ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

A Literature Survey on Searchable Public Key Encryption

Puneet Dheer

Department of Computer Science and Engineering, S.R.M. University, T.N., India

ABSTRACT: In a searchable public key encryption scheme it allows users to search the encrypted data with a keyword without disclosing any information. There are various keyword based searches using public key encryption are implemented. In 2004, the first PEKS scheme was proposed by Boneh et al. and then Baek et al. who improved PEKS scheme into a secure channel (SSL) free PEKS scheme (SCF-PEKS) which removes an assumption of secure channel (SSL) between users and a server. In this paper, it shows an overview of six existing security models of PEKS/SCF-PEKS/PKEDS scheme and discussed seven security requirements that must satisfy to construct a develop PEKS/SCF-PEKS/PKEDS scheme. Then it compares the security and complexity/efficiency of the security models and discuss the future researches.

KEYWORDS*:* PEKS, SCF-PEKS, PKEDS, trapdoor

I. INTRODUCTION

Assume that user Bob wants to send the data to Alice. Bob to share data in the cloud server as the storage media. Traditionally, users upload and store their data in the cloud server as a storage media. Users can upload, download, manipulate and delete the data in few seconds, and they can further grant other users to use the data according to the purposes, for example: Google drive.The cloud server just like an untrusted third party.The security, integrity and confidentiality of data in the cloud server cannot be guaranteed because users cannot control their data directly and cannot supervise the cloud server to manage the data.Therefore, users generally encrypted their data for the privacy propose before uploading data to the cloud server. However, the data in converted into ciphertext, it produces another problem that is how users can obtain the encrypted data.Then the users download all the encrypted data (no preference for ciphertext is the way they want) and decrypt them, so that users can find the right data they want without revealing any information to the administrator of the server. But it might causes lots of transfer cost and storage space locally whenever users query data. If Alice wants to retrieve the data which contain the word W, by downloading the whole encrypted data is not a suitable solution. Another way is to set up keywords for each encrypted data and user can search the encrypted data with specific keywords they want to query. In 2000, to achieve this task, Song et al. [5] first proposed the concept of searching the encrypted data with certain words. In that there are two way to search on the ciphertext, that is to build up an index for each word W and perform a sequential scan without an index. In another one do not need extra space to store the index, but slower than the previous one. However, the index-based schemes seem to require less complex constructions, Song et al. proposed a scheme which works by computing the bitwise XOR of the plaintext with a sequence of pseudorandom bits which have a special structure [5]. The technique of Song et al. requires very less communication between the user and the server, requires only one round of interaction [2]. Therefore, Boneh et al. further proposed a new scheme that searches the encrypted data based on keyword [2].

Public Key Encryption with Keyword Search scheme, which is also known as searchable public-key encryption scheme, allows to search encrypted data on the untrusted server without compromising any information. Boneh et al. first proposed PEKS scheme with a mail routing system. There are three components in PEKS: data sender, receiver and server. Suppose user Alice (receiver) has a number of devices: laptop, desktop, etc. User Bob (data sender) wants to send an email to Alice. First, he encrypts the email M with keywords w1, w2, . . . , wm using Alice's public key and also appends the encrypted keywords PEKS(Apub, w1), PEKS(Apub, w2), • • • , PEKS(Apub, wm). Then he sends the following ciphertext to the mail server (server):

$$EApub(M) \,||\, PEKS(Apub,w1) \,||\, \bullet\bullet\bullet \,||\, PEKS(Apub,wm)$$

Where Apub is Alice's public key. For Alice, she wishes to read the mails that contain keyword "office" using her mobile devices. For this purpose, Alice can give the server a certain trapdoor Tw of keyword "office" that allows the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

server to find out the encrypted emails associated with "office" keyword. The mail routing system must have the ability to test whether “office” 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. [2] scheme has to develop the secure channel (such as SSL) to protect trapdoors throughout the communication. Setup a secure channel which is usually costly. To resolve this problem, Baek et al. [1] proposed a new PEKS,“Secure Channel Free - Public Key Encryption with Keyword Search (SCF-PEKS)” that removes the secure channel assumption. In SCF-PEKS scheme, the data sender Bob 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 Alice wants to search the encrypted data associated with a specific keyword, Alice can send the trapdoor to retrieve data via a 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 captured by the outside attackers can derive the embedded keyword because trapdoor transferred in the public network. In 2006, Byun et al. [4] pointed out that PEKS might be attacked by the off-line keyword-guessing attacks. Since keywords are chosen from much smaller space than passwords and users usually use well-known keywords (low entropy) for searching data [4]. Therefore, attackers can capture the trapdoor and have chance to guess keyword. In 2008, Yau et al. [19] also demonstrated that outside attackers that capture the trapdoors sent in a public channel can reveal encrypted keywords by performing offline keyword guessing attacks.

Now onwards, most of the PEKS/SCF-PEKS/PKEDS scheme pay more concentration on improving the security to protect from the outside off-line key word guessing attacks [6], [8], [12], [13], [17], and [19]. But, all of the schemes still cannot protect from off-line keyword guessing attacks and only few schemes [9], [20] can protect off-line keyword guessing attacks from outside attackers.

A. Pre-requisite Security

Due to cloud computing becomes the popular issue in recent years. Thus, PEKS/SCF-PEKS/PKEDS scheme can increase the personal data protection over cloud environment. To develop a secure PEKS/SCF-PEKS/PKEDS scheme with privacy protection, there are some security requirements needed to achieve as follows:

- *Trapdoor indistinguishability:* The trapdoor is generated by Alice’s private key that searches the encrypted data and the keyword, it is infeasible for an adversary (except the server) to learn any information about any word from the trapdoor.
- *Ciphertext in distinguishability:*Sender Bob will generate the keyword ciphertext that contains keywords w1, w2, . . . , wm and append to the encrypted emails, before sending the data to receiver Alice. Even the keyword ciphertext is captured in the transfer process, it is infeasible for an adversary (other than the server) to learn any information about any word from the ciphertext.
- *Ciphertext one-wayness:* It is hard for an attacker to invert the ciphertext and to learn the word even if the attackers holds the server’s private key, the master trapdoor and the trapdoor associated with that word.
- *Authorized identity protection:*Sender Bob send the ciphertext to the server with the public key of an authorized user who can search and download the encrypted emails. Similarly to cipher- text indistinguishability, none should learn the authorized users’ identity from the keyword ciphertext for the privacy purpose.
- *User authentication:* Although none 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:*The trapdoor might be captured by the outside attackers easilybecauseeverything transferred over the public network is totally easy to trap. Contrary, the untrusted server might regard as the inside attacker if it tries to manipulate or derive the secret information from the trapdoor. Thus, the proposed scheme should protect from outside and inside off-line keyword guessing attacks successfully.

B. Organization of the Paper

The rest of the paper is organized as follows: In Section 2, discuss the background development of the PEKS schemes and analyze their advantages and disadvantages. In Section 3 to evaluate whether the schemes in Section 2 confirm the requirements mentioned above, and make an efficiency comparison. Then discuss the futures research in Section 4 and conclude in Section 5.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

II. SECURITY FRAMEWORK FOR PEKS/SCF-PEKS/PKEDS

A. Public Key Encryption with Keyword Search (PEKS) Schemes

The Public Key Encryption with Keyword Search (PEKS) scheme is proposed by Boneh et. al. [2]. It is based on a variant of the Computational Diffie-Hellman problem. In abstract, it uses 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:

- **Computable:** given $g,h \in G_1$ there is a polynomial time algorithms to compute $e(g,h) \in G_2$.
- **Bilinear:** for any integers $x,y \in [1,p]$ we have $e(g^x,g^y) = e(g,g)^{xy}$.
- **Non-degenerate:** if g is a generator of G_1 then $e(g,g)$ is a generator of G_2.
- Two hash functions $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : G_2 \rightarrow \{0,1\}^{logp}$ and the security parameter $\{G_1,G_2,e,H_1,H_2,g,h\}$.

It consist of the four algorithms:

1) **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 $A_{priv} = \alpha$.
2) **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 $S = [g^r,H_2(t)]$.
3) **Trapdoor(A_{priv},w^0):** Output $T_w0 = H_1(w^0)^{\alpha} \in G_1$.
4) **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.

Alice runs the **KeyGen** algorithm to generate her public/private key pair. Alice uses Trapdoor to generate trapdoors T_W for any keywords w that she wants the mail server or mail gateway to search for. The mail server uses the given trapdoors as input to the **Test ()** algorithm to determine whether a given email contains one of the keywords W specified by Alice

B. Secure Channel Free - Public Key Encryption with Keyword Search (SCF-PEKS) Schemes (Revisited)

In Baek *et. al.*view, Boneh *etal.*'s scheme [2] uses a secure channel which is costly developing the secure channel and inefficient between receiver and server. This is not suitable for some applications [1]. Thus, Baek *etal.* proposed a mechanism to remove the secure channel and by making server keep its own public key pair. To generate a PEKS ciphertext, data sender bob uses server's public key and receiver's alice public key to encrypt the keywords. As a receiver alice wants to query the encrypted data with keyword w^0, alice has to generate the trapdoor with his private key. At this time, the trapdoor can be sent via public a network because only the server which has the corresponding private key can excute the Test Algorithm.

*1) **Baek et al.'s Scheme**: In 2008, Baek et al.[1] proposed that to remove the need for secure channel for transmitting the trapdoors in the original PEKS scheme.*

Baek et al.'s scheme consist of the following algorithms:

1) GlobalSetup(k): Take a security parameter k and generate a group G1 =< P > with prime order q ≥ 2k. Then construct a bilinear pairing e : G1 × G1 → G2, where the order of G2 is q. It use two hash functions $H_1 : \{0,1\}^* \rightarrow G_1^*$ and H2 : G2 → {0,1}k. Then output the global parameter gp = (q,G1,G2,e,P,H1,H2,dw), where dw denotes a description of a keyword space.
2) KeyGenServer(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) KeyGenReceiver(gp): Choose a random value $y \in Z_p^*$ and $Q \in G_1^*$ compute Y = yP. Output public key pkR = (gp,Y) and private key sk_R = (gp,y).
4) SCF-PEKS(gp,pkS,pkR,w): Choose a random value $r \in Z_q^*$ and compute S = (U,V) = (rP,H2(κ)) , where κ = $(e(Q,X)e(H1(w),Y))^r$. Output S as a PEKS ciphertext.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

5) Trapdoor(gp,skR,w^0): Compute Tw^0 = yH1(w^0). Output Tw^0 as a trapdoor for keyword w^0.
6) Test(gp,Tw^0,S,skS): Check if *H2(e(xQ + Tw,U))* = V .Output 'yes' if the equation holds and 'no' otherwise.

*2) **Rhee et al.'s Scheme(designated tester):** In 2009, Rhee et al. defines the "trapdoor indistinguishability"[16]. The data sender uses server's public key and receiver's public key to generate a PEKS ciphertext. Then 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, they cannot perform the keyword-guessing attack successfully without the server's private key.*

The algorithms of Rhee *etal.*'s SCF-PEKS scheme [16] 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 $H_2 : G_2 \rightarrow \{0,1\}^\lambda$.
It 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 $pkS = (pk_S,1,pk_S,2) = (gsk_S,u1/sk_S)$.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 $A=pk_R{}^r{}_1$ 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^0): Choose a random value $r' \in Z_q^*$ and compute $T_1 = g^{r0}$ and $T_2 = H_1(w^0)^{\beta 1} \cdot H(pk_{s,1}^r)$. Output a trapdoor $T_w0 = [T_1,T2]$.
6) Test(gp,S,T_w0,sk_S): First compute $T = T_2/H(T_1^\alpha)$ and check if $B = H_2(e(A,T^\alpha))$. Output 'yes' if the equation holds and 'no' otherwise.

*3) **Zhao et al.'s Scheme (new trapdoor- indistinguishable):** In 2012, Zhao et al. [20] proposed another SCF-PEKS that can successful stand against an outside keyword-guessing attack and achieve better performance than Rhee et al.'s scheme [16].*

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^0):** Choose a random value $\tilde{a} \in \{0,1\}^*$.L Then compute $T_{w1} = [y^{-1}H_1(w^0) + H_1(\tilde{a})]$ $[H_1(e(yQ,xP))]$ and $T_{w2} = yH_1(\tilde{a}) \in G_1$. Output $T_{w0} = (T_{w1},T_{w2})$ as a trapdoor for keyword w^0.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

6) **Test(L_{gp},S,T_{w0},sk_S):** First compute $\eta = T_{w1}\, H_1(e(xQ,yP))$, and compute $\delta = e(T_{w2},U)$,$t^0 = e(xQ,U)^{-1}$ and $T = tt^0 = e(H_1(w),rP)$. Finally, Test if $H_2(e(\eta,V)) = H_2(T \cdot \delta)$.Output 'yes' if the equation holds and 'no' otherwise.

C. *Public Key Encryption with Delegated Keyword Search (PKEDS) Schemes*

In 2011, Ibraimi et al. [11] proposed scheme based on ElGamal publickey encryption (PKE). It generate the ciphertexts which are both searchable and decryptable. This property is crucial since an entity can search the entire content of the message, in contrast to existing searchable public-key encryption schemes where the search is done only in the metadata part (PEKS).

Ibraimi*et al.*'s scheme consists of the following algorithms:

1) **Setup(1^λ):** This algorithm outputs public parameters (pp) which contain the description of groups $< G_1,G_2,G_T >$of prime order p, a bilinear map $\hat{e}$: $G_1 \times G_2 \rightarrow G_T$, g_1 and g_2 as the generators of groups G_1 and G_2 respectively.

2) **sKeyGen(pp):** Run by a server, this algorithm selects $x \in_R Z_p$ and outputs the server's key pair: $(SK_s,PK_s) = (x, g^x_2)$.

3) **rKeyGen(pp):**Run by a receiver, this algorithm selects $y,\alpha \in_R Z_p$ and outputs the receiver's private/public key pair:$(SK_r,PK_r) = ((y, g^\alpha_2), g_1^y)$.

4) **Encrypt(PK_r,w):** On input of the receiver's public key and a word $w \in G_1$, this algorithm selects $k \in_R Z_p$ and outputs the ElGamal ciphertext:

$$\begin{aligned} c_w &= (c_1, c_2) \\ &= (w \cdot (PK_r)^k,\ g_1^k) \\ &= (w \cdot g_1^{y \cdot k},\ g_1^k). \end{aligned}$$

5) **Delegate(PK_s,SK_r):** The algorithm creates a master trapdoor to let the server search the encrypted data for any word of his choice. The algorithm picks at random $r_1,r_2 \in Z_p$ and outputs the master trapdoor:

$$\begin{aligned} t_* &= (t_1, t_2, t_3, t_4) \\ &= (g_2^\alpha \cdot (PK_s)^{r_1},\ g_2^{r_1},\ g_2^{y \cdot \alpha} \cdot (PK_s)^{r_2},\ g_2^{r_2}) \\ &= (g_2^\alpha \cdot g_2^{x \cdot r_1},\ g_2^{r_1},\ g_2^{y \cdot \alpha} \cdot g_2^{x \cdot r_2},\ g_2^{r_2}). \end{aligned}$$

6) **TrapGen(SK_r,PK_s,w)** : The algorithm creates a trapdoor to let the server search for a specific massage w. The algorithm selects $\delta \in_R Z_p$ and outputs the trapdoor:

$$\begin{aligned} t_w &= (t_5,\ t_6) \\ &= (\hat{e}(w, g_2^\alpha) \cdot \hat{e}(PK_r, (PK_s)^\delta),\ g_2^\delta) \\ &= (\hat{e}(w, g_2^\alpha) \cdot \hat{e}(g_1, g_2)^{y \cdot x \cdot \delta},\ g_2^\delta). \end{aligned}$$

7) **Test$_1$(c_w,t_*,t_w,SK_s):** The algorithm tests whether the ciphertext contains the same massage as the trapdoor. The algorithm parses c_w as (c_1,c_2), t_*as (t_1,t_2,t_3,t_4), t_w as (t_5,t_6) and defines:

$$t_7 = \frac{t_1}{t_2^x},\ t_8 = \frac{t_3}{t_4^x}, \tilde{a} = \frac{\hat{e}(PK_r, t_6^x) \cdot \hat{e}(c_1, t_7)}{t_5}, \tilde{b} = \hat{e}(c_2, t_8).$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

Finally, the algorithm checks whether $a\tilde{} = b\tilde{}$. If this equation holds, the algorithm outputs TRUE indicating that the ciphertext contains the same massage as the trapdoor, otherwise it outputs FALSE.

8) Test$_2$(c_w,t_*,w,SK_s): The algorithm tests whether the ciphertext contains the word w. The algorithm parses c_w as (c_1,c_2), t_*as (t_1,t_2,t_3,t_4), and defines:

$$t_7 = \frac{t_1}{t_2^x}, t_8 = \frac{t_3}{t_4^x}, \tilde{c} = \hat{e}(c_1, t_7), \hat{d} = \hat{e}(c_2, t_8).$$

9) Decrypt(SK_r,c_w): The algorithm outputs:

$$w = \frac{c_1}{c_2^y}.$$

D. Refinement of Public Key Encryption with Delegated Keyword Search (PKEDS) Schemes

In 2012, Tang et al. [18], removed the undesirable function in original PKEDS. Suppose one server performsvirus scanning by running Test2, while the other server search and fetch the email by running Test1. The rationale behind this is that usersoften forward their emails from different accounts to an email archive (e.g. Gmail) and retrieve emails fromthe archiveafterwards. In such a situation, it is unnecessary to assign a master trapdoor tothe servers,which only performsearch based onmessage-dependent trapdoors.

It uses the bilinear groups. Let G_1,G_2 and G_T be groups of prime order p, and let g_1 and g_2 be generator of G_1 and G_2, respectively. A bilinear map ê: $G_1 \times G_2 \rightarrow G_T$ has the following properties:

- Bilinearity: for all $u \in G_1$, $v \in G_2$ and $a,b \in Z^*_p$, we have $\hat{e}(u^a,v^b) = \hat{e}(u,v)^{ab}$.
- Non-degeneracy: ê(g_1, g_2) not equal to1.
- Efficient computability: There is a polynomial time algorithm to compute $e\hat{}(u,v)$, for any $u \in G_1$ and $v \in G_2$.

1) Setup(λ): On input of the security parameterλ, the algorithm outputs public parameters (pp) which contain the description of groups G_1,G_2 of order p, the bilinear map $e\hat{}$: $G_1 \times G_2 \rightarrow G_T$, generators g_1 and g_2 of groups G_1 andG_2 respectively. Additionally, let (Enc, Dec) be a symmetric key encryption scheme, and KDF be a key derivation function.

2) sKeyGen(pp): On input of the public parameters pp, the algorithm picks uniformly at random $x \in_R Z_p$ and outputs a key pair:

$$(SK_s,PK_s) = (x, g^x_2)$$

3) rKeyGen(pp): On input of the public parameters pp, the algorithm picks uniformly at random $y \in_R Z_p$ and $h \in_R G_2$, and outputs the receiver‘s key pair:

$$(SK_r,PK_r) = ((y,h^y,MSK^\dagger),(g_1^y,h,MPK^\dagger)) ,$$

where ($MPK^\dagger$,$MSK^\dagger$) is the master puclic/private key pair for the BonehFranklin IBE scheme.

4) Encrypt(w,PK_r): On input of a message $w \in G_1$ and the receiver‘s public key PK_r, the algorithm picks uniformly at random $k \in_R Z_p$ and outputs a ciphertext:

$$c_w = (c_1,c_2,c_3) = (w \cdot g_1^{y \cdot k}, g^k_1, \text{Encrypt}^\dagger(g^k_2,w)).$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

The value Encrypt$^{\dagger}(g^{k}_{2},w)$ represents a ciphertext of g_2^k under the identity w in the Boneh-Franklin IBE scheme.

5) **TrapGen(w,PK_s,SK_r) :** On input of a message w, a server's public key $PK_s = g^{x}_{2}$, and the receiver's private key SK_r, the algorithm performs as follows.
   (a) Compute $t_3 = e\hat{}(w, g_2)$ and $SK_w^{\dagger}$.
   (b) Select uniformly at random $r_2 \in_R G_2$ and $r_3 \in_R Z_p$, and compute the following:

$$t4 = r2 \cdot gx2 \cdot r3,\ t5 = g2r3,\ t6 = \mathrm{Enc}(SKw\dagger \,||t3,\ \mathrm{KDF}(r2)).$$

   (c) Output the trapdoor $t_{w,s} = (t_4,t_5,t_6)$.

The value $SK_w^{\dagger}$ represents the secret key corresponding to the identity w in the Boneh-Franklin IBE scheme[7].

6) **Test$_1$($c_w,t_{w,s},SK_s$):** On input of a ciphertext c_w, a message-dependent trapdoor $t_{w,s}$ and a server's private key SK_s, the algorithm performs as follows.
   a) Parse c_w as (c_1,c_2,c_3) and $t_{w,s}$ as (t_4,t_5,t_6).
   b) Compute r2 from t4,t5 by an ElGamal decryption, and decrypt t6 with KDF(r2) to obtain (SKw† ,t3).
   c) Run the decryption algorithm of the Boneh-Franklin scheme to decrypt c3 using SKw† , and recover gk2.
   d) Compute a˜ as follows:

$$\tilde{a} = \frac{\hat{e}(c_1, g_2)}{\hat{e}(g_1^{y}, g_2^{k})}.$$

   e) Check whether the following two equalities hold. If both of them hold, the algorithm outputs 1, otherwise outputs 0.

$$\tilde{a} = t_3,\ \hat{e}(g_1, g_2^{k}) = \hat{e}(g_1^{k}, g_2).$$

7) **Delegate(PK_s,SK_r):** On input of a server's public key PK_s and the receiver's private key SK_r, the algorithm picks uniformly at random $r_1 \in_R Z_p$ and outputs the master trapdoor $t_{*,s}$,

$$t*,s = (t_1, t_2) = (h^{y} \cdot g_2^{x \cdot r_1}, g_2^{r_1})$$

After receiving $t_{*,s}$, the server with private key SK_s can recover

$$h^{y} = \frac{t_1}{t_2^{x}}$$

8) **Test$_2$($c_w,w,t_{*,s},SK_s$):** On input of a ciphertext c_w, a message w, a master trapdoor $t_{*,s}$ and a server's private key SK_s, the algorithm performs as follows.
   a) Parse c_w as (c_1,c_2,c_3).
   b) Compute $c\tilde{}$ and $d\tilde{}$ as follows:

$$\tilde{c} = \hat{e}(\frac{c_1}{w}, h),\ \tilde{d} = \hat{e}(c_2, h^{y}).$$

   Note that h^y can be pre-computed by the server once receiving t∗,s.
   c) Check whether c˜ = d˜. If this equation holds, the algorithm outputs 1, otherwise outputs 0.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

9) Decrypt(c_w,SK_r): On input of the ciphertext c_w and the receiver's private key SK_r, the algorithm outputs

$$w = \frac{c_1}{c_2^{\frac{1}{y}}}.$$

III. COMPARISONS

A. *Security Analysis*

Table 1 shows the security comparison of PEKS/SCF-PEKS/PKEDS schemes and it includes Trapdoor indistinguishability (Trap Ind), PEKS (SCF-PEKS) Ciphertext indistinguishability (Ciph Ind), authorized identity protection (AuthID Prot), user authentication (User Auth), against inside off-line keyword-guessing attack (Inside KG) and against outside off-line keyword-guessing attack (Outside KG).All the schemes satisfy the property of ciphertext indistinguishability, authorized identity protection and user authentication, but all the schemes cannot guarantee that the security of the malicious server. Since the data sender and receiver should provide enough information to the server to recognize the authorized user identities.

B. *Efficiency Analysis*

- Table 2 displays the evaluation of performance aimed at complexity of each algorithm with previous schemes including one PEKS schemes [2], three SCF-PEKS schemes ([1], [16] and [20]).
- Table 3 shows the performance between Ibraimi et al. and Tang et al.
- Let E denotes an exponentiation operation, P denotes a Map to point 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]. The operation of hash function that maps a keyword to an element in Z_p^* used in all the schemes is not considered here because it only requires little operating time.
- Zhao *etal.*'s scheme needs the less complexity to generate the PEKS ciphertexts.
- Baek *etal.*'s scheme produces less complexity for the receiver Alice at generating trapdoor phase.
- Although Baek *etal.*'s scheme has smaller complexity in PEKS/SCF-PEKS, Trapdoor and Test than other schemes, it is not secure enough in handling inside attacker (Baek *etal.*'s scheme cannot protect the insider off-line keyword-guessing attack as shown in Table 1). At last Tang et al improved the security and efficiency of security model.

Table 1: Security comparison

	Boneh et al.	Baek et al.	Rhee et al.	Zhao et al.	Ibraimi et al.	Tang et al.
Trapdoor Indis	No	No	Yes	Yes	Yes	Yes
Cipher Indis	Yes	Yes	Yes	Yes	Yes	Yes
AuthID Prot	Yes	Yes	Yes	Yes	Yes	Yes
User Auth	Yes	Yes	Yes	Yes	Yes	Yes
Inside KeyGuess	No	No	No	No	-	-
Outside KeyGuess	No	No	Yes	Yes	-	-
Ciphertext one-wayness	-	-	-	-	Yes	Yes

Table 2: Efficiency comparison 1

	Boneh et al.	Baek et al.	Rhee et al.	Zhao et al.
KeyGenServer	-	M	2E	M
KeyGenReceiver	E	M	2E	M
PEKS/SCF-PEKS	2E + 2P + e	E + M + P + 2e	2E + P + e	4M + P + 2e
Trapdoor	E + P	P + M	2E + 2P	3M + 4P + e
Test	e	M + e	2E + P + e	2M + P + 4e

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

Table 3: Efficiency comparison 2

		Ibraimi et al.	Tang et al.
Server(s)	sKeyGen	1 Exp	1 Exp
	Test1	3 Exp + 3 Pairing	1 Exp + 5 Pairing
	Test2	2 Exp + 3 Pairing	2 Pairing
Receiver	rKeyGen	2 Exp	3 Exp
	Delegate	5 Exp	2 Exp
	TrapGen	2 Exp + 2 Pairing	3 Exp + 1 Pairing
	Decrypt	1 Exp	1 Exp
Senders	Encrypt	2 Exp	4 Exp + 1 Pairing

IV. FUTURE RESEARCH

The main focus of future research in the field of the multi-user setting must be efficient. Existing multiuser searchable encryption scheme is not practical in real world applications and not scalable for large datasets. Only the single sender and single receiver is efficient. Moreover one of the goals offuture work should be the reduction of the computational complexity. To achieve that, the use of different, more efficient primitives or different data representations like trees or inverted index.It should focus on improving the query expressiveness and mainly the efficiency/scalability of searchable encryption scheme in multiuser setting.Research on query expressiveness needs to move forward for closing the gap between existing searchable encryption schemes and plaintext searches. It includes but not limited to functionalities like phrase search, or regular expressions.

V. CONCLUSION

Significant progress in the field of Public Key Encryption with Keyword Search scheme allows users to search over encrypted data without compromising any information to anyone has been made. In this paper, study of six important schemes and analyze their security and efficiency. In addition, concluded seven security requirements that must satisfy as developing PEKS/SCF-PEKS/PKEDS scheme.This paper can help the development of public key encryption with keyword search schemes and it extend the complexity issues that can be rapidly developed.

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] 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.

[6] L. Fang, W. Susilo, C. Ge, and J. Wang, “A se- cure 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.

[7] P. Golle, J. Staddon, and B. Waters, “Secure con- junctive 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.

[8] C. Gu and Y. Zhu, “New efficient searchable encryp- tion schemes from bilinear pairings,” International Journal of Network Security, vol. 10, no. 1, pp. 25– 31, 2010.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

[9] 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 Communica- tions in Computer and Information Science, pp. 131- 136, Wuhan, China, 2011. Springer Berlin/ Heidelberg

[10] 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, Hei- delberg, 2007.

[11] 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.

[12] 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.

[13] 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, Van- couver, BC, 2009. IEEE computer society.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] W. C. Yau, S. H. Heng, and B. M. Goi, "Off-line key- word 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.

[20] 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 Depend- able Applications, vol. 3, no. 1/2, pp. 72–81, 2012.

BIOGRAPHY

PUNEET DHEER is a Post Graduate in the Computer Science and Engineering Department, SRM UNIVERSITY, Tamil Nadu. He received Master of Technology (M.Tech) degree in 2016 from SRM UNIVERSITY, Tamil Nadu, India. His research interests are Network Security, Artificial Intelligence, and Application of Machine Learning.