5), 763-771. https://doi.org/10.1016/j.jss.2009.11.726
25. Rhee, H., Park, J.-H., Susilo, W., Lee, D.-H. Improved Searchable Public-Key Encryption with Designated Tester. Proceedings of 4th International Symposium on Information, Computer, and Communications Security (ASIACCS 2009), Sydney, Australia, March 10-12, 2009, 376-379. https://doi.org/10.1145/1533057.1533108
26. Song, D., Wagner, D., Perrig, A. Practical Techniques for Searching on Encrypted Data. Proceedings of the 31st IEEE Symposium on Security and Privacy (S&P 2000), Oakland, California, USA, May 16-19, 2000, 44-55. https://doi.org/10.1109/SECPRI.2000.848445
27. Sun, W., Yu, S., Lou, W., Hou, Y., Li, H. Protecting Your Right: Attribute-Based Keyword Search with Fine-Grained Owner-Enforced Search Authorization in the Cloud. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(4), 1187-1198. https://doi.org/10.1109/TPDS.2014.2355202
28. Vallent, T., Kim, H. A Pairing Free Public Key Encryption with Keyword Searching for Cloud Storage Services. Proceedings of 5th International Conference on e-Infrastructure and e-Services for Developing Countries (AFRICOMM 2013), Blantyre, Malawi, November 25-27, 2013, 70-78. https://doi.org/10.1007/978-3-319-08368-1_8
29. Xia, Z., Wang, X., Sun, X., Wang, Q. A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data. IEEE Transactions on Parallel and Distributed Systems, 2015, 27(2), 340-352. https://doi.org/10.1109/TPDS.2015.2401003
30. Zhang, B., Zhang, F. An Efficient Public Key Encryption with Conjunctive-Subset Keywords Search. Journal of Network and Computer Applications, 2011, 34(1), 262-267. https://doi.org/10.1016/j.jnca.2010.07.007
31. Zhou, Y., Xu, G. Wang, Y., Wang, X. Chaotic Map-Based Time-Aware Multi-Keyword Search Scheme with Designated Server. Wireless Communications and Mobile Computing, 2016, 16(3), 1851-1858. https://doi.org/10.1002/wcm.2656