Journal of Network Intelligence ©2018 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **3**, Number **1**, February **2018**

Security Analysis of Rhee et al.'s Public Encryption with Keyword Search Schemes: A Review

Tsu-Yang Wu[1,2], Chien-Ming Chen[3], King-Hang Wang[4],
Jeng-Shyang Pan[1,2], Weimin Zheng[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
No33 Xuefunan Road, University Town, Fuzhou 350118, China
wutsuyang@gmail.com; jengshyangpan@fjut.edu.cn; zhengweimin@iie.ac.cn
[*]Corresponding author's email: zhengweimin@iie.ac.cn

[3]School of Computer Science and Technology
Harbin Institute of Technology, Shenzhen
HIT Campus of University Town of Shenzhen, Shenzhen 518055, China
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

Shu-Chuan Chu[5], John F. Roddick[5]

[5]College of Science and Engineering
Flinders University
Sturt Rd, Bedford Park SA 5042, South Australia
jan.chu@flinders.edu.au, john.roddick@flinders.edu.au

Received January, 2018; Revised February 2018

Abstract. *Public key encryption with keyword search (PEKS) provides an efficient way to search encrypted files. Recently, Rhee et al. contributed their knowledge to propose several literatures in this research area. In this paper, we first review their three famous schemes and then summarize the security weaknesses of the three schemes. Finally, we discuss the security problems about Rhee et al. like scheme and remain an open problem.*
Keywords: Searchable encryption, Keyword search, Keyword guessing attack, Cryptanalysis.

---

1. **Introduction.** With the fast growth of cloud and big data technologies [1, 2, 3], to outsource the personal files such as photos, videos, etc. to the cloud becomes popular behaviors. Meanwhile, user may adopt the related encryption technologies [4, 5, 6] to protect their files. Public key encryption with keyword search (PEKS) (or called searchable public key encryption) is a cryptographic primitive. It provides an efficient way to solve a critical problem that how to search an encrypted file using keyword in cloud server. The first PEKS scheme is introduced by Boneh et al. [7] in 2004 and the framework of PEKS is depicted in Figure 1. It describes three roles: a data owner, a server, and a data user, who can be the data owner himself or any other designated individual who has the

right of accessing the file. The data owner first encrypts the keywords with the file and user's public key. Then, she/he uploads to the server together with the encrypted data files. A data user wishes to retrieve file with a particular keyword, she/he will generate a trapdoor using her/his private key and the keyword she/he wants to search. This trapdoor is securely sent to the server. The server can test an encrypted keyword ciphertext matching with the trapdoor using some cryptographic means. The matching encrypted data will then sent to the user. Such framework was used in the subsequent works [8, 9].

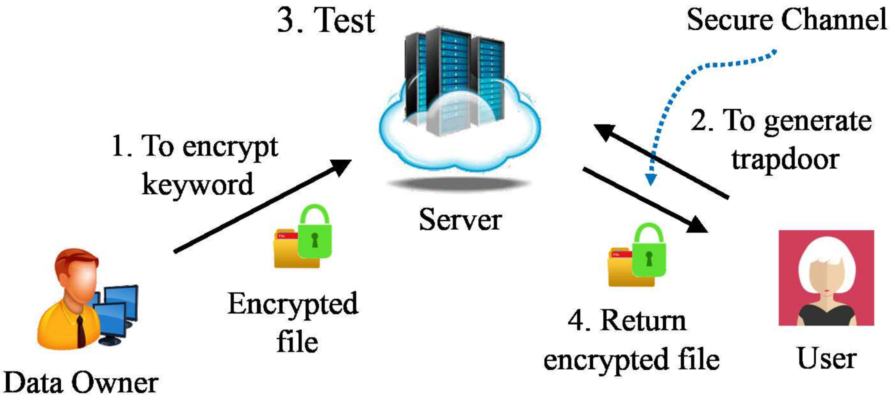

Figure 1. The framework of Boneh et al.'s PEKS scheme

In 2008, Beak et al. [10] proposed a PEKS scheme with designated verifier, namely SCF-PEKS. Their scheme first introduced the server role and pointed out that the attacker can be divided into malicious servers and outside attackers. However, their SCF-PEKS scheme is insecure against an off-line keyword guessing attack pointed by Rhee et al. [11]. Meanwhile, they proposed an improvement based on SCF-PEKS scheme. In 2010, Rhee et al. [12] proposed a variant of SCF-PEKS scheme called SCF-dPEKS (or called dPEKS for short). A dPEKS allows only a designated server to perform the keyword searching. When the encrypted keyword and the trapdoor are generated, both the user's public key and the server's public key are used. This framework allows the removal of secure channel between the data user and the server depicted in Figure 2. Later, several PEKS or dPEKS schemes based on different public key cryptosystems were proposed such as identity (ID)-based [?] and certificateless based [13, 14, 15, 16, 17].

However, as pointed by Shen et al. [18] it is inherently impossible to protect a trapdoor in the above PEKS framework. It is because everyone can generate the encrypted keyword using user's public key. Because the size of meaningful keyword space has a limitation about 2^{18}, attacker can simply enumerate on all possible keywords to construct an encrypted keyword and test that with the trapdoor. On the other hand, attacker can capture the trapdoor sent by the user (or called receiver) and then tests the trapdoor is related to which keyword in the above dPEKS framework. The two kinds of attacks are referred to off-line keyword guessing attacks [19, 20, 21, 22, 23, 24, 25, 26]. In 2010, Rhee et al. [12] defined a new security notion of dPEKS scheme called "Trapdoor indistinguishability" which allows a scheme to be formally proven secure against an outside attacker who wants to launch an off-line keyword guessing attack.

In this paper, we review and analyze Rhee et al.'s three famous dPEKS schemes [11, 20, 12]. We demonstrate the scheme [20] is suffered from an off-line keyword guessing (KG) attack launched by an outside attacker and all schemes are suffered from off-line KG attacks launched by a malicious (curious) server even the three schemes are proved

TABLE 1. Notations

Notation	Meaning
S	Server.
R	Receiver.
$\mathcal{KS}$	Keyword space.
λ	Security parameter.
g	Generator of $\mathbb{G}$.
sk_S	Server's private key.
pk_S	Server's public key.
sk_R	Receiver's private key.
pk_R	Receiver's public key.
H_1, H_4	Cryptographic map-to-point hash function, $H_1, H_4 : \{0,1\}^* \rightarrow \mathbb{G}$.
H_2	Cryptographic hash function, $H_2 : \mathbb{G}_T \rightarrow \{0,1\}^{\log p}$.
H_3	Cryptographic hash function, $H_3 : \mathbb{G}_T \rightarrow \{0,1\}^{\lambda}$.

"trapdoor indistinguishability". Finally, we summarize the security problems of the three schemes and remain an open problem about to resist off-line keyword guessing attacks launched by a malicious (curious) server in Rhee et al. like scheme is possible? This paper is organized as follows. In Section 2, we review and analyze Rhee et al.'s first dPEKS scheme called dPEKS-1 scheme including the concept of pairing. Then, we review and analyze Rhee et al.'s second dPEKS scheme called dPEKS-2 scheme in Section 3. In Section 4, we review and analyze Rhee et al.'s third dPEKS scheme called dPEKS-3 scheme. The conclusion and discussion are drew in Section 5.

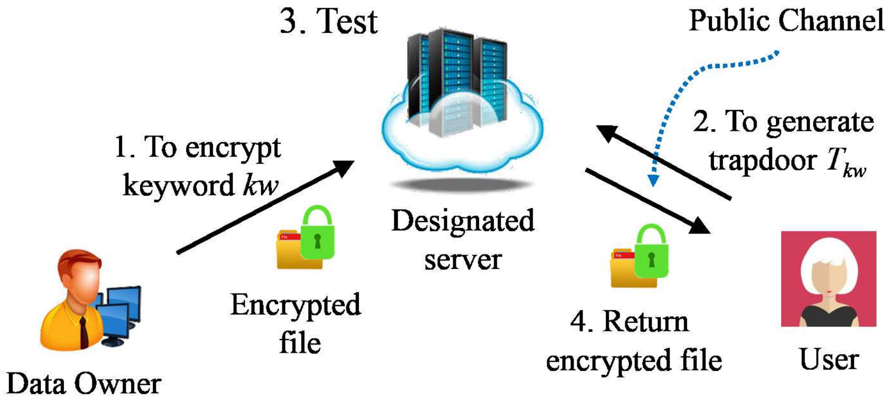

FIGURE 2. The framework of Rhee et al.'s dPEKS scheme

2. **Analysis of Rhee et al.'s first dPEKS scheme (dPEKS-1).**

In 2009, Rhee et al. [11] proposed a dPEKS scheme (named dPEKS-1 here) and claimed their dPEKS-1 scheme is secure against off-line keyword guessing (KG) attacks by outside attacker. However, we demonstrate that their dPEKS-1 scheme is still insecure against other off-line KG attacks by malicious (curious) server in this section. Firstly, we introduce the concept of pairing in the following subsection and the notations throughout in this paper are summarized in Table 1.

2.1. **Pairing.**

Let $\mathbb{E}$ be a non-singular elliptic curve over a finite field $\mathbb{F}$. To select two groups $\mathbb{G}$ and $\mathbb{G}_T$ with prime order p, where $\mathbb{G}$ is a multiplicative cyclic group of $\mathbb{E}_{\mathbb{F}}(x,y)$

and $\mathbb{G}_T$ is also a multiplicative cyclic group of $\mathbb{F}$. A pairing (or called bilinear pairing) is a map defined by $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ and satisfies the following properties.

1. *Bilinear.* For all u, $v \in \mathbb{G}$ and a, $b \in \mathbb{Z}_p^*$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. *Non-degenerate.* For any identity $1_{\mathbb{G}} \in \mathbb{G}$, we have $e(1_{\mathbb{G}}, 1_{\mathbb{G}}) = 1_{\mathbb{G}_T}$, an identity of $\mathbb{G}_T$.
3. *Computable.* There exist several algorithms to compute $e(u, v)$ for all u, $v \in \mathbb{G}$.

For the details about pairing, please refer [27, 28, 29, 30, 31, 32, 33, ?, 34] to for a full descriptions.

2.2. **Review of Rhee et al.'s dPEKS-1 scheme.** The dPEKS-1 scheme consists of following algorithms (phases).

1. *System setup.* Inputting a security parameter λ, this algorithm returns public parameters $param = \{\mathbb{G}, \mathbb{G}_T, p, e, g, H_1, H_2, \mathcal{KS}\}$, where g is a generator of $\mathbb{G}$, $H_1 : \{0, 1\}^* \to \mathbb{G}$, $H_2 : \mathbb{G}_T \to \{0, 1\}^{\log p}$, and $\mathcal{KS}$ is a keyword space.
2. *Key generation.* The server S's private key sk_S is defined by $sk_S = \alpha \in_R \mathbb{Z}_p^*$ and the corresponding public key pk_S is computed by $pk_S = g^\alpha$. Similarly, the receiver R's private/public key pair is defined by (sk_R, pk_R), where $sk_R = x \in_R \mathbb{Z}_p^*$ and $pk_R = g^x$.
3. *Keyword encryption.* To encrypt some keyword $w \in \mathcal{KS}$, this algorithm first selects a value $r_1 \in_R \mathbb{Z}_p^*$ and then computes the correspond cithertext of w by $C_w = \langle C_1, C_2 \rangle$, where
$$C_1 = (pk_R)^{r_1} \tag{1}$$
and
$$C_2 = H_2(e(pk_S, H_1(w)^{r_1})). \tag{2}$$
4. *Trapdoor generation.* For a specific keyword $w \in \mathcal{KS}$ selected by the receiver, this algorithm first selects a value $r_2 \in_R \mathbb{Z}_p^*$ and then computes the corresponding trapdoor of w by $T_w = \langle T_1, T_2 \rangle$, where
$$T_1 = (pk_S)^{r_2} \tag{3}$$
and
$$T_2 = H_1(w)^{1/x} \cdot g^{r_2}. \tag{4}$$
5. *Test.* To retrieve the encrypted keyword C_w, the receiver sends a trapdoor T_w to the server S. Then, S first computes
$$\Lambda = (T_2)^\alpha / (T_1)^{\alpha^2} \tag{5}$$
and then verifies
$$C_2 \overset{?}{=} H_2(e(C_1, \Lambda)). \tag{6}$$
If the verification holds, the server returns "1". Otherwise, it returns "0".

2.3. **Security weaknesses in dPEKS-1 scheme.** Here, we demonstrate that Rhee et al.'s dPEKS-1 scheme is insecure against off-line KG attacks by malicious (curious) server S. The functionality of S is defined by it can execute the steps of algorithms honestly but it is curious about the content of ciphertext C_w and trapdoor T_w.

2.3.1. *Hu and Liu's attack.* In 2012, Hu and Liu [22] pointed out the insecurity of Rhee et al.'s dPEKS-1 scheme. Assume that S received a trapdoor $T_w = \langle T_1, T_2 \rangle$ sent by the receiver. Then, it can execute the following steps to launch an off-line KG attack as follows.

(1) To compute

$$\Lambda = (T_1)^{1/\alpha}. \tag{7}$$

(2) To guess an appropriate keyword $w' \in \mathcal{KS}$.

(3) To verify

$$e(pk_R, T_2) \overset{?}{=} e(pk_R, \Lambda) \cdot e(g, H_1(w')). \tag{8}$$

If the verification is true, it means that T_w is generated by w'. Otherwise, S goes back to (2) and continues to execute (3).

2.3.2. *Our attack.* Here, we propose a similar attack approach to show the insecurity of Rhee et al.'s dPEKS-1 scheme. Also assume that S received a trapdoor $T_w = \langle T_1, T_2 \rangle$ sent by the receiver. Our attack is described that S executes the following steps to launch an off-line KG attack as follows.

(1) To compute $\Lambda = T_2 / T_1^{1/\alpha}$.

(2) To guess an appropriate keyword $w' \in \mathcal{KS}$.

(3) To verify

$$e(pk_R, \Lambda) \overset{?}{=} e(g, H_1(w')). \tag{9}$$

If the verification is true, it means that T_w is generated by w. Otherwise, S goes back to (2) and continues to execute (3).

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

$$e(pk_R, \Lambda) = e(g^x, H_1(w)^{1/x}) = e(g, H_1(w)). \tag{10}$$

3. Analysis of Rhee et al.'s second dPEKS scheme (dPEKS-2).

In 2009, Rhee et al. [20] proposed another dPEKS scheme (named dPEKS-2 here) and claimed their dPEKS-2 scheme is also secure against off-line KG attacks by outside adversary. However, we demonstrate that their dPEKS-2 scheme is still insecure against other off-line KG attacks by outsider adversary and malicious (curious) server in this section.

3.1. Review of Rhee et al.'s dPEKS-2 scheme.

The dPEKS-2 scheme consists of following algorithms (phases).

1. *System setup.* Inputting a security parameter λ, this algorithm returns public parameters $param = \{\mathbb{G}, \mathbb{G}_T, p, e, g, v, u, \widetilde{u}, H_1, H_3, \mathcal{KS}\}$, where g is a generator of $\mathbb{G}$, v, u, $\widetilde{u} \in_R \mathbb{G}$, $H_1 : \{0,1\}^* \to \mathbb{G}$, $H_3 : \mathbb{G}_T \to \{0,1\}^\lambda$, and $\mathcal{KS}$ is a keyword space.
2. *Key generation.* The server S's private key sk_S is defined by $sk_S = \alpha \in_R \mathbb{Z}_p^*$ and the corresponding public key pk_S is computed by $pk_S = (pk_{S,1}, pk_{S,2}, pk_{S,3}) = (g^\alpha, v^{1/\alpha}, u^{1/\alpha})$. Similarly, the receiver R's private/public key pair is defined by (sk_R, pk_R), where $sk_R = x \in_R \mathbb{Z}_p^*$ and $pk_R = (pk_{R,1}, pk_{R,2}, pk_{R,3}) = (g^x, v^{1/x}, \widetilde{u}^{1/x})$.
3. *Keyword encryption.* To encrypt some keyword $w \in \mathcal{KS}$, this algorithm first selects a value $r \in_R \mathbb{Z}_p^*$ and then computes the correspond cithertext of w by $C_w = \langle C_1, C_2 \rangle$, where

$$C_1 = (pk_{R,1})^r \tag{11}$$

and

$$C_2 = H_3(e(pk_{S,1}, H_1(w)^r)). \tag{12}$$

4. *Trapdoor generation.* For a specific keyword $w \in \mathcal{KS}$ selected by the receiver, this algorithm computes the corresponding trapdoor of w by

$$T_w = H_1(w)^{1/sk_R}. \tag{13}$$

5. *Test.* To retrieve the encrypted keyword C_w, the receiver sends a trapdoor T_w to the server S. Then, S verifies

$$C_2 \overset{?}{=} H_3(e(C_1, T_w^{sk_S})). \tag{14}$$

   If the verification holds, the server returns "1". Otherwise, it returns "0".

3.2. **Security weaknesses in dPEKS-2 scheme.** Here, we demonstrate that Rhee et al.'s dPEKS-2 scheme is insecure against off-line KG attacks by outside adversary $\mathcal{A}$ and malicious (curious) server S.

3.2.1. *Hu and Liu's attack.* Hu and Liu [22] also pointed out the insecurity of Rhee et al's dPEKS-2 scheme. Assume that $\mathcal{A}$ captures a trapdoor T_w sent by the receiver. Then, it can executes the following steps to launch an off-line KG attack as follows.

(1) To guess an appropriate keyword $w' \in \mathcal{KS}$.
(2) To verify

$$e(pk_{R,1}, T_w) \overset{?}{=} e(g, H_1(w')). \tag{15}$$

If the verification is true, it means that T_w is generated by w. Otherwise, $\mathcal{A}$ goes back to (1) and executes.

Note that this attack approach also can be launched by malicious (curious) server S.

3.2.2. *Our attack.* Here, we propose a similar attack approach to show the insecurity of Rhee et al.'s dPEKS-2 scheme. Also assume that $\mathcal{A}$ received a trapdoor T_w sent by the receiver. Our attack is described that $\mathcal{A}$ executes the following steps to launch an off-line KG attack as follows.

(1) To guess an appropriate keyword $w' \in \mathcal{KS}$.
(2) To verify

$$e(v, T_w) \overset{?}{=} e(pk_{R,2}, H_1(w')). \tag{16}$$

If the verification is true, it means that T_w is generated by w'. Otherwise, $\mathcal{A}$ goes back to (1) and executes.

Note that this attack approach also can be launched by malicious (curious) server S. Here, we explain the correctness of our attack. Assume that w' is the success guessed keyword. Then,

$$e(v, T_w) = e(v, H_1(w)^{1/sk_R}) = e(pk_{R,2}, H_1(w)). \tag{17}$$

4. **Analysis of Rhee et al.'s third dPEKS scheme (dPEKS-3).** In 2010, Rhee et al. [12] proposed a dPEKS scheme (named dPEKS-3 here) and claimed their dPEKS-3 scheme is also secure against off-line KG attacks by outside adversary. However, we demonstrate that their dPEKS-3 scheme is still insecure against other off-line KG attacks by malicious (curious) server in this section.

4.1. **Review of Rhee et al.'s dPEKS-3 scheme.** The dPEKS-3 scheme consists of following algorithms (phases).

1. *System setup.* Inputting a security parameter λ, this algorithm returns public parameters $param = \{\mathbb{G}, \mathbb{G}_T, p, e, g, u, \widetilde{u}, H_1, H_3, H_4, \mathcal{KS}\}$, where g is a generator of $\mathbb{G}$, u, $\widetilde{u} \in_R \mathbb{G}$, H_1, $H_4 : \{0,1\}^* \to \mathbb{G}$, $H_3 : \mathbb{G}_T \to \{0,1\}^\lambda$, and $\mathcal{KS}$ is a keyword space.
2. *Key generation.* The server S's private key sk_S is defined by $sk_S = \alpha \in_R \mathbb{Z}_p^*$ and the corresponding public key pk_S is computed by $pk_S = (pk_{S,1}, pk_{S,2}) = (g^\alpha, u^{1/\alpha})$. Similarly, the receiver R's private/public key pair is defined by (sk_R, pk_R), where $sk_R = x \in_R \mathbb{Z}_p^*$ and $pk_R = (pk_{R,1}, pk_{R,2}) = (g^x, \widetilde{u}^{1/x})$.
3. *Keyword encryption.* To encrypt some keyword $w \in \mathcal{KS}$, this algorithm first selects a value $r_1 \in_R \mathbb{Z}_p^*$ and then computes the correspond cithertext of w by $C_w = \langle C_1, C_2 \rangle$, where
$$C_1 = (pk_{R,1})^{r_1} \tag{18}$$
and
$$C_2 = H_3(e(pk_{S,1}, H_1(w)^{r_1})). \tag{19}$$
4. *Trapdoor generation.* For a specific keyword $w \in \mathcal{KS}$ selected by the receiver, this algorithm first selects a value $r_2 \in_R \mathbb{Z}_p^*$ and then computes the corresponding trapdoor of w by $T_w = \langle T_1, T_2 \rangle$, where
$$T_1 = g^{r_2} \tag{20}$$
and
$$T_2 = H_1(w)^{1/x} \cdot H_4(pk_{S,1}^{r_2}). \tag{21}$$
5. *Test.* To retrieve the encrypted keyword C_w, the receiver sends a trapdoor T_w to S. Then, the server first computes
$$\Lambda = T_2 / H_4(T_1^\alpha) \tag{22}$$
and then verifies
$$C_2 \overset{?}{=} H_3(e(C_1, \Lambda^\alpha)). \tag{23}$$
If the verification holds, the server returns "1". Otherwise, it returns "0".

4.2. **Security weaknesses in dPEKS-3.** Here, we demonstrate that Rhee et al.'s dPEKS-3 is insecure against off-line KG attacks by malicious (curious) server S.

4.2.1. *Wang et al.'s attack.* In 2011, Wang et al. [21] pointed out the insecurity of Rhee et al.'s dPEKS-3 scheme. Assume that S received a trapdoor $T_w = \langle T_1, T_2 \rangle$ sent by the receiver. Then, it can execute the following steps to launch an off-line KG attack as follows.

(1) To compute $\Lambda = T_2 / H_4(T_1^\alpha)$.
(2) To guess an appropriate keyword $w' \in \mathcal{KS}$.
(3) To verify
$$e(pk_{R,1}, \Lambda) \overset{?}{=} e(g, H_1(w')). \tag{24}$$
If the verification is true, it means that T_w is generated by w'. Otherwise, S goes back to (2) and continues to execute (3).

Table 2. Summary of off-line keyword guessing attacks on Rhee et al.'s three dPEKS schemes

Launched by	dPEKS-1 [11]	dPEKS-2 [20]	dPEKS-3 [12]
Outside adversary	No	**Yes** ([22], Our)	No
Malicious (curious) server	**Yes** ([22], Our)	**Yes** ([22], Our)	**Yes** ([21], Our)

4.2.2. *Our attack.* Here, we propose a similar attack approach to show the insecurity of Rhee et al.'s dPEKS-3 scheme. Also assume that S received a trapdoor $T_w = \langle T_1, T_2 \rangle$ sent by the receiver. Our attack is described that S executes the following steps to launch an off-line KG attack as follows.

(1) To compute $\Lambda = T_2/H_4(T_1^{\alpha})$.
(2) To guess an appropriate keyword $w' \in \mathcal{KS}$.
(3) To verify

$$e(\widetilde{u}, \Lambda) \stackrel{?}{=} e(pk_{R,2}, H_1(w')). \tag{25}$$

If the verification is true, it means that T_w is generated by w. Otherwise, S goes back to (2) and continues to execute (3).

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

$$e(\widetilde{u}, \Lambda) = e(\widetilde{u}, H_1(w)^{1/x}) = e(pk_{R,2}, H_1(w)). \tag{26}$$

5. Conclusions and discussions.

In this paper, we have reviewed Rhee et al.'s three famous dPEKS schemes and summarized the existed weaknesses of their schemes in Table 2. It is easy to see that to resist the off-line keyword guessing (KG) attacks launched by outside attacker in dPEKS scheme becomes possible, especially Rhee et al. [12] formalized the security model of trapdoor. However, it is very hard to resist the off-line KG attacks launched by malicious server in Rhee et al. like dPEKS scheme. It may remain to be an open problem.

Acknowledgment. The authors would like to thank the anonymous referees' constructive comments and valuable suggestions.

REFERENCES

[1] J. C.-W. Lin, W. Gan, P. Fournier-Viger, T.-P. Hong, and V. S. Tseng, Efficient algorithms for mining high-utility itemsets in uncertain databases, *Knowledge-Based Systems*, vol. 96, pp. 171–187, 2016.

[2] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, A survey of sequential pattern mining, *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 54–77, 2017.

[3] J. C.-W. Lin, W. Gan, P. Fournier-Viger, T.-P. Hong, and H.-C. Chao, Fdhup: Fast algorithm for mining discriminative high utility patterns, *Knowledge and Information Systems*, vol. 51, no. 3, pp. 873–909, 2017.

[4] 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.

[5] 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.

[6] C.-M. Chen, W. Fang, K.-H. Wang, and T.-Y. Wu, Comments on an improved secure and efficient password and chaos-based two-party key agreement protocol, *Nonlinear Dynamics*, vol. 87, no. 3, pp. 2073–2075, 2017.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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*, 2017.

[17] M. Ma, D. He, M. K. Khan, and J. Chen, Certificateless searchable public key encryption scheme for mobile healthcare system, *Computers & Electrical Engineering*, 2017.

[18] E. Shen, E. Shi, and B. Waters, Predicate privacy in encryption systems., in *TCC*, vol. 5444, pp. 457–473, Springer, 2009.

[19] 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.

[20] 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.

[21] 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.

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

[23] T.-Y. Wu, F. Meng, C.-M. Chen, S. Liu, and J.-S. Pan, On the security of a certificateless searchable public key encryption scheme, in *International Conference on Genetic and Evolutionary Computing*, pp. 113–119, Springer, 2016.

[24] C.-T. Li, C.-C. Lee, C.-Y. Weng, T.-Y. Wu, and C.-M. Chen, Cryptanalysis of an efficient searchable encryption against keyword guessing attacks for shareable electronic medical records in cloud-based system, in *International Conference on Information Science and Applications*, pp. 282–289, Springer, 2017.

[25] T.-Y. Wu, C. Meng, C.-M. Chen, K.-H. Wang, and J.-S. Pan, On the security of a certificateless public key encryption with keyword search, in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 191–197, Springer, 2017.

[26] T.-Y. Wu, C. Meng, K.-H. Wang, C.-M. Chen, and J.-S. Pan, Comments on islam et al.s certificateless designated server based public key encryption with keyword search scheme, in *International Conference on Genetic and Evolutionary Computing*, pp. 199–205, Springer, 2017.

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

[28] L. Chen, Z. Cheng, and N. P. Smart, Identity-based key agreement protocols from pairings, *International Journal of Information Security*, vol. 6, no. 4, pp. 213–241, 2007.

[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] T.-Y. Wu and Y.-M. Tseng, An efficient user authentication and key exchange protocol for mobile client–server environment, *Computer Networks*, vol. 54, no. 9, pp. 1520–1530, 2010.

[31] T.-Y. Wu, Y.-M. Tseng, and T.-T. Tsai, A revocable id-based authenticated group key exchange protocol with resistant to malicious participants, *Computer Networks*, vol. 56, no. 12, pp. 2994–3006, 2012.

[32] T.-Y. Wu and Y.-M. Tseng, Publicly verifiable multi-secret sharing scheme from bilinear pairings, *IET Information Security*, vol. 7, no. 3, pp. 239–246, 2013.

[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.-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.