A Secure Trapdoor-Indistinguishable Public Encryption Scheme with Keyword Search for Cloud Storage Service

Cheng-Chi Lee, Chun-Ta Li, and Shih-Ting Chiu

***Abstract*—Cloud storage allows users to easily access their data in cloud anytime and anywhere by using any device that can get online, such as a wireless PDA, a smartphone, or a notebook computer. Nevertheless, how can we make sure that this simple access to cloud storage comes at a satisfactory security level? Keyword search with data encryption seems to be a good answer. Recently, Zhao *et al.* proposed a trapdoor-indistinguishable public key encryption scheme with keyword search to be applied to the field of cloud storage service. However, we found a weakness in Zhao *et al.*'s scheme. In this paper, we shall point out the weakness and offer an improved version of trapdoor-indistinguishable public key encryption with keyword search for cloud environments. In our improved scheme, we make the keyword trapdoor indistinguishable while protecting the PEKS ciphertext against forgery attacks. Compared with other PEKS schemes, our new design is not only more efficient but gives better performance in terms of correctness and security.**

***Index Terms*—Cloud storage, keyword search, PEKS, security, searchable encryption.**

I. Introduction

Cloud computing refers to both the applications delivered as services over the Internet and the hardware as well as systems software in the data centers that provide those services [1]. Cloud storage is one of the most popular applications served by the cloud. Nowadays, more and more people and businesses keep their data in the cloud. Thanks to the cloud storage service, with a tiny, lightweight device such as a wireless PDA, smartphone or notebook in their hands, users can readily access their data anytime and anywhere. As cloud storage technologies advance, the security of the data stored in cloud environments becomes a more and more important issue. To keep any malicious party from accessing and making use of the data stored in the cloud, data owners often need to encrypt the data before uploading them to the cloud server. In that case, when a legal user wishes to access the data stored in the cloud, he/she will have to download the data as a whole instead of picking out and downloading only the relevant part or parts. For example, let's suppose both Alice and Bob are legal users of some specific data. Alice stored the data in the cloud, and Bob wants to access some information about "computer". Bob has no choice but to download all the data stored in the cloud before he can sort out the parts of the data that are actually related to "computer". The downloading of the whole pack of data can be a real waste of time and resources especially when the data stored in the cloud are in very large quantities while only very small portions of them need to be accessed. To retrieve only the part or parts of the data that the user really needs, keyword search seems to be a good solution.

However, if the uploaded data in the cloud has been encrypted by the data owner, then how can we make keyword search work? In 2000, Song *et al.* [2] proposed a secure keyword search scheme using a symmetric cipher. In 2004, in their well-celebrated article entitled "Public Key Encryption with Keyword Search", Boneh *et al*. [3] went a step further and offered a scheme later often referred to as PEKS. Boneh *et al*.'s PEKS scheme has a secure channel between the cloud server and the user. In 2008, in order to reduce the cost, Baek *et al.* extended Boneh *et al.*'s PEKS scheme into a secure-channel-free public key encryption scheme with keyword search (SCF-PEKS) [4]. However, in 2009, Rhee *et al.* pointed out that Baek *et al.*'s SCF-PEKS was vulnerable to the keyword guessing attack [5], and so they proposed the concept of trapdoor indistinguishability [6]. On the other hand, Liu *et al*. proposed an efficient privacy-preserving keyword search scheme (EPPKS) [7] which improved the performance of PEKS. Meanwhile, in 2010, Li *et al*. [8] proposed a fuzzy keyword search scheme based on keyword similarity semantics capable of responding with the closest possible matching files. In 2012, Liu *et al.* [9] improved their earlier work EPPKS and proposed a secure and privacy-preserving keyword search (SPKS) scheme. Besides, Zhao *et al*. [10] also proposed a trapdoor-indistinguishable public key encryption scheme with keyword search that does not require a secure channel between the receiver and the server. In addition to the researches mentioned above, quite a number of studies can be found in the literature concerned that focus on the quest for PEKS and keyword search with high efficiency and security [11]-[18].

Although PEKS schemes do enable users to get to the data they wish to access, how to make that happen in cloud environments with privacy fully protected is an important research issue. In 2013, Hsu *et al.* [14] made a list of some security requirements to be met in cloud computing environments as follows:

Manuscript received March 6, 2019; revised June 12, 2019. This work was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C., under Grant MOST 107-2410-H-165-001.

Cheng-Chi Lee and Shih-Ting Chiu are with the Department of Library and Information Science, Fu Jen Catholic University, New Taipei City 24205, Taiwan, R.O.C. and with the Department of Photonics and Communication Engineering, Asia University, Taichung City 41354, Taiwan, R.O.C. (e-mail: cclee@mail.fju.edu.tw, shihting0404@hotmail.com).

Chun-Ta Li is now with the Department of Information Management, Tainan University of Technology, Tainan City 71002, Taiwan, R.O.C. (e-mail: th0040@mail.tut.edu.tw).

doi: 10.18178/jacn.2019.7.1.263

A. User Authentication

The CSP (Cloud Service Provider) needs to confirm that the trapdoor of the keyword is sent from the authorized user and no one can discover the authorized user's real identity except for the CSP.

B. Authentication of Data Owner

When the CSP receives the ciphertext from the data owner, in order to avoid having fake ciphertext stored, the CSP needs to authenticate that the ciphertext is sent from the real data owner.

C. Protection of Authorized Identity

In case an attacker has the trapdoor ciphertext intercepted on the way from the data owner to the CSP, the attacker cannot derive the user's identity from the intercepted trapdoor ciphertext.

D. Trapdoor Indistinguishability

Due to the fact that the trapdoor ciphertext is sent via a public channel, an attacker may intercept the trapdoor ciphertext and try to figure out the real keyword. Trapdoor indistinguishability is the kind of protection that ensures no malicious attacker can obtain the information hidden in the trapdoor ciphertext by analyzing the trapdoor ciphertext.

E. Resistance to Keyword-Guessing Attack

The trapdoor is frequently updated, and that is why it is said to be indistinguishable. With the trapdoor collected, an attacker still cannot offline/online guess the real keyword from the trapdoor.

The PEKS schemes currently available can indeed provide user authentication and identity protection. However, there is not a mechanism to keep the CSP from storing fake ciphertext. Fig. 1 shows a scenario where the data owner intends to send the data's ciphertext and PEKS ciphertext to the CSP, but both pieces of ciphertext get intercepted by an attacker. The attacker then sends some fake ciphertext to the CSP. When receiving the fake ciphertext, without verifying the validity of the data owner, the CSP stores them as always so that the data can be searched and retrieved by users. Later on, when a legal user needs to access some data which can be directed to by a certain keyword, he/she creates a trapdoor for that keyword and sends it to the CSP. Since the CSP stored the wrong ciphertext, the server fails to retrieve the correct data. Finally, the user cannot get the due ciphertext to decrypt.

To mend this flaw, in this paper, we propose a secure trapdoor-indistinguishable public key encryption scheme with keyword search for cloud storage that satisfies the following requirements:

- There is no need for a secure channel between the cloud user and the cloud service provider (CSP). In other words, the trapdoor can be sent via a public channel.
- The trapdoor is indistinguishable. Even though an attacker can intercept the trapdoor, he/she still has no way to derive the real keyword by analyzing the trapdoor.
- The CSP can search through the ciphertext for keywords. The CSP can check whether or not the data contains certain keywords specified by the user without knowing the keywords and the content of the data.
- The CSP can verify whether the PEKS ciphertext is sent from the data owner and thereby avoid the forgery attack.

The rest of this paper is organized as follows: In Section II, we shall review Zhao *et al*.'s trapdoor-indistinguishable public key encryption scheme with keyword search and point out the weakness of the scheme we have found. In Section III, we shall present an improved version of the scheme with the security flaw mended. In Section IV, we shall analyze the improved scheme in terms of security and performance. Finally, the conclusion will be drawn in Section V.

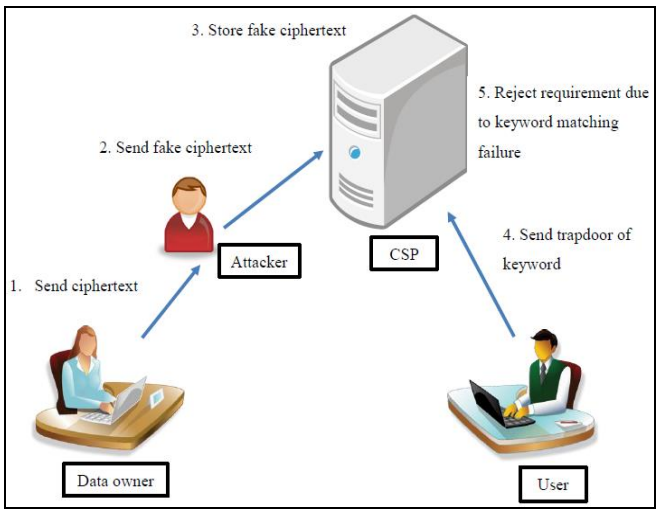

Fig. 1. PEKS without data owner authentication.

II. Preliminaries

In this section, we will quickly introduce the bilinear pairing technique [19] as well as some complexity assumptions and review the trapdoor-indistinguishable public key encryption scheme with keyword search (TI-PEKS) by Zhao *et al*. [10].

A. Bilinear Pairing

Let $\mathbb{G}_1$ be a cyclic additive group with prime order q and $\mathbb{G}_2$ be a cyclic multiplicative group with prime order q, and suppose P is the generator of group $\mathbb{G}_1$. With $x, y \in \mathbb{Z}_q$ and bilinear map $e: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, there are some properties as follows:

1. Bilinearity: For all $x, y \in \mathbb{Z}_q$ and $R, Q \in \mathbb{G}_1$, $e(xR, yQ) = e(R, Q)^{xy}$.
2. Computability: For any $R, Q \in \mathbb{G}_1$, there exists an efficient algorithm to compute $e(R, Q) \in \mathbb{G}_2$.
3. Non-degeneration: $e(R, Q) \neq 1$.

B. Complexity Assumptions

Some complex problems can be created out of $\mathbb{G}_1$ as follows:

1. Discrete Logarithm Problem (DLP): Given two elements R and Q in $\mathbb{G}_1$, it is difficult to find $n \in \mathbb{Z}_q$ such that $R = nQ$ if n exists.

2. Computation Diffie-Hellman Problem (CDHP): Given R, xR, yR for $x, y \in \mathbb{Z}_q$, it is difficult to compute xyR.
3. Bilinear Diffie-Hellman Problem (BDHP): Given R, R^x, R^y, R^z for $x, y, z \in \mathbb{Z}_q$, it is difficult to compute $e(R,R)^{xyz} \in \mathbb{G}_2$.

C. Trapdoor-Indistinguishable Public Key Encryption with Keyword Search

In this subsection, we will review Zhao *et al.*'s trapdoor-indistinguishable public key encryption scheme with keyword search. In Zhao *et al.*'s TI-PEKS, there are three parties involved, namely the sender, the server, and the receiver. The scheme contains six algorithms as follows:

1. $KeyGen_{Param}(k)$: A common parameter generation algorithm. With a security parameter $k \in \mathbb{N}$ entered, the algorithm outputs the system's common parameter cp.
2. $KeyGen_{Server}(cp)$: The public/private key generation algorithm for the server. It takes in the common parameters cp and outputs the public key pk_S and the private key sk_S for the server.
3. $KeyGen_{Receiver}(cp)$: The public/private key generation algorithm for the receiver. With the common parameters cp taken in, the algorithm outputs the public key pk_R and the private key sk_R for the receiver.
4. $PEKS(cp, pk_S, pk_R, w)$: The generation algorithm of the ciphertext's PEKS R. The data owner inputs the system's common parameters cp, server's public key pk_S, receiver's public key pk_R, as well as the keyword w, and then the algorithm outputs the ciphertext's PEKS R that is searchable.
5. $Trapdoor(cp, sk_R, w)$: The trapdoor generation algorithm. The receiver inputs the system's common parameters cp, his/her private key sk_R, as well as the keyword w, and then the algorithm generates the trapdoor TW of the keyword w.
6. $Test(cp, TW, sk_S, R)$: The keyword test algorithm. Input the system's common parameters cp, the server's public key sk_S, the ciphertext's PEKS R and the trapdoor TW of the keyword w, and the algorithm will return "correct" if $w' = w$ and "incorrect" otherwise.

III. The Proposed Scheme

In this section, we shall first illustrate the architecture of our improved TI-PEKS scheme and then give the details of each step.

A. Architecture of Our Scheme

In our improved scheme, there are 8 steps to take, namely system parameter generation, key generation for cloud service provider (CSP), key generation for user, key generation for data owner, PEKS ciphertext generation, ciphertext verification, keyword trapdoor generation, and search. Three participants are involved, including the data owner, who generates the data's ciphertext and PEKS ciphertext and sends them to the CSP; the CSP, who provides the storage, stores the data and searches the data for the specific parts that the user requests; and the user, who wishes to retrieve certain parts of the data that contain a specific keyword and therefore sends the keyword's trapdoor to the CSP. Fig. 2 is the flowchart of our scheme with the purpose each step serves specified:

1. $KeyGen_{Param}$: In this step, some security parameters will be input to the system, and the system will output the common parameters.
2. $KeyGen_{CSP}$: With the public parameter taken in as input, the system outputs the CSP's public key and private key.
3. $KeyGen_{User}$: With the public parameter and the user's identity entered as input, the system outputs the user's public key and private key.
4. $KeyGen_{Data\ owner}$: Taking in the public parameter and the data owner's identity as input, the system outputs the data owner's public key and private key.
5. $PEKS$: With the data encrypted, the data owner uses the common parameters and the user's public key to generate the keyword w's PEKS ciphertext. In addition, the data owner uses his/her private key to generate the verification message and sends the data's ciphertext, PEKS ciphertext and verification message to the CSP.
6. $Verify$: Upon receiving the encrypted data, the CSP uses the data owner's public key to verify whether the ciphertexts were actually sent by the data owner. If yes, the CSP stores the data; otherwise, the ciphertexts are rejected.
7. $Trapdoor$: When the user wants to retrieve some parts of the data that contain a certain keyword, he/she uses his/her private key and the CSP's public key to generate the keyword's trapdoor and sends it to the CSP.
8. $Test$: Upon retrieving the trapdoor, the CSP uses his/her private key and the user's public key to check whether the trapdoor is equal to the PEKS ciphertext sent from the data owner. If positive, the CSP sends the ciphertext to the user; otherwise, the CSP denies the request.

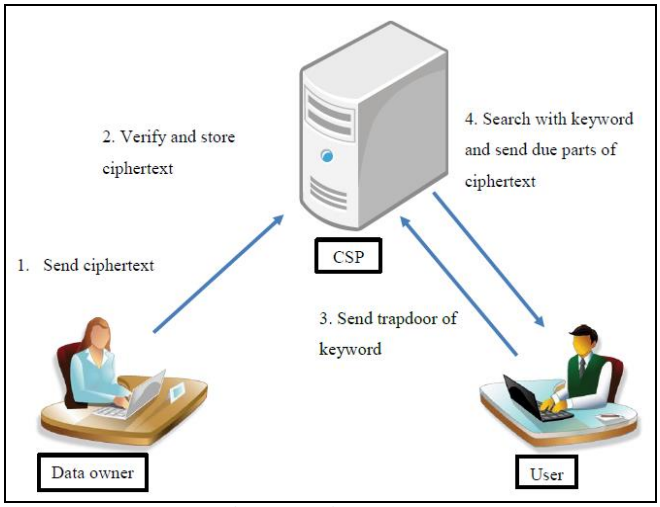

Fig. 2. The flowchart of our proposed scheme.

B. The Detailed Steps of Our Proposed Scheme

First of all, Table I lists the notations that will be used throughout our scheme. Then, each step that is to be taken in the scheme will be detailed.

TABLE I: SYMBOLS USED IN THE PAPER

Notation	Description
k	Security parameter, $k \in \mathbb{N}$
ID_O	Identity of data owner
ID_U	Identity of user
pk_s, sk_s	Public key and private key of CSP
pk_O, sk_O	Public key and private key of data owner
pk_U, sk_U	Public key and private key of CSP
w	Keyword
$\oplus$	XOR operation

1. $KeyGen_{Param}$: With a security parameter $k \in \mathbb{N}$ input, the system generates a group $\mathbb{G}_1$ of prime order $q \geq 2^k$, a random generator P of $\mathbb{G}_1$, and a bilinear map $e: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Three hash functions are produced, namely $H_0: \{0,1\}^* \to Z$, $H_1: \{0,1\}^* \to \mathbb{G}_1$ and $H_2: \mathbb{G}_2 \to \{0,1\}^k$. In addition, d_w denotes a description of the keyword space, and the common parameters are $cp = (q, \mathbb{G}_1, \mathbb{G}_2, e, P, H_0, H_1, H_2, d_w)$.
2. $KeyGen_{Server}$: Input the common parameters cp, choose a random number $x \in \mathbb{Z}_q^*$ and $Q \in \mathbb{G}_1^*$, and compute $X = xP$. Output the server's public key $pk_s = (cp, Q, X)$ and private key $sk_s = x$.
3. $KeyGen_{User}$: The CSP inputs the common parameters cp and the user's identity ID_U. Then the CSP computes $Y = H_0(ID_U)P$ and $y = xY$ and sends the public key $pk_U = (cp, Y)$ and private key $sk_U = y$ to the user.
4. $KeyGen_{Data\ owner}$: The CSP inputs the common parameters cp and the data owner's identity ID_O. Then the CSP computes $A = H_0(ID_O)P$ and $a = xA$ and sends the public key $pk_O = (cp, A)$ and private key $sk_O = a$ to the data owner.
5. $PEKS$: The data owner inputs $cp, pk_S, pk_U, sk_O, w,$ and A and chooses a random number $r \in \mathbb{Z}_q^*$. Then the data owner computes R as PEKS ciphertext, where $R = (U, V, t, V_O)$, $U = rP$, $V = rA$, $t = e(H_1(w), U)e(rQ, X)$, $V_O = H_1(t) \oplus H_1(\alpha)$ and $\alpha = e(Q, ra)$. The data owner sends R and the data's ciphertext to the CSP.
6. $Verify$: Upon receiving the data, the CSP inputs pk_S, pk_O, V_O, and V and computes $V_O' = H_1(t) \oplus H_1(e(xQ, V))$. The CSP checks whether V_O' is equal to V_O or not. If yes, the CSP stores the received data; otherwise, the CSP rejects the ciphertext.
7. $Trapdoor$: The user inputs cp, sk_R, w and Y and chooses a random number $\tilde{a} \in \{0,1\}^*$. Then the user computes $T_{w1} = [y^{-1}H_1(w) + H_1(\tilde{a})] \oplus [H_1(e(Q, \tilde{a}y))]$ and $T_{w2} = yH_1(\tilde{a}) \in \mathbb{G}_1$ and returns TW and $\tilde{a}\ Y$, where $TW = (T_{w1}, T_{w2})$, as a trapdoor for the keyword w.
8. $Test$: The CSP inputs $cp, TW, sk_S, R,$ and $\tilde{a}\ Y$ and computes $Tw = T_{w1} \oplus H_1(e(xQ, \tilde{a}Y))$, $S = e(T_{w2}, U)$, $t' = e(xQ, U)^{-1}$ and $T = tt' = e(H_1(w), U)$. If $H_2(e(T_w, V)) = H_2(T \cdot S)$, it returns "Correct"; otherwise, it returns "Incorrect".

IV. ANALYSIS OF OUR SCHEME

In this section, we shall show how our improved PEKS scheme compares with Boneh *et al.*'s [3], Beak *et al.*'s [4], Liu *et al.*'s [9], Rhee *et al.*'s [6], and Zhao *et al.*'s [10] in terms of security and performance. Then there will be a BAN logic [20], [21] correctness verification of the proposed scheme, followed by a security analysis.

A. Comparison

To begin with, let's evaluate the security of the proposed scheme by comparing it with a number of related schemes. Table II shows the comparison results, where abbreviations User Auth, Owner Auth, AuthID Pro, Trap Ind and KW Gue are used to represent user authentication, data owner authentication, authorized identity protection, trapdoor indistinguishability and resistance to keyword-guessing attack, respectively. As Table II reveals, the proposed scheme does reach a higher security level and is therefore more user-friendly.

TABLE II: SECURITY COMPARISON AMONG RELATED SCHEMES

	Boneh *et al.*'s	Beak *et al.*'s	Liu *et al.*'s	Rhee *et al.*'s	Zhao *et al.*'s	Ours
User Auth	○	○	○	○	○	○
Owner Auth	×	×	×	×	×	○
AuthID Pro	○	○	○	○	○	○
Trap Ind	×	×	×	○	○	○
KW Gue	×	×	×	○	○	○

TABLE III: PERFORMANCE COMPARISON AMONG RELATED SCHEMES

	Boneh *et al.*'s	Beak *et al.*'s	Liu *et al.*'s	Rhee *et al.*'s	Zhao *et al.*'s	Ours
PEKS/SCF-PEKS	1*P*+1*E*	1*P*+2*E* +1*M*	1*P*+1*E*	1*P*+1*E*	1*P*+2*E* +3*M*	3*P*+3*E* +3*M*
Verification	×	×	×	×	×	2*P*+1*E*
Trapdoor	1*P*	1*P*+1*M*	1*P*	2*P*	4*P*+1*E* +3*M*	4*P*+1*E* +2*M*
Test	1*P*+1*E*	1*E*+1*M*	1*E*	1*P*+1*E*	1*P*+4*E* +2*M*	1*P*+4*E* +2*M*

Since PEKS ciphertext generation, data owner verification, trapdoor generation, and keyword test are the four major parts of a secure PEKS scheme and should be performed in each session, we only took the computation costs of these four steps into consideration when comparing our improved scheme with the others in terms of performance. Table III shows the comparison results, where simplified expressions such as PEKS, Verification, Trapdoor, and Test are used to represent PEKS ciphertext generation, data owner verification, trapdoor generation, and keyword test, respectively. In addition, P denotes a map-to-point hash function operation, E denotes a pairing operation, and M denotes a multiplication operation. As Table III reveals, Liu *et al.*'s PEKS scheme is the most efficient of them all. However, Liu *et al.*'s scheme, as well as Boneh *et al.*'s and Beak *et al.*'s, does not satisfy the trapdoor indistinguishability requirement. On the other hand, although our improved scheme requires more computation in PEKS and Test, in

Trapdoor it costs less than Zhao *et al*.'s scheme. Considering the fact that our improved scheme offers an obviously higher level of security with data owner authentication, trapdoor indistinguishability and resistance to keyword-guessing attack all covered, we find the slight extra computation in PEKS and Test pays off well.

B. Correctness Analysis

The BAN logic is a well-accepted method to analyze the correctness of cryptographic protocols. In this subsection, we will have some notations, goals and assumptions defined and then use the BAN logic [20], [21] to verify the correctness of our scheme.

1)Notations

Let's take a quick look at the syntax and notations of the BAN logic. First, we have A and B that denote two specific participators, X stands for a formula (statement), and $\overset{K_A}{\longmapsto} A$, $\overset{K_B}{\longmapsto} B$, K_A^{-1} and K_B^{-1} are A's and B's public key and secret key, respectively. There are some rules as follows [1, 20]:

(i) $A|\equiv X$ means A believes that formula X is ture.

(ii) $A|\equiv B$ means A believes B's action.

(iii) $A|\Rightarrow X$ means A has complete control over formula X.

(iv) $A \lhd X$ means A holds or sees formula X.

(v) $\#(X)$ means formula X is fresh or has not been used before.

(vi) $\overset{K_A}{\longmapsto} A$ means K is the public key for A and K_A^{-1} is the private key for A.

(vii) $\frac{Rule\ 1}{Rule\ 2}$ means $Rule\ 2$ can be derived from $Rule\ 1$.

2)Goals

With three roles involved, namely the data owner ($Owner$), the cloud service provider (CSP) and the user ($User$), in our scheme, there are two goals to be achieved: in the data owner verification process, CSP is to believe that $Owner$ has the private key to create the PEKS ciphertext; in the keyword search process, CSP is to believe that $User$ has the private key to create the trapdoor of the keyword. These two goals of our scheme can be rephrased in the language of the BAN logic as follows:

$$G1.\ CSP|\equiv Owner \lhd K_{owner}^{-1}$$
$$G2.\ CSP|\equiv User \lhd K_{User}^{-1}$$

3)Assumptions

To analyze the correctness of our scheme, there are some assumptions as follows:

$$A1.\ CSP|\equiv \overset{K_{owner}}{\longmapsto} Owner$$
$$A2.\ CSP|\equiv \overset{K_{User}}{\longmapsto} User$$
$$A3.\ Owner|\equiv \overset{K_{CSP}}{\longmapsto} CSP$$
$$A4.\ Owner|\equiv \overset{K_{User}}{\longmapsto} User$$
$$A5.\ User|\equiv \overset{K_{CSP}}{\longmapsto} CSP$$
$$A6.\ CSP|\Rightarrow K_{CSP}^{-1}$$
$$A7.\ CSP|\Rightarrow K_{owner}^{-1}$$
$$A8.\ CSP|\Rightarrow K_{User}^{-1}$$

4)Verification of the data owner

With the goals and assumptions confirmed, now we can analyze the correctness of our data owner verification process with the BAN logic. The details are as follows:

Message 1: $Owner \longrightarrow CSP: R = (U, V, t, V_0, rA)$

$$V1.\ CSP \lhd R$$
$$V2.\ \frac{CSP \lhd R, CSP \lhd K_{CSP}^{-1}, CSP \lhd rA}{CSP \lhd V_0'}$$
$$V3.\ \frac{CSP|\Rightarrow K_{CSP}^{-1}, CSP \lhd V_0'}{CSP|\equiv V_0}$$
$$V4.\ \frac{CSP|\equiv V_0}{CSP|\equiv Owner \lhd K_{owner}^{-1}}$$

Finally, we can infer from formula $V4$ that our scheme does achieve the goal we set up. In the end, CSP does believe that $Owner$ has the private key to create the PEKS ciphertext.

5)Verification of the user

Now we analyze the correctness of our user verification process with the BAN logic as follows:

Message 1: $User \longrightarrow CSP: TW = (T_{w1}, T_{w2})$ and $\tilde{a}\ Y$.

$$V1.\ CSP \lhd TW, \tilde{a}\ H_1(ID_U)$$
$$V2.\ \frac{CSP \lhd T_{w1}, CSP \lhd K_{CSP}^{-1}, CSP \lhd \tilde{a}\ Y}{CSP \lhd T_w}$$
$$V3.\ \frac{CSP \lhd (T_w, T_{w1}, T_{w2}, U, t)}{CSP \lhd (S, t', T)}$$
$$V4.\ \frac{CSP|\equiv (T_w, V, T), CSP|\equiv K_{CSP}^{-1}}{CSP|\equiv S}$$
$$V5.\ \frac{CSP|\equiv S}{CSP|\equiv T_{w2}}$$
$$V6.\ \frac{CSP|\equiv T_{w2}}{CSP|\equiv User \lhd K_{User}^{-1}}$$

Finally, we can infer from formula $V6$ that our scheme does achieve the goal we set up. In the end, CSP does believe that $User$ holds the private key to create the trapdoor of the keyword.

C. Security Analysis

In this subsection, we shall analyze the proposed scheme to see if it satisfies the following security requirements:

1. Only the CSP can use the keyword created by the data owner to do keyword search: If an attacker captures the PEKS ciphertext $R = (U, V, t, V_0, rH_1(ID))$ through the communication channel between the data owner and the CSP and captures the trapdoor of keyword $TW = (T_{w1}, T_{w2})$ through the communication channel between the user and the CSP, he/she still cannot compute $H_1(e(xQ, \tilde{a}H_1(ID_U)P))$, $H_1(e(Q, \tilde{a}y)$ and $t' = e(xQ, U)^{-1}$ from the captured (*R*, *TW*) because to do that is as difficult as to solve the BDH problem. In other words, only the CSP, who owns the private key, can determine whether the trapdoor of the keyword is truly sent from the user by confirming it against what the data owner set up.
2. The trapdoor of the keyword is indistinguishable: In our scheme, since the random string $\tilde{a}$ chosen by the user differs from session to session, a keyword cannot generate the same trapdoor a second time. In other words, the trapdoor of the same keyword will be changed in every session. This way, even if an attacker captures the trapdoor in a given session, the captured trapdoor

still cannot be used to come by the keyword in any following session.

3. The CSP can determine that the PEKS ciphertext is sent by an authorized data owner: Only an authorized data owner have A and a, both generated by the CSP. The authorized data owner can use A and a to generate the PEKS ciphertext and authentication information. Upon receiving the message, the CSP can utilize its private key to determine whether the PEKS ciphertext is truly sent by the authorized data owner.
4. Only the CSP can verify the user's identity: Even if an attacker captures the data delivered through the communication channel between the user and the CSP, there is still no way the attacker can analyze the information and get to know the user's identity. Only the CSP can verify the user's identity by testing the received data against the values the CSP holds.

V. Conclusions

Since Boneh *et al*. offered their concept of public key encryption with keyword search (PEKS), many researchers have extended it to various PEKS schemes such as the secure channel-free public key encryption scheme with keyword search (SCF-PEKS), the efficient privacy-preserving keyword search scheme (EPPKS), the trapdoor-indistinguishable public key encryption scheme with keyword search (TI-PEKS) and so on. In this paper, we have proposed a secure trapdoor-indistinguishable public key encryption scheme with keyword search. Using a public channel, the proposed scheme is capable of keeping the CSP from being tricked by an attacker sending in fake ciphertext.

References

[1] A. Weiss, "Computing in the clouds," *Networker*, vol. 11, no. 4, pp. 16-25, 2007.

[2] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. the 2000 IEEE Symposium on Security and Privacy*, 2000, pp. 44-55.

[3] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. 2004 International Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'04)*, 2004, vol. 3027, pp. 506-522.

[4] J. Baek, R. Safiavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *Proc. the 8th International Conf. on Computational Science and Its Applications (ICCSA'08)*, 2008, vol. 5072, pp. 1249-1259.

[5] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Improved searchable public key encryption with designated tester," in *Proc. of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS '09)*, 2009, pp. 376-379.

[6] 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.

[7] Q. Liu, G. Wang, and J. Wu, "An efficient privacy preserving keyword search scheme in cloud computing," in *Proc. of the 2009 International Conf. on Computational Science and Engineering*, 2009, pp. 715-720.

[8] J. Li, Q. Wang, C. Wang, K. R. N. Cao, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *Proc. of the 29th conf. on Information Communications (INFOCOM'10)*, San Diego, California, USA, 2010, pp. 1-5.

[9] Q. Liu, G. Wang, and J. Wu, "Secure and privacy preserving keyword searching for cloud storage services," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 927-933, 2012.

[10] 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.

[11] Z. Chen, C. Wu, D. Wang, and S. Li, "Conjunctive keywords searchable encryption with efficient pairing, constant ciphertext and short trapdoor," *Lecture Notes in Computer Science*, vol. 7299, pp. 176-189, 2012.

[12] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," *Lecture Notes in Computer Science*, vol. 3089, pp. 31-45, 2004.

[13] 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.

[14] S. T. Hsu, C. C. Yang, and M. S. Hwang, "A study of public key encryption with keyword search," *International Journal of Network Security*, vol. 15, no. 2, pp. 71-79, 2013.

[15] 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.

[16] C. C. Lee, S. T. Hsu, and M. S. Hwang, "A study of conjunctive keyword searchable schemes," *International Journal of Network Security*, vol. 15, no. 5, pp. 311-320, 2013.

[17] D. J. Park, K. Kim, and P. J. Lee, "Public key encryption with conjunctive field keyword search," *Lecture Notes in Computer Science*, vol. 3325, pp. 73-86, 2005.

[18] E. K. Ryu and T. Takagi, "Efficient conjunctive keyword-searchable encryption," in *Proc. of the 21st International Conf. on Advanced Information Networking and Application Workshops (AINAW '07)*, 2007, vol. 1, pp. 409-414.

[19] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *Lecture Notes in Computer Science*, vol. 2139, pp. 213-229, 2001.

[20] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM* Transactions *Computer Systems*, vol. 8, no. 1, pp. 18-36, 1990.

[21] Jan Wessels, "Application of BAN-logic," *CMG FINANCE B.V.*, vol. 19, 2001.

Cheng-Chi Lee received the Ph.D degree in computer science from National Chung Hsing University (NCHU), Taiwan, in 2007. He is currently a professor with the Department of Library and Information Science at Fu Jen Catholic University. His current research interests include data security, cryptography, network security, mobile communications and computing, wireless communications.

Chun-Ta Li received the Ph.D degree in computer science and engineering from National Chung Hsing University (NCHU), Taiwan, in 2008. He is currently an associate professor with the Department of Information Management at Tainan University of Technology, Taiwan. His research interests include information and network security, wireless sensor networks, mobile computing, and security protocols for RFID, IoTs and cloud computing.

Shih-Ting Chiu received the degree of B.S. and M.S. in library and information science from Fu Jen Catholic University, Taiwan, in 2013 and 2015, respectively. Her current research interests include information security, cryptography, and mobile communications.