A Study of Public Key Encryption with Keyword Search

Shih-Ting Hsu[1], Chou-Chen Yang[1], and Min-Shiang Hwang[2]
(Corresponding author: Min-Shiang Hwang)

Department of Management Information Systems & National Chung Hsing University[1]
250 Kuo Kuang Road, Taichung, Taiwan 402, R.O.C.
Department of Computer Science and Information Engineering & Asia University[2]
500 Liufeng Road, Wufeng, Taichung, Taiwan 402, R.O.C.
(Email: corresponding_tracy01601@gmail.com)
(Invited Paper)

Abstract

Public Key Encryption with Keyword Search (PEKS) scheme enable one to search the encrypted data with a keyword without revealing any information. The concept of a PEKS scheme was proposed by Boneh et al. in 2004 and Baek et al. who extended PEKS scheme into a secure channel free PEKS scheme (SCF-PEKS) which removes the assumption, a secure channel between users and a server. In this paper, we show an overview of six existing security models of PEKS/SCF-PEKS scheme and conclude five security requirements that must satisfy to construct a secure PEKS/SCF-PEKS scheme. Then we compare the security and efficiency of the security models and discuss the future researches of PEKS/SCF-PEKS.

Keywords: PEKS, off-line keyword-guessing attack, designate tester, trapdoor indistinguishability

1 Introduction

Considering a scenario that user Bob wants to share documents with user Alice, Bob has two options to achieve this work. Bob stores the documents in the mobile devices such as flash drive and portable hard drive, and sends it to Alice. However there are many uncertain situations in deliverying process; for example, the mobile devices might be stolen by the corporate espionage so to cause huge damage to the company. Another option for Bob to share documents is taking the server as the storage media. Traditionally, users upload and store their data in the remote server and take the remote server as a storage media. Users can upload, download, update and delete the data in seconds, and they can further authorize other users to use the data for some specific purposes. However, the security, integrity and confidentiality of data in the remote server cannot be guaranteed because users cannot control their data directly and cannot supervise how a remote server manages them clearly. The remote server is just like an untrusted third party. Therefore, users usually encrypted their documents for the privacy propose and ensuring data security before uploading to the remote server. However, as the document is transformed into a ciphertext, it produces another problem; that is how users can obtain the encrypted data without decryptin them.

Although attackers and the server administrator caiesnot distinguish what the context is as they capture the encrypted data, users can not define which ciphertext is the one they want, either. One of the solutions is that users download all the encrypted data and decrypt them, so that users can find the right documents they want without revealing any information to the server administrator. Nevertheless, this solution might cause lots of transfer cost and storage space whenever users query data. If Alice wishes to only retrieve the documents which contain the word W, downloading the whole encrypted data is not a suitable solution and unrealistic. Another solution is to set up keywords for each encrypted documents and a user can search the encrypted documents with specific keywords they wish to query. In order to achieve this goal, Song et al. [6] first brought up the concept of searching the encrypted data with certain words in 2000. They thought that there are two alternatives to search on the ciphertext; that is to build up an index for each word W and perform a sequential scan without an index. The latter do not need extra space to store the index, but slower than the former. However, the index-based schemes seem to require less sophisticated constructions, Song et al. proposed a scheme which works by computing the bitwise exclusive or (XOR) of the clear-text with a sequence of pseudorandom bits which have a special structure [6]. The solution of Song et al. requires very little communication between the user and the server, requires only one round of interaction [2]. Therefore, Boneh et al. further proposed a brand-new scheme that searches the encrypted data based on keyword [2].

Public Key Encryption with Keyword Search (PEKS in short) scheme, which is also name searchable public-key encryption scheme, enables one to search encrypted documents on the untrusted server without revealing any information. Boneh et al. first introduced PEKS scheme with a mail routing system in 2004. There are three entities in PEKS: data sender, receiver and server. Suppose user Alice (receiver) has a number of devices: laptop, desktop, mobile device, etc. User Bob (data sender) wishes to send an email to Alice. First, he encrypts the email M with keywords $w_1, w_2, \ldots, w_m$ using Alice's public key and also appends the encrypted keywords $PEKS(A_{pub}, w_1)$, $PEKS(A_{pub}, w_2)$, $\cdots$, $PEKS(A_{pub}, w_m)$. Then he sends the following ciphertext to the mail server (server):

$$E_{A_{pub}}(M) \mid\mid PEKS(A_{pub}, w_1) \mid\mid \cdots \mid\mid PEKS(A_{pub}, w_m)$$

Where A_{pub} is Alice's public key. For Alice, she wishes to read the mails that contain keyword "urgent" using her mobile devices. For this purpose, Alice can give the server a certain trapdoor T_w of keyword 'urgent' that enables the server to find out the encrypted emails associated with 'urgent'. However, Alice does not want to reveal any private information to anyone including the server. In other words, the mail routing system must have the ability to test whether "urgent" is a keyword in the emails and route these mails to Alice's mobile device without getting anything else about the email.

However, Boneh et al.'s scheme has to construct the secure channel to protect trapdoors through out the transport process. This is not suitable for some applications as building a secure channel which is usually costly. To solve this problem, Baek et al. [1] proposed a new PEKS that removes the secure channel assumption and names "Secure Channel Free Public Key Encryption with Keyword Search (SCF-PEKS in short)" in 2008. In SCF-PEKS scheme, the data sender uses the server's public key and receiver's public key to encrypt the keywords each time he stores the encrypted data to the server. Whenever a receiver wants to search the encrypted data associated with a specific keyword, he can send the trapdoor to retrieve data via a *public channel*(public network) since only the server has the corresponding private key which can test whether the PEKS ciphertext matches the trapdoor. Nevertheless, the trapdoors can be transferred in the public network because trapdoors can be captured by anyone and it produces another problem: whether the outside adversaries can derive the embedded keyword and user information from the trapdoor by any means? Byun et al. [4] pointed out that PEKS might be attacked by the off-line keyword-guessing attacks in 2006. Since keywords are chosen from much smaller space than passwords and users usually use well-known keywords (low entropy) for searching documents [4]. Therefore, attackers can capture the trapdoor and have chance to guess keyword. In the other hand, Yau et al. [23] also demonstrated that outside adversaries that capture the trapdoors sent in a public channel can reveal encrypted keywords by performing off-line keyword-guessing attacks. From now on, most of the PEKS/SCF-PEKS scheme pay more attention on improving the security against the outside off-line keyword guessing attacks [7, 9, 14, 16, 20, 23]. However, all of the schemes still cannot stand against off-line keyword guessing attacks and only few schemes [11, 24] can stand against off-line keyword guessing attacks from outside adversaries.

1.1 Security Requirements

In short, PEKS idea provides a mechanism that enables users to search encrypted emails with keywords without revealing any information including the server. Also, the server can retrieve encrypted emails containing specific keywords, but learn nothing else about the emails. Besides SCF-PEKS eliminate the limitation of PEKS which require a secure channel between server and receiver and search encrypted data with keywords, which is more applicable in reality. On the other hand, PEKS/SCF-PEKS augments the security and privacy protection of data storage applications. Since cloud computing becomes the popular issue in recent years, more and more cloud services bloom in a very short time including cloud storage service. Thus, PEKS/SCF-PEKS scheme can increase the personal documents protection over cloud environment. To construct a secure PEKS or SCF-PEKS scheme with privacy protection, there are some security requirements needed to achieve as follows:

Trapdoor indistinguishability [19]
The trapdoor is produced by Alice's private key that searches the encrypted documents and the keyword. No one can distinguish the difference if two trapdoors are generated by the same keyword. Namely, no one can obtain any information from the trapdoor.

Ciphertext indistinguishability [19]
As users send encrypted documents to Alice, they will generate the keyword ciphertext that contains keywords $w_1, w_2, \ldots, w_m$ and append it to the encrypted emails. Even the keyword ciphertext is captured in the transfer process, no one can get the embedded keywords from the ciphertext.

Authorized identity protection (Anonymity)
Users send the ciphertext to the server with the public key of an authorized user who can search and download the encrypted emails. Similarly to ciphertext indistinguishability, no one should learn the authorized users' identity from the keyword ciphertext for the privacy purpose.

User authentication
Although no one can know the authorized users' identity, the server still has to recognize whether the trapdoor is uploaded by the authorized users. Therefore, the server must have the ability to authenticate the users' identities.

Against off-line keyword-guessing attacks
Since everything transferred over the public network is totally appreciable and easy to eavesdrop, the trapdoor might be captured by the outside attackers easily. On the contrary, the untrusted server might regard as the inside attacker if it tries to alter, expose, or derive the secret information from the trapdoor. Thus, the proposed scheme should stand against outside and inside off-line keyword-guessing attacks successfully.

1.2 Organization

This paper is organized as follows: In Section 2, we introduce the development of the PEKS schemes and analyse their advantages and shortcomings. We further evaluate whether the schemes in Section 2 conform the requirements mentioned above, and make a performance comparison in Section 3. We discuss futures issue such as conjunctive keyword search schemes in Section 4 and conclude in Section 5.

2 Security Model for PEKS/SCF-PEKS

2.1 PEKS Schemes

The notion of Public Key Encryption with Keyword Search (PEKS) scheme is proposed by Boneh, Crescenzo, Ostrovsky and Persiano in 2004 [2]. Their construction is based on a variant of the Computational Diffie-Hellman problem. In abstracto, they use two cyclic groups G_1, G_2 of prime order p, a bilinear map $e : G_1 \times G_1 \rightarrow G_2$. The map satisfies the following properties:

1) Computable: given $g, h \in G_1$ there is a polynomial time algorithms to compute $e(g, h) \in G_2$.

2) Bilinear: for any integers $x, y \in [1, p]$ we have $e(g^x, g^y) = e(g, g)^{xy}$.

3) Non-degenerate: if g is a generator of G_1 then $e(g, g)$ is a generator of G_2.

There also needs two hash functions $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : G_2 \rightarrow \{0,1\}^{logp}$ and the security parameter $\{G_1, G_2, e, H_1, H_2, g, h\}$. Boneh et al.'s scheme works as follows:

- KeyGen: The input security parameter determines the size, p, of the groups of G_1 and G_2. Then, the algorithm chooses a random value $\alpha \in G_q^*$ and a generator g of G_1. It outputs $A_{pub} = [g, h = g^\alpha]$ and $A_{priv} = \alpha$.
- PEKS(A_{pub}, w): First choose a random value $r \in Z_p^*$ and compute $t = e(H_1(w), h^r) \in G_2$. Output $S = [g^r, H_2(t)]$.
- Trapdoor(A_{priv}, w'): Output $T_{w'} = H_1(w')^\alpha \in G_1$.
- Test($A_{pub}, S, T_{w'}$): Let $S = [A, B]$. Test if $H_2(e(T_{w'}, A)) = B$. Output 'yes' if the equation holds and 'no' otherwise.

2.2 SCF-PEKS Schemes

In Baek, Safavi-Naini and Susilo's opinion, Boneh *et al.*'s scheme [2] uses a secure channel between receiver and server and constructing the secure channel is costly and inefficient. In other words, the trapdoor cannot be sent via a public network. This is not suitable for some applications [1]. Thus, Baek *et al.* proposed a mechanism to remove the secure channel in an efficient way. The basic idea they use is making server keep its own public key pair. To create a PEKS ciphertext, data sender uses server's public key and receiver's public key to encrypt the keywords. As a receiver wishes to query the encrypted documents with keyword w', he has to generate the trapdoor with his private key. At this time, the trapdoor can be sent via public a network since only the server which has the corresponding private key can perform the Test algorithm.

2.2.1 Baek *et al.*'s Scheme

The secure channel free public key encryption with keyword search scheme which is also named as searchable keyword encryption with a designated tester proposed by Baek, Safavi-Naini and Susilo in 2008 [1]. Baek et al.'s scheme is based on bilinear pairing consisting of the following algorithms:

1) GlobalSetup(k): Take a security parameter k and generate a group $G_1 =< P >$ with prime order $q \geq 2^k$. Then construct a bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$, where the order of G_2 is q. And use two hash functions $H_1 : \{0,1\}^* \rightarrow G_1^*$ and $H_2 : G_2 \rightarrow \{0,1\}^k$. Then output the global parameter $gp = (q, G_1, G_2, e, P, H_1, H_2, d_w)$, where d_w denotes a description of a keyword space.

2) KeyGen$_{Server}$(gp): Choose two random value $x \in Z_q^*$ and $Q \in G_1^*$ then compute $X = xP$. Output public key $pk_S = (gp, Q, X)$ and private key $sk_S = (cp, x)$.

3) KeyGen$_{Receiver}$(gp): Choose a random value $y \in Z_p^*$ and compute $Y = yP$. Output public key $pk_R = (gp, Y)$ and private key $sk_R = (gp, y)$.

4) SCF-PEKS(gp, pk_S, pk_R, w): Choose a random value $r \in Z_q^*$ and compute $S = (U, V) = (rP, H_2(\kappa))$, where $\kappa = (e(Q, X)e(H_1(w), Y))^r$. Output S as a PEKS ciphertext.

5) Trapdoor(gp, sk_R, w'): Compute $T_{w'} = yH_1(w')$. Output $T_{w'}$ as a trapdoor for keyword w'.

6) Test($gp, T_{w'}, S, sk_S$): Check if $H_2(e(xQ + T_w, U)) = V$. If the equation holds return 'yes' and 'no' otherwise.

2.2.2 Rhee *et al.*'s Scheme

In 2009, Rhee, Park, Susilo and Lee pointed out that Baek *et al.*'s scheme [1] might be attacked by using a keyword-guessing attack if the outside attacker captures the trapdoor [18]. Therefore, Rhee *et al.* enhances the model of Baek *et al.* to prevent such attacks and defines the "trapdoor indistinguishability" [19]. On one hand, the data sender uses server's public key and receiver's public key to generate a PEKS ciphertext. On the other hand, the receiver uses the server's public key and his private key to generate the trapdoor. Thus, if the trapdoor is captured by the outside attacker, he cannot perform the keyword-guessing attack successfully without the server's private key. The algorithms of Rhee *et al.*'s SCF-PEKS scheme [19] are as follows:

1) GlobalSetup(λ): Let G_1 and G_2 be bilinear groups of prime order p. Given a security parameter λ, first picks a random generator $g \in G_1$ and two random elements $u, \tilde{u} \in G_1$. Then construct a bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$ and use three hash functions $H : \{0,1\}^* \rightarrow G_1, H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : G_2 \rightarrow \{0,1\}^\lambda$. This algorithm outputs a global parameter $gp = (p, G_1, G_2, e, H, H_1, H_2, g, u, \tilde{u})$.

2) KeyGen$_{Server}(gp)$: First chooses a random value $\alpha \in Z_p$ and set private key $sk_S = \alpha$, and compute public key $pk_S = (pk_{S,1}, pk_{S,2}) = (g^{sk_S}, u^{1/sk_S})$. Output server's public key pairs (pk_S, sk_S).

3) KeyGen$_{Receiver}(gp)$: Choose a random value $\beta \in Z_p$ and set $sk_R = \beta$, and compute $pk_R = (pk_{R,1}, pk_{R,2}) = (g^\beta, \tilde{u}^\beta)$. Output receiver's public key pairs (pk_R, sk_R).

4) SCF-PEKS(gp, pk_S, pk_R, w): Choose a random value $r \in Z_q^*$ and set $A = pk_{R,1}^r$ and $B = H_2(e(pk_{S,1}, H_1(w)^r))$. Output PEKS ciphertext $S = [A, B]$.

5) Trapdoor(gp, pk_S, sk_R, w'): Choose a random value $r' \in Z_q^*$ and compute $T_1 = g^{r'}$ and $T_2 = H_1(w')^{\frac{1}{\beta}} \cdot H(pk_{s,1}^{r'})$. Output a trapdoor $T_{w'} = [T_1, T2]$.

6) Test$(gp, S, T_{w'}, sk_S)$: First compute $T = T_2/H(T_1^\alpha)$ and check if $B = H_2(e(A, T^\alpha))$. If the above equalities are satisfied, then output 'yes' and 'no' otherwise.

2.2.3 Zhao *et al.*'s Scheme

After Rhee et al. introduced the notion of "trapdoor indistinguishability", Zhao, Chen, Ma, Tang and Zhu [24] proposed another SCF-PEKS that can successful stand against an outside keyword-guessing attack and achieve better performance than Rhee *et al.*'s scheme [19] in 2012. Zhao *et al.*'s scheme consists of the following algorithms:

1) GlobalSetup(k): Generate a group G_1 of prime order $q \geq 2^k$, a random generator P of G_1 and construct a bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$. This algorithm uses two hash function $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : G_2 \rightarrow \{0,1\}^k$. Output global parameter $gp = (q, G_1, G_2, e, P, H_1, H_2, d_w)$, where d_w denotes a description of a keyword space.

2) KeyGen$_{Server}(gp)$: Choose $x \in Z_q^*$ uniformly at random and compute $X = xP$. Choose $Q \in Z_q^*$ uniformly at random. Output Server's public key $pk_S = (gp, Q, X)$ and private key $sk_S = (gp, x)$.

3) KenGen$_{Receiver}(gp)$: Choose $y \in Z_q^*$ uniformly at random and compute $Y = yP$. Output Receiver's public key $pk_R = (gp, Y)$ and private key $sk_R = (gp, y)$.

4) SCF-PEKS(gp, pk_S, pk_R, w): Choose a random value $r \in Z_p^*$ and compute $S = (U, V, t) = (rP, rY, t)$ where $t = e(H_1(w), rP)e(rQ, X)$. Output S as a PEKS ciphertext.

5) Trapdoor(gp, sk_R, w'): Choose a random value $\tilde{a} \in \{0,1\}^*$. Then compute $T_{w1} = [y^{-1}H_1(w') + H_1(\tilde{a})] \bigoplus [H_1(e(yQ, xP))]$ and $T_{w2} = yH_1(\tilde{a}) \in G_1$. Output $T_{w'} = (T_{w1}, T_{w2})$ as a trapdoor for keyword w'.

6) Test$(gp, S, T_{w'}, sk_S)$: First compute $\eta = T_{w1} \bigoplus H_1(e(xQ, yP))$, and compute $\delta = e(T_{w2}, U), t' = e(xQ, U)^{-1}$ and $T = tt' = e(H_1(w), rP)$. Finally, Test if $H_2(e(\eta, V)) = H_2(T \cdot \delta)$. If the equation holds, output 'yes' and 'no' otherwise.

2.3 PEKS Scheme Without Using Pairing

2.3.1 Khader's Scheme

In Boneh et al.'s PEKS scheme [2], they presented several methods based on different security models but these methods had some limitations. Since Boneh et al.'s scheme is proven secure in random oracle which has been shown possibly not secure in the standard model [5], their schemes were not secure enough in Khader's opinion [15]. Therefore, Khader presented a new scheme called Public Key Encryption with Keyword Search based on K-Resilient IBE [10](KR-PEKS in short) in 2006.

- KeyGen:

  1) Choose a group G of order q and two generator g_1, g_2.
  2) Choose 6 random k degree polynomials chosen over Z_q.

  $f_1(x) = a_0 + a_1x + a_2x^2 + \cdots + a_kx^k = \sum_{t=0}^k a_tx^t$,
  $f_2(x) = a'_0 + a'_1x + a'_2x^2 + \cdots + a'_kx^k = \sum_{t=0}^k a'_tx^t$,
  $h_1(x) = b_0 + a_1x + b_2x^2 + \cdots + b_kx^k = \sum_{t=0}^k b_tx^t$,
  $h_2(x) = b'_0 + a_1x + b'_2x^2 + \cdots + b'_kx^k = \sum_{t=0}^k b'_tx^t$,
  $p_1(x) = d_0 + a_1x + d_2x^2 + \cdots + d_kx^k = \sum_{t=0}^k d_tx^t$,
  $p_2(x) = d'_0 + d'_1x + d'_2x^2 + \cdots + d'_kx^k = \sum_{t=0}^k d'_tx^t$,

3) For $0 \le t \le k$; Compute $A_t = g_1^{a_t} g_2^{a_t'}$, $B_t = g_1^{b_t} g_2^{b_t'}$, $D_t = g_1^{d_t g_2^{d_t'}}$.
4) Choose a random collision resistant hash function $H : G \to \{0,1\}^\lambda$.
5) Choose a random targeted collision resistant hash function TCR.
6) Assign public key $pk_R = (g_1, g_2, A_0, \ldots, A_k, B_0, \ldots, B_k, D_0, \ldots D_k, H, TCR)$ and private key $sk_R = (f_1, f_2, h_1, h_2, p_1, p_2)$.

- KR-PEKS:
  1) Choose a random value $r \in Z_q$.
  2) Compute $u_1 = g_1^r, u_2 = g_2^r$.
  3) Calculate for each keyword w $A_w \leftarrow \prod_{t=0}^{k} A_t^{(w^t)}$; $B_w \leftarrow \prod_{t=0}^{k} B_t^{(w^t)}$; $D_w \leftarrow \prod_{t=0}^{k} D_t^{(w^t)}$.
  4) $s \leftarrow D_w^r$.
  5) $e \leftarrow (0^\lambda) \bigoplus H(s)$.
  6) $\alpha \leftarrow TCR(u_1, u_2, e)$.
  7) $v_w \leftarrow (A_w)^r \cdot (B_w)^{r\alpha}$.
  8) $C \leftarrow < u_1, u_2, e, v_w >$.
- Trapdoor: $T_{w'} =< f_1(w'), f_2(w'), h_1(w'), h_2(w'), p_1(w'), p_2(w') >$.
- Test:
  1) $\alpha \leftarrow TCR(u_1, u_2, e)$.
  2) Test if $v_w \ne (u_1)^{f_1(w')+h_1(w')\alpha} (u_2)^{f_2(w')+h_2(w')\alpha}$.
  3) $s \leftarrow (u_1)^{p_1(w')}(u_2)^{p_2(w')}$.
  4) $M \leftarrow e \bigoplus H(s)$.
  5) If $M = 0^\lambda$, output 'yes' and 'no' otherwise.

2.3.2 Yang *et al.*'s Schemes

Most of PEKS/SCF-PEKS schemes presented were constructed based on bilinear pairing. Moreover, keywords have low entropy and are chosen from the much smaller space than passwords [4]. In Yang et al.'s opinion, PEKS schemes(including SCF-PEKS schemes) based on pairing are susceptible to off-line keyword-guessing attacks. In order to construct a more secure scheme against off-line keyword-guessing attacks, Yang, Xu and Zhao presented a public key encryption with keyword search scheme not using pairing in 2011 [22]. Their scheme is a variant of Khader's PEKS scheme [15] which overcomes the shortcoming of Khader's scheme (do not satisfy the consistency) and improves efficiency.

- KeyGen
  1) Choose a group G of order q and two generator g_1, g_2.
  2) Choose 4 random k degree polynomials chosen over Z_q. $f_1(x) = a_0 + a_1x + a_2x^2 + \cdots + a_kx^k = \sum_{t=0}^{k} a_tx^t$,
  $f_2(x) = a_0' + a_1'x + a_2'x^2 + \cdots + a_k'x^k = \sum_{t=0}^{k} a_t'x^t$,
  $h_1(x) = b_0 + a_1x + b_2x^2 + \cdots + b_kx^k = \sum_{t=0}^{k} b_tx^t$,
  $h_2(x) = b_0' + a_1x + b_2'x^2 + \cdots + b_k'x^k = \sum_{t=0}^{k} b_t'x^t$,
  3) For $0 \le t \le k$; Compute $A_t = g_1^{a_t} g_2^{a_t'}$, $B_t = g_1^{b_t} g_2^{b_t'}$, $D_t = g_1^{d_t} g_2^{d_t'}$.
  4) Choose a random collision resistant hash function $H : G \to \{0,1\}^\lambda$.
  5) Choose a random targeted collision resistant hash function TCR.
  6) Let SKE be a one-time symmetric-key encryption scheme.
  7) Assign public key $pk_R = (g_1, g_2, A_0, \ldots, A_k, B_0, \ldots, B_k, H, TCR)$ and private key $sk_R = (f_1, f_2, h_1, h_2)$.
- KR-PEKS
  1) Choose a random value $r \in Z_q$.
  2) Compute $u_1 = g_1^r, u_2 = g_2^r$.
  3) Calculate for each keyword w $A_w \leftarrow \prod_{t=0}^{k} A_t^{(w^t)}$; $B_w \leftarrow \prod_{t=0}^{k} B_t^{(w^t)}$.
  4) $\alpha \leftarrow TCR(u_1, u_2)$.
  5) $v_w \leftarrow (A_w)^r \cdot (B_w)^{r\alpha}$.
  6) $K \leftarrow H(v_w); R \leftarrow \{0,1\}^\lambda$.
  7) $e \leftarrow SKE.Enc(K, R)$.
  8) $C \leftarrow < R, u_1, u_2, e >$.
- Trapdoor $T_{w'} =< f_1(w'), f_2(w'), h_1(w'), h_2(w') >$.
- Test
  1) $\alpha \leftarrow TCR(u_1, u_2)$.
  2) Test if $v_w \ne (u_1)^{f_1(w')+h_1(w')\alpha} \cdot (u_2)^{f_2(w')+h_2(w')\alpha}$.
  3) $K \leftarrow H(v_w)$.
  4) $R' \leftarrow SKE.Dec(K, c)$.
  5) If $R = R'$, output 'yes' and 'no' otherwise.

3 Comparisons

In this section, we present a comparison of security and performance for the schemes mentioned in Section 2.

3.1 Security Analysis

Table 1 shows the security comparison among PEKS/SCF-PEKS schemes. We use Trap Ind, Ciph Ind, AuthID Prot, User Auth, Inside KG and Outside KG to denote Trapdoor ingistinguishability, PEKS(SCF-PEKS) Ciphertext indistinguiability, authorized identity protection, user authentication, against inside off-line keyword-guessing attack and against outside off-line keyword-guessing attack, respectively.

Since Boneh *et al.*'scheme and Baek *et al.*'s scheme do not use random number in the trapdoor algorithm, adversaries can easily distinguish the embedded keyword of the captured trapdoors from previous trapdoors that had found out the keywords by off-line keyword-guessing attack. Besides, we could find out that all the schemes satisfy the property of ciphertext indistinguishability, authorized identity protection and user authentication, but on the other hand, all the schemes cannot guarantee the security of the malicious server. Since the data sender and receiver should provide enough information to the server to recognize the authorized users' identities, the server gain sufficient messages from PEKS ciphertexts and trapdoors and can perform off-line keyword-guessing attack easily.

3.2 Performance Analysis

Let E denotes an exponentiation operation, P denotes a Maptopoint hash function operation [3], M denotes a multiplication operation in G_1, e denotes a pairing operation and f denotes a polynomial operation. Maptopoint hash function means the operation of mapping a keyword to an element in G_1, which is so inefficient [9]. Besides, k represents the maximum number of trapdoors (private key) generated in the KR-PEKS [10]. We neglect the operation of hash function that maps a keyword to an element in Z_p^* used in all the schemes because it only requires little of operating time. Table 2 displays the evaluation of performance aimed at computational load of each algorithm with previous schemes including three PEKS schemes ([2], [15] and [22]) and three SCF-PEKS schemes ([1], [19] and [24]).

For the data sender, Zhao *et al.*'s scheme needs the less computational load to generate the PEKS ciphertexts. On the other hand, Baek *et al.*'s scheme produces minimal computational load for the receiver at generating trapdoor phase. Although Baek *et al.*'s scheme has smaller computational loads in PEKS/SCF-PEKS, Trapdoor and Test than other schemes, it is not secure enough in facing inside adversaries (Baek *et al.*'s scheme cannot against the insider off-line keyword-guessing attack as showen in Table 1).

4 Future Research

4.1 Multi-keyword Search

Suppose Alice's friends send a number of emails to Alice and those emails are all stored in the same mail server. Alice would wish to retrieve the emails which contain some keywords, e.g. "urgent", "Monday" and "Marking Department". In PEKS/SCF-PEKS schemes, Alice cannot generate the trapdoor using more than one keyword. If Alice only uses one keyword to search through hundreds of emails, she might retrieve a huge number of related emails and most of them are undesired. In 2004, Golle, Staddon and Waters first proposed the notion of *secret key* encryption with conjunctive field keyword search scheme [8]. Park, Kim and Lee proposed a new security model in an asymmetrical cryptography system which is named Public Key Encryption with Conjunctive Field Keyword Search (PECKS) [17]. Later, many researches improved the efficiency of the conjunctive keyword search, but most of the schemes still have room for improvement.

4.2 Delegated Search

The concept of public key encryption with delegated keyword search (PKEDS) is proposed by Ibraimi, Nikova, Hartel and Jonker in 2011 [13]. Suppose Alice encrypts an email with the public key of Bob, and Alice's computer is infected by some virus and embeds a malware into the email in the unknown situation. If Bob decrypts those email himself, his computer will be infected by the malware. Since malware is encrypted, the server is unable to scan and detect the malware directly. The simple idea is sending receiver's private key to the server, but it is not secure for receiver. Thus, the notion of PEKDS is that users give server an delegated master trapdoor that does not reveal users' private key and the server will check all the encrypted data for them. Tang, Zhao, Chen and Ma showed that Ibraimi *et al.*'s scheme has some defects and inefficient, so they presented a more secure and efficient PKEDS scheme in 2012 [21].

4.3 Multi-user Keyword Search

Consider a situation that data sender wishes to share his document with more than one user; in most of existing PEKS security models and PECKS security models, he has to stores N documents if there are N users he wishes to authorize. It is inefficient that data sender stores a number of same encrypted documents. A multi-user PECK (mPECK) was introduced by Hwang and Lee [12] in 2007, but has much less discussion than PEKS and PECKS later.

5 Conclusions

Public Key Encryption with Keyword Search scheme enables one to search encrypted data without revealing

Table 1: Security comparison

	Boneh et al.	Khader's	Baek et al.	Rhee et al.	Zhao et al.	Yang et al.
Trap Ind	×	○	×	○	○	○
Ciph Ind	○	○	○	○	○	○
AuthID Prot	○	○	○	○	○	○
User Auth	○	○	○	○	○	○
Inside KG	×	×	×	×	×	×
Outside KG	×	○	×	○	○	○

Table 2: Performance comparison

	Boneh et al.	Khader's	Baek et al.	Rhee et al.	Zhao et al.	Yang et al.
KeyGen_{Server}	-	-	M	$2E$	M	-
$\mathsf{KeyGen}_{Receiver}$	E	$6E$	M	$2E$	M	$6E$
PEKS/SCF-PEKS	$2E + 2P + e$	$(3k + 8)E$	$E + M + P + 2e$	$2E + P + e$	$4M + P + 2e$	$(2k + 6)E$
Trapdoor	$E + P$	$6f$	$P + M$	$2E + 2P$	$3M + 4P + e$	$4f$
Test	e	$4E + 6f$	$M + e$	$2E + P + e$	$2M + P + 4e$	$2E + 4f$

any information to anyone. In this paper, we study six important schemes and analyze their efficiency and performance. Moreover, we conclude five security requirements that must satisfy as constructing PEKS/SCF-PEKS scheme. Finally, we briefly discuss three extend issues about a keyword search scheme. We hope that this paper can help more researchers deeply understand this field. Therefore, the development of public key encryption with keyword search schemes and its extend issues can be rapidly developed.

Acknowledgments

This study was supported by the National Science Council of Taiwan under grant NSC 95-2416-H-159-003. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *ICCSA 2008*, vol. 5072 of *Lecture Notes in Computer Science*, pp. 1249–1259, Perugia, Italy, 2008. Springer Berlin/Heidelberg.

[2] D. Boneh, G. D. Crescenzom, R. Ostrovsky, and G. Rersiano, "Public key encryption with keyword search," in *Advances in Cryptology - EUROCRYPT 2004, Lecture Notes in Computer Science*, vol. 3027, pp. 506–522, Interlaken, Switzerland, 2004. Springer Berlin/Heidelberg.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Crytology-CRYPTO 2001*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229. Springer Berlin, Heidelberg, 2001.

[4] J. W. Byun, H. A Park, H. S. Rhee, and D. H. Lee, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data," in *Secure Data Management, Lecture Notes in Computer Science*, vol. 4165, pp. 75–83, Seoul, Korea, 2006. Springer Berlin/Heidelberg.

[5] R. Canetti, O. Goldreich, and S. Halavi, "The random oracle methodology, revisited," in *In: Proc. of 30th ACM STOC*, pp. 209–218, New York, 2004.

[6] D. W. Dawn, X. Song 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, 2000.

[7] L. Fang, W. Susilo, C. Ge, and J. Wang, "A secure channel free public key encryption with keyword search scheme without random oracle," in *Cryptology and Network Security*, vol. 5888 of *Lecture Notes in Computer Science*, pp. 248–258, Kanazawa, Japan, 2009. Springer Berlin/Heidelberg.

[8] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *In Proceedings of Applied Cryptography and Network Security Conference*, vol. 3089 of *Lecture Notes in Computer Science*, pp. 31–45. Springer, Heidelberg, 2004.

[9] C. Gu and Y. Zhu, "New efficient searchable encryption schemes from bilinear pairings," *International Journal of Network Security*, vol. 10, no. 1, pp. 25–31, 2010.

[10] S. H. Heng and K. Kurosawa, "k-resilient identity-based encryption in the standard model," in *CT-RSA*

2004, vol. 2964 of *Lecture Notes in Computer Science*, pp. 67–80. Springer Berlin, Heidelberg, 2004.

[11] C. Hu and P. Liu, "A secure searchable public key encryption scheme with a designated tester against keyword guessing attacks and its extension," in *Advances in Computer Science, Environment, Ecoinformatics, and Education*, vol. 215 of *Communications in Computer and Information Science*, pp. 131–136, Wuhan, China, 2011. Springer Berlin/ Heidelberg.

[12] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Pairing-Based Cryptography- Pairing 2007*, vol. 4575 of *Lecture Notes in Computer Science*, pp. 2–22. Springer, Heidelberg, 2007.

[13] L. Ibraimi, S. Nikova, P. Hartel, and W. Honker, "Public-key encryption with delegated search," in *Applied Cryptography and Network Security - 9th International Conference, ACNS 2011*, vol. 6715 of *Lecture Notes in Computer Science*, pp. 532–549. Springer, Heidelberg, 2011.

[14] 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.

[15] D. Khader, "Public key encryption with keyword search based on k-resilient ibe," in *Computational Science and Its Application - ICCSA 2006*, vol. 3982 of *Lecture Notes in Computer Science*, pp. 298–308. Springer, Heidelberg, 2006.

[16] Q. Liu, G. Wang, and J. Wu, "An efficient privacy preserving keyword search scheme in cloud computing," in *2009 International Conference on Computational Science and Engineering*, pp. 715–720, Vancouver, BC, 2009. IEEE computer society.

[17] D. J. Park, K. Kim, and P. J. Lee, "Public key encryption with conjunctive field keyword search," in *Information Security Applications, 5th Interational Workshop, WISA 2004*, vol. 3325 of *Lecture Notes in Computer Science*, pp. 73–86. Springer, Heidelberg, 2005.

[18] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Improved searchable public key encryption with designated tester," in *ASIACCS '09 Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pp. 376–379, Sydney, NSW, Australia, 2009. ACM New York, NY, USA.

[19] 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," *The Journal of Systems and Software*, vol. 83, no. 5, pp. 763–771, 2010.

[20] 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.

[21] Q. Tang, Y. Zhao, X. Chen, and H. Ma, "Refine the concept of public key encryption with delegated search," 2012.

[22] H. M. Yang, C. X. Xu, and H. T. Zhao, "An efficient public key encryption with keyword scheme not using pairing," in *2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pp. 900–904, Beijing, China, 2011.

[23] W. C. Yau, S. H. Heng, and B. M. Goi, "Off-line keyword guessing attacks on recent public key encryption with keyword search schemes," in *Autonomic and Trusted Computing*, vol. 5060 of *Lecture Notes in Computer Science*, pp. 100–105, Oslo, Norway, 2008. Springer Berlin/ Heidelberg.

[24] Y. Zhao, X. Chen, H. Ma, Q. Tang, and H. Zhu, "A new trapdoor-indistinguishable public key encryption with keyword search," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 3, no. 1/2, pp. 72–81, 2012.

Shih-Ting Hsu was born in Taoyang County, Taiwan, in 1988. She received her B.M. in Information Management from Yuan Ze University in 2011. She is currently pursuing the M.S. degree with the Department of Management Information Systems from National Chung Hsing University. Her research interests include cloud computing, information security, and cryptography.

Chou-Chen Yang received his B.S. in Industrial Education from the National Kaohsiung Normal University, in 1980, and his M.S. in Electronic Technology from the Pittsburg State University, in 1986, and his Ph.D. in Computer Science from the University of North Texas, in 1994. From1994 to 2004, Dr. Yang was an associate professor in the Department of Computer Science and Information Engineering, Chaoyang University of Technology. Currently, he is a professor in the Department of Management Information Systems, National Chung Hshing University. His research interests include network security, mobile computing, and distributed system.

Min-Shiang Hwang received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, ROC, in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician the first class in 1990. He was a chairman of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999-2002. He is currently a professor of the department of Management Information System, National Chung Hsing University, Taiwan, ROC. He obtained 1997, 1998, 1999, 2000, and 2001 Outstanding Research Awards of the National Science Council of the Republic of China. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang had published 170+ articles on the above research elds in international journals.