Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **1**, January **2018**

Comments on Recent Proposed Cui Et Al.'s KASE and Lu Et Al.'s dIBEKS Schemes

Tsu-Yang Wu[1,2], Chao Meng[3], King-Hang Wang[4]
Chien-Ming Chen[3*], and Jeng-Shyang Pan[1,2]

[1]Fujian Provincial Key Laboratory of Big Data Mining and Applications
[2]National Demonstration Center for
Experimental Electronic Information and Electrical Technology Education
Fujian University of Technology
No3 Xueyuan Road, University Town, Fuzhou 350118, China
wutsuyang@gmail.com; jengshyangpan@fjut.edu.cn

[3]School of Computer Science and Technology
Harbin Institute of Technology Shenzhen Graduate School
HIT Campus of University Town of Shenzhen, Shenzhen 518055, China
171521532@qq.com; chienming.taiwan@gmail.com

[4]Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
kevinw@cse.ust.hk
[*]Corresponding author's email: chienming.taiwan@gmail.com

Received July, 2017; revised October, 2017

Abstract. *Searchable encryption is a cryptographic primitive used to search an encrypted data in cloud storage. Recently, Cui et al. and Lu et al. proposed two variants of secure searchable encryption schemes, respectively. However, based on our best knowledge we demonstrate that the both schemes are insecure against different types of off-line keyword guessing attacks in this paper. Finally, we make discussions about searchable encryption schemes whether resisting off-line keyword guessing attacks.*
Keywords: Searchable encryption, Public key encryption with keyword search, Designated server, Off-line keyword guessing attack, Cryptanalysis

1. **Introduction.** With the fast growth of cloud technologies, cloud storage [1, 2, 3] provides convenient, ubiquitous, on-demand access to huge amount of data shared over the Internet. Nowadays, people popularly upload their personal data such as photo and video or share these data with their friends via social network applications based on cloud storage. However, data leakage, a serious security risk in cloud storage, had been occurred, for example celebrity photos being leaked in iCloud. The data leakage problem is caused by a malicious attacker or a misbehaving cloud operator in cloud storage. To address this problem, one common approach is that data owner must encrypt their data before uploading to the cloud. Searchable encryption (SE) [4, 5, 6, 7] is a cryptographic primitive which can be used to solve how to retrieve an encrypted data stored in the cloud. In the SE scheme, data owner is required to encrypt potential keywords related to data and upload them with encrypted data to the cloud. To retrieve the encrypted data,

user sends a trapdoor generated by a chosen keyword to the cloud. Finally, the cloud performs the search functionality over the encrypted data.

Public key encryption with keyword search (PEKS) (or called searchable public key encryption) is a variant of searchable encryption and was first introduced by Boneh et al. [8] in 2004. The PEKS scheme proposed by Boneh et al. described a framework depicted in Figure 1 to address how to search an encrypted data stored in the cloud problem. It describes three roles: a server, a data owner, and a data user, who can be the data owner himself or any other designated individual who has the right of accessing the data. The data owner first encrypted the keywords with the data user's public key and uploaded to the server together with the encrypted data files. When a data user wish to retrieve document with a particular keyword, she/he will generate a trapdoor using her/his private key and the keyword. This trapdoor is securely sent to the server. The server can test an encrypted keyword ciphertext matching with the trapdoor using some mathematical equations. The matching encrypted data will then sent to the user. Such framework was used in the subsequent works [9, 10].

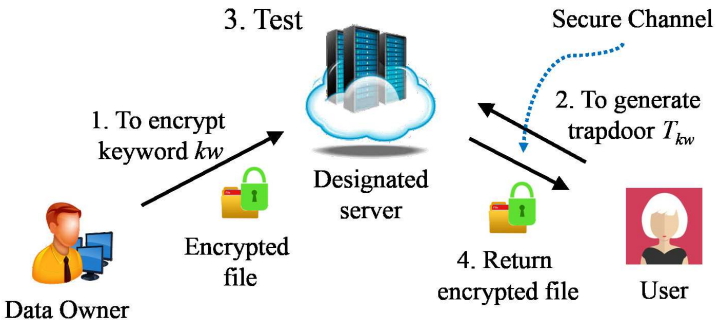

FIGURE 1. The framework of a PEKS scheme

In order to remove the required secure channel in Boneh et al.'s model, Baek et al. [11] redefined a new model and proposed a new scheme called PEKS with designated server (dPEKS). However, Rhee et al. [12] pointed out that the security model of Baek et al. seriously limits the ability of the adversary. They enhanced the security model of the dPEKS scheme. Further, Rhee et al. [13] defined a new security notion of dPEKS called "Trapdoor indistinguishability" which allows a scheme to be formally proven secure against a non-designated person who wants to launch an off-line keyword guessing attack. Note that in the kinds of attacks attacker including malicious server can simply enumerate on all possible keywords to test generated trapdoor or encrypted keyword ciphertext [14, 15, 16, 17, 18, 19]. After that, several dPEKS schemes based on different public key cryptosystems were proposed such as identity (ID)-based public key cryptosystems [20, 19] and certificateless public key cryptosystems [21, 22, 23, 24, 25].

Recently, Cui et al. [26] combined the concepts of searchable encryption [7] and aggregate encryption [27] to propose a key-aggregate searchable encryption (KASE) scheme. Lu et al. [19] pointed out Wu et al.'s dIBEKS scheme [20] did not provide ciphertext indistinguishability. To enhance the security weakness and to provide supporting multi-keyword search functionality, they proposed a designated server identity-based encryption scheme with conjugate keyword search called dIBECKS. In this paper, we demonstrate that both KASE and dIBECKS schemes are suffered from different types of off-line keyword guessing attacks. In Cui et al.'s KASE scheme, we point out that outside attacker and malicious cloud server can launch off-line keyword guessing attacks to trapdoor. Further, we point out that malicious designated server can launch off-line keyword guessing

attacks to ciphertext and trapdoor in Lu et al.'s dIBECKS scheme. Finally, we make discussions for PEKS and dPEKS schemes whether resisting off-line keyword guessing attacks.

2. Cryptanalysis of Cui et al.'s KASE Scheme.

2.1. Review of Cui et al.'s KASE scheme.

In order to solve the problem that how to search encrypted files by users with different encryption keys, Cui et al. [26] combined the concepts of searchable encryption and aggregate encryption to propose a key-aggregate searchable encryption (KASE) scheme as depicted in Figure 2. In their scheme, there are three roles: data owner, user, and cloud, where data owner needs to distribute a single aggregate key to user for shareing files, user needs to send a single trapdoor to the cloud for searching the shared files, and cloud performs test procedures to search encrypted files using the trapdoor. Cui et al.'s KASE scheme consists of seven algorithm:

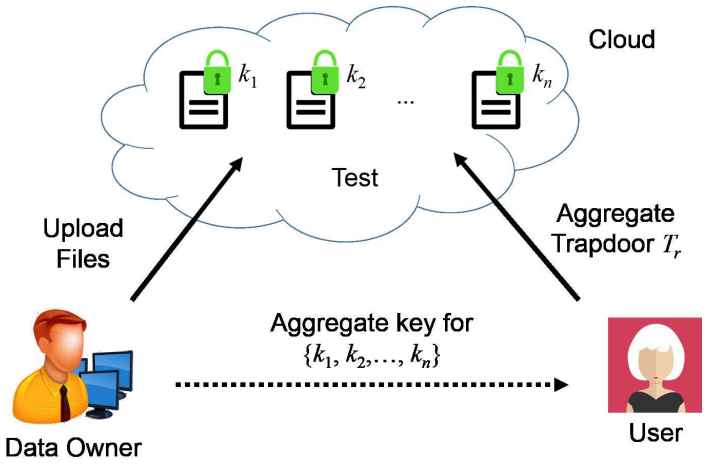

FIGURE 2. The framework of Cui et al.'s KASE scheme

1. *Setup.* The cloud server adopts this algorithm to generate system parameters as follows.
   (a) Generating a bilinear map group system $\mathcal{B} = \{e, \mathcal{G}, \mathcal{G}_1, p\}$, where $e : \mathcal{G} \times \mathcal{G} \to \mathcal{G}_1$, $2^\lambda \leq p \leq 2^{\lambda+1}$ is the order of $\mathcal{G}$, and λ is a security parameter.
   (b) Setting n as the maximum number of files belonging to data owner.
   (c) Selecting a generator $g \in_R \mathcal{G}$ and $\alpha \in_R \mathbb{Z}_p$. Then, computing $g_i = g^{(\alpha^i)}$ for $i = 1, 2, \ldots, 2n$.
   (d) Choosing a cryptographic hash function $H : \{0,1\}^* \to \mathcal{G}$.
   Finally, the cloud publishes public parameters $param = \{\mathcal{B}, g, g_1, g_2, \ldots, g_{2n}, H\}$. Note that for the details of bilinear maps, readers can refer to [28, 29, 30, 31, 32, 33, 34] for a full descriptions.
2. *Key generation.* Data owner adopts this algorithm to generate her/his private/public key pair (γ, v), where $\gamma \in_R \mathbb{Z}_p$ and $v = g^\gamma$.
3. *Keyword encryption.* Data owner adopts this algorithm to encrypt file's keyword as follows. Inputting the index of file $i \in \{1, 2, \ldots, n\}$.
   (a) Selecting $t \in_R \mathbb{Z}_p$ as encryption key k_i of file.
   (b) Defining $\Delta_i = (C_1, C_2)$ for k_i, where $C_1 = g^t$ and $C_2 = (v \cdot g_i)^t$.
   (c) For keyword w, computing $C_w = e(g, H(w))^t / e(g_1, g_n)^t$.
   Note that C_1 and C_2 are public and C_w is stored in the cloud.

4. *Aggregate key generation.* Data owner adopts this algorithm to generate an aggregate searchable encryption key k_{agg}. Given a subset $S \subseteq \{1, 2, \ldots, n\}$, k_{agg} is computed by

$$k_{agg} = \prod_{j \in S} g_{n+1-j}^{\gamma}.$$

   To delegate the keyword search right to a user, data owner sends k_{agg} and S to the user via a secure channel.
5. *Trapdoor generation.* User adopts this algorithm to generate a trapdoor T_r which is used to perform keyword search. Assume that all searched files are relevant to the aggregate key k_{agg}. The trapdoor T_r for keyword w is computed by

$$T_r = k_{agg} \cdot H(w).$$

   Then, the user sends T_r and S to the cloud.
6. *Adjust.* The cloud adopts this algorithm to generate a right trapdoor T_{r_i} for the file with index $i \in S$. T_{r_i} is computed by

$$T_{r_i} = T_r \cdot \prod_{j \in S, j \neq i} g_{n+1-j+i}.$$

7. *Test.* The cloud server adopts this algorithm to perform keyword search for the file with index i. The cloud only verifies

$$C_w \stackrel{?}{=} e(T_{r_i}, C_1)/e(pub, C_2),$$

   where $pub = \prod_{j \in S} g_{n+1-j}$.

2.2. **The proposed attacks.** In this subsection, we demonstrate that Cui et al.'s KASE scheme is insecure against off-line keyword guessing attacks to trapdoor by outside attacker and malicious cloud server.

2.2.1. *Keyword guessing attack by outside attacker.* Assume that a outside adversary $\mathcal{A}$ intercepts (T_r, S) sent by user. Then, $\mathcal{A}$ can launch a off-line keyword guessing attack as follows.

1. Computing $pub = \prod_{j \in S} g_{n+1-j}$.
2. Guessing an appropriate keyword w'.
3. To verify

$$e(v, pub) \cdot e(g, H(w')) \stackrel{?}{=} e(g, T_r),$$

   where $v = g^{\gamma}$ is the data owner's public key.

If the verification is true, it means that the guessed keyword w' is related to the trapdoor T_r, the attack success. Otherwise, $\mathcal{A}$ goes back to the step 2 and continues to execute the step 3.

Here, we provide the correctness of our first attack. Assume that the keyword w is the success guessed keyword. Then,

$$e(g^{\gamma}, pub) \cdot e(g, H(w)) = e(g, \prod_{j \in S} g_{n+1-j})^{\gamma} \cdot e(g, H(w)) = e(g, \prod_{j \in S} g_{n+1-j}^{\gamma} \cdot H(w)) = e(g, T_r).$$

2.2.2. *Keyword guessing attack by malicious cloud server.* Note that the malicious cloud server (MS) can also launch a off-line keyword guessing attack for (T_r, S) sent by user in the Adjust phase. The similar attack method is mentioned in Subsection 2.2.1. Here, we demonstrate that the MS also launch another off-line keyword guessing attack for (T_{r_i}, S) in the Test phase as follows.

1. Computing $pub = \prod_{j \in S} g_{n+1-j}$.
2. Guessing an appropriate keyword w'.
3. To verify
$$e(g, H(w'))/e(g_1, g_n) \stackrel{?}{=} e(T_{r_i}, g)/e(pub, v \cdot g_i),$$
where $v = g^{\gamma}$ is the data owner's public key.

If the verification is true, it means that the guessed keyword w' is related to the trapdoor T_{r_i}, the attack success. Otherwise, the MS goes back to the step 2 and continues to execute the step 3.

Here, we provide the correctness of our second attack. Assume the keyword w is the success guessed keyword. By the verification in the Test phase, we have

$$e(g, H(w'))^t/e(g_1, g_n)^t = e(T_{r_i}, g^t)/e(pub, (v \cdot g_i)^t),$$
where $v = g^{\gamma}$ is the data owner's public key. Then, it implies that
$$e(g, H(w'))/e(g_1, g_n) = e(T_{r_i}, g)/e(pub, v \cdot g_i).$$

3. Cryptanalysis of Lu et al.'s dIBECKS Scheme.

3.1. Review of Lu et al.'s dIBECKS Scheme.

Recently, Lu et al. [19] pointed out Wu et al.'s dIBEKS scheme [20] did not provide ciphertext indistinguishability. To enhance the security weakness and to provide supporting multi-keyword search functionality, they proposed a designated server identity-based encryption scheme with conjugate keyword search called dIBECKS. The dIBECKS scheme consists of seven algorithms described as follows.

1. *PKG Setup.* Inputting a security parameter k, the PKG selects a bilinear map $e : G_1 \times G_1 \to G_2$, where G_1 and G_2 are two cyclic groups with a same prime order q. Then, the PKG chooses $s \in_R \mathbb{Z}_q^*$ as master key and corresponding public key P_{pub} is computed by $s \cdot P$, where P is a generator of G_1. Four cryptographic hash functions are selected $H_1 : \{0,1\}^* \to G_1$, $H_2 : \{0,1\}^* \to G_1$, $H_3 : \{0,1\}^* \to G_1$, and $H_4 : G_2 \to \mathbb{Z}_q^*$. Finally, the PKG publishes the public parameters $param = \{e, G_1, G_2, q, P, P_{pub}, H_1, H_2, H_3, H_4\}$.
2. *Server Key Extract.* Inputting a server's identity $ID_S \in \{0,1\}^*$, the PKG computes server's secret key $d_S = s \cdot Q_S$, where $Q_S = H_1(ID_S)$.
3. *Server Setup.* Given the secret key d_S, the server with ID_S selects $x \in_R \mathbb{Z}_q^*$ and then sets its private key $SK_S = (SK_{S1}, SK_{S2}) = (d_S, x)$. The server's public key PK_S is computed by $x \cdot P$.
4. *User Key Extract.* Inputting user's identity $ID_U \in \{0,1\}^*$, the PKG computes user's private key SK_U by $s \cdot Q_U$, where $Q_U = H_2(ID_U)$.
5. *Keyword Set Encryption.* To encrypt a keyword set $W = \{w_1, w_2, \ldots, w_n\}$, sender selects r_1, $r_2 \in_R \mathbb{Z}_q^*$ and computes
   (a) $C_1 = r_1 \cdot r_2 \cdot H_1(ID_S)$,
   (b) $C_{2i} = r_1 \cdot H_3(w_i)$ for $i = 1, 2, \ldots, n$,
   (c) $C_3 = r_2 \cdot PK_S$,
   (d) $C_4 = r_1 \cdot r_2 \cdot P$,
   (e) $C_5 = H_4(e(H_1(ID_S) + H_2(ID_U), r_1 \cdot r_2 \cdot P_{pub}))$.

The ciphertext of keyword set W is defined by $C_W = (C_1, C_{21}, C_{22}, \ldots, C_{2n}, C_3, C_4, C_5)$.

6. *Trapdoor Generation.* To generate a trapdoor of a selected keyword set $W_T = \{w_{I_1}, w_{I_2}, \ldots, w_{I_l}\}$ with indices $L_I = \{I_1, I_2, \ldots, I_l\}$, receiver with identity ID_R chooses $t \in_R \mathbb{Z}_q^*$ and computes
   (a) $T_1 = t \cdot PK_S$,
   (b) $T_2 = H_4(e(t \cdot H_1(ID_S), P_{pub}))$,
   (c) $T_3 = SK_U - \sum_{i=1}^{l} H_3(w_{I_i}) - t \cdot T_2 \cdot H_1(ID_S)$.
   The trapdoor of keyword set W_T is defined by $T_{W_T} = (T_1, T_2, T_3, L_I)$.
7. *Test.* Given a cithertext $C_W = (C_1, C_{21}, C_{22}, \ldots, C_{2n}, C_3, C_4, C_5)$ sent by sender and a trapdoor $T_{W_T} = (T_1, T_2, T_3, L_I)$ sent by receiver, the designated server first computes $C_2^* = r_1 \cdot \sum_{i=1}^{l} H_3(w_{I_i})$ and then verifies
$$C_5 \stackrel{?}{=} H_4(e(SK_{S1} + T_3, C_4) \cdot e(C_2^*, SK_{S_2}^{-1} \cdot C_3) \cdot e(C_1, SK_{S2}^{-1} \cdot T_2 \cdot T_1)).$$
   If the verification holds, the server returns 1 meaning that $W_T \subseteq W$. Otherwise, returning 0.

3.2. **The proposed attacks.** Though Lu et al. demonstrated that their dIBECKS Scheme achieves ciphertext indistinguishability, trapdoor indistinguishability, and resisting off-line keyword guessing attack, we point out that their dIBECKS Scheme is insecure against off-line keyword guessing attacks to ciphertext and trapdoor by a malicious designated server MS.

3.2.1. *Keyword guessing attack to ciphertext.* Assume that the malicious designated server MS receives a ciphertext $C_W = (C_1, C_{21}, C_{22}, \ldots, C_{2n}, C_3, C_4, C_5)$. Then, MS can launch an off-line keyword guessing attack on C_W described as follows.

1. Guessing an appropriate keyword w_i',
2. To verify
$$e(C_{2i}, C_3) \stackrel{?}{=} e(SK_{S2} \cdot H_3(w_i'), C_4).$$

If the verification is true, it means that the guessed keyword w_i' is related to the ciphertext C_W, the attack success. Otherwise, the MS goes back to the step 1 and continues to execute the step 2.

Here, we provide the correctness of our first attack. Assume that the keyword w_i is the success guessed keyword. Then,
$$e(C_{2i}, C_3) = e(r_1 \cdot H_3(w_i), r_2 \cdot x \cdot P) = e(x \cdot H_3(w_i), r_1 \cdot r_2 \cdot P) = e(SK_{S2} \cdot H_3(w'), C_4).$$

3.2.2. *Keyword guessing attack to trapdoor.* Assume that the malicious designated server MS receives a trapdoor $T_{W_T} = (T_1, T_2, T_3, L_I)$. Then, MS can launch an off-line keyword guessing attack on T_{W_T} described as follows.

1. Guessing an appropriate keyword set $W_T' = \{w_{I_1}', w_{I_2}', \ldots, w_{I_l}'\}$,
2. Computing $X = \sum_{i=1}^{l} H_3(w_{I_i}')$,
3. To verify
$$e(X + T_3, P) \cdot e(T_2 \cdot H_1(ID_S), x^{-1} \cdot T_1) \stackrel{?}{=} e(H_2(ID_U), P_{pub}).$$

If the verification is true, it means that the guessed keyword set W_T' is related to the trapdoor T_{W_T}, the attack success. Otherwise, the MS goes back to the step 1 and continues to execute the steps 2-3.

Here, we provide the correctness of our first attack. Assume that the keyword set W_T is the success guessed keyword set. Then,
$e(X + T_3, P) \cdot e(T_2 \cdot H_1(ID_S), t \cdot P) = e(T_3 + X + t \cdot T_2 \cdot H_1(ID_S), P) =$
$e(SK_U, P) = e(H_2(ID_U), P_{pub})$.

4. **Conclusions and Discussions.** In this paper, we have demonstrated that Cui et al.'s KASE and Lu et al.'s dIBEKS schemes suffered from different types of off-line keyword guessing attacks to ciphertext and trapdoor, respectively. In 2009, Jeong et al. [35] proved that to construct a secure PEKS scheme against off-line keyword guessing attacks is impossible, while the number of keywords is bounded by some polynomial. In other words, any PEKS acheme based on Bonech et al.'s framework [8] is insecure against off-line keyword guessing attacks. Later, Baek et al. [11] and Rhee et al. [12, 13] formalized and enhanced the security model of dPEKS such that to resist off-line keyword guessing attacks becomes possible. Several dPEKS scheme based on their security models were proposed in [20, 21, 22, 23, 24, 25]. However, Wu et al.'s scheme [20] is insecure against off-line keyword guessing attacks to cithertext demonstrated by Lu et al. in [19]. We have pointed out a malicious server in Lu et al.'s dIBEKS scheme can launch off-line keyword guessing attacks on ciphertext and trapdoor. Hence, a dPEKS scheme is secure against off-line keyword guessing attacks is possible or impossible? The existed security model of dPEKS is needed enhanced? I think it remains open problems.

Acknowledgment. The authors thank the referrers for the helpful comments and suggestions. This paper was supported in part by the Project NSFC (National Natural Science Foundation of China) under Grant number 61402135 and in part by Shenzhen Technical Project under Grant number JCYJ20170307151750788.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, *et al.*, A view of cloud computing, *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] T.-Y. Wu, C.-M. Chen, X. Sun, S. Liu, and J. C.-W. Lin, A countermeasure to sql injection attack for cloud environment, *Wireless Personal Communications*, vol. 96, no. 4, pp. 5279–5293, 2017.

[3] T.-Y. Wu, Y.-M. Tseng, S.-S. Huang, and Y.-C. Lai, Non-repudiable provable data possession scheme with designated verifier in cloud storage systems, *IEEE Access*, vol. 5, pp. 19333–19341, 2017.

[4] D. X. Song, D. Wagner, and A. Perrig, Practical techniques for searches on encrypted data, in *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*, pp. 44–55, IEEE, 2000.

[5] P. Golle, J. Staddon, and B. Waters, Secure conjunctive keyword search over encrypted data, in *ACNS*, vol. 4, pp. 31–45, Springer, 2004.

[6] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, Searchable symmetric encryption: improved definitions and efficient constructions, *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.

[7] R. A. Popa and N. Zeldovich, Multi-key searchable encryption., *IACR Cryptology ePrint Archive*, vol. 2013, p. 508, 2013.

[8] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, Public key encryption with keyword search, in *Advances in Cryptology-Eurocrypt 2004*, pp. 506–522, Springer, 2004.

[9] D. J. Park, K. Kim, and P. J. Lee, Public key encryption with conjunctive field keyword search, in *International Workshop on Information Security Applications*, pp. 73–86, Springer, 2004.

[10] Y. Hwang and P. Lee, Public key encryption with conjunctive keyword search and its extension to a multi-user system, *Pairing-Based Cryptography–Pairing 2007*, pp. 2–22, 2007.

[11] J. Baek, R. Safavi-Naini, and W. Susilo, Public key encryption with keyword search revisited, *Computational science and its applications–ICCSA 2008*, pp. 1249–1259, 2008.

[12] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, Improved searchable public key encryption with designated tester, in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pp. 376–379, ACM, 2009.

[13] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, Trapdoor security in a searchable public-key encryption scheme with a designated tester, *Journal of Systems and Software*, vol. 83, no. 5, pp. 763–771, 2010.

[14] J. W. Byun, H. S. Rhee, H.-A. Park, and D. H. Lee, Off-line keyword guessing attacks on recent keyword search schemes over encrypted data, in *Workshop on Secure Data Management*, pp. 75–83, Springer, 2006.

[15] H. S. Rhee, W. Susilo, and H.-J. Kim, Secure searchable public key encryption scheme against keyword guessing attacks, *IEICE Electronics Express*, vol. 6, no. 5, pp. 237–243, 2009.

[16] B. Wang, T. Chen, and F. Jeng, Security improvement against malicious server's attackfor a dpeks scheme, *International Journal of Information and Education Technology*, vol. 1, no. 4, p. 350, 2011.

[17] C. Hu and P. Liu, An enhanced searchable public key encryption scheme with a designated tester and its extensions, *J. Comput*, vol. 7, no. 3, pp. 716–723, 2012.

[18] W.-C. Yau, R. C.-W. Phan, S.-H. Heng, and B.-M. Goi, Keyword guessing attacks on secure searchable public key encryption schemes with a designated tester, *International Journal of Computer Mathematics*, vol. 90, no. 12, pp. 2581–2587, 2013.

[19] Y. Lu, G. Wang, J. Li, and J. Shen, Efficient designated server identity-based encryption with conjunctive keyword search, *Annals of Telecommunications*, vol. 72, no. 5-6, pp. 359–370, 2017.

[20] T.-Y. Wu, T.-T. Tsai, and Y.-M. Tseng, Efficient searchable id-based encryption with a designated server, *annals of telecommunications-annales des télécommunications*, vol. 69, no. 7-8, pp. 391–402, 2014.

[21] Y. Peng, J. Cui, P. Changgen, and Z. Ying, Certificateless public key encryption with keyword search, *Communications, China*, vol. 11, no. 11, pp. 100–113, 2014.

[22] Q. Zheng, X. Li, and A. Azgin, Clks: Certificateless keyword search on encrypted data, in *International Conference on Network and System Security*, pp. 239–253, Springer, 2015.

[23] S. H. Islam, M. S. Obaidat, V. Rajeev, and R. Amin, Design of a certificateless designated server based searchable public key encryption scheme, in *International Conference on Mathematics and Computing*, pp. 3–15, Springer, 2017.

[24] M. Ma, D. He, N. Kumar, K.-K. R. Choo, and J. Chen, Certificateless searchable public key encryption scheme for industrial internet of things, *IEEE Transactions on Industrial Informatics*, p. DOI: 10.1109/TII.2017.2703922, 2017.

[25] M. Ma, D. He, M. K. Khan, and J. Chen, Certificateless searchable public key encryption scheme for mobile healthcare system, *Computers & Electrical Engineering*, p. https://doi.org/10.1016/j.compeleceng.2017.05.014, 2017.

[26] B. Cui, Z. Liu, and L. Wang, Key-aggregate searchable encryption (kase) for group data sharing via cloud storage, *IEEE Transactions on computers*, vol. 65, no. 8, pp. 2374–2385, 2016.

[27] C.-K. Chu, S. S. Chow, W.-G. Tzeng, J. Zhou, and R. H. Deng, Key-aggregate cryptosystem for scalable data sharing in cloud storage, *IEEE transactions on parallel and distributed systems*, vol. 25, no. 2, pp. 468–477, 2014.

[28] D. Boneh and M. Franklin, Identity-based encryption from the weil pairing, in *Annual International Cryptology Conference*, pp. 213–229, Springer, 2001.

[29] T.-Y. Wu and Y.-M. Tseng, An id-based mutual authentication and key exchange protocol for low-power mobile devices, *The Computer Journal*, vol. 53, no. 7, pp. 1062–1070, 2010.

[30] C.-M. Chen, C.-T. Li, S. Liu, T.-Y. Wu, and J.-S. Pan, A provable secure private data delegation scheme for mountaineering events in emergency system, *IEEE Access*, vol. 5, pp. 3410–3422, 2017.

[31] T.-Y. Wu, J. C.-W. Lin, C.-M. Chen, Y.-M. Tseng, J. Frnda, L. Sevcik, and M. Voznak, A brief review of revocable id-based public key cryptosystem, *Perspectives in Science*, vol. 7, pp. 81–86, 2016.

[32] C.-T. Li, T.-Y. Wu, C.-L. Chen, C.-C. Lee, and C.-M. Chen, An efficient user authentication and user anonymity scheme with provably security for iot-based medical care system, *Sensors*, vol. 17, no. 7, p. 1482, 2017.

[33] C.-M. Chen, K.-H. Wang, T.-Y. Wu, J.-S. Pan, and H.-M. Sun, A scalable transitive human-verifiable authentication protocol for mobile devices, *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1318–1330, 2013.

[34] C.-M. Chen, L. Xu, T.-Y. Wu, and C.-R. Li, On the security of a chaotic maps-based three-party authenticated key agreement protocol, *Journal of Network Intelligence (2)*, pp. 61–65, 2016.

[35] I. R. Jeong, J. O. Kwon, D. Hong, and D. H. Lee, Constructing peks schemes secure against keyword guessing attacks is possible, *Computer communications*, vol. 32, no. 2, pp. 394–396, 2009.