Security Improvement Against Malicious Server's Attack for a dPEKS Scheme

Wang BingJian, Chen TzungHer, and Jeng FuhGwo, *Member, IACSIT*

***Abstract*—While the original public-key encryption with keyword search scheme (PEKS) has been pointed out to be insecure against off-line keyword-guessing attacks, some public-key encryption with designated tester schemes (dPEKS) proposed recently also encounter the same attacks. Rhee et al. proposed a dPEKS which is intended to prevent the off-line keyword-guessing attacks. However, we find that the off-line keyword-guessing attacks are still working in the test phase when some malicious servers exist. Hence, we add a random parameter into the test phase of Rhee et al.'s scheme to get a more secure and improved dPEKS scheme so as to prevent from keyword-guessing attacks and to benefit the advantages of dPEKS as well.**

***Index Terms*—Searchable encryption, designated tester, data security.**

I. Introduction

To protect the confidentiality of sensitive data in clouding-computing environments, a reliable encryption technology is used to encrypt the sensitive data stored in the server. For a user issuing a keyword searching on the encrypted data, the server unavoidably faces the security problem of how to process the search without revealing any sensitive information. Especially, the server maintaining the database of encrypted data is not trusty.

The public-key encryption with keyword search scheme (PEKS) is first proposed by Boneh et al. (2004) [1]. Based on Boneh et al.'s scheme, Hwang and Lee (2007) [2] proposed another PEKS scheme for multi-receiver. The concept of proxy re-encryption is applied in keyword search by Shao et al. (2010) [3] and by Yau and Phan (2010) [4] as well. Recently, a conjunctive subset keywords search is proposed by Zhang et al. (2011) [5].

However, Baek et al. (2006) [6] pointed out that an outside attacker in the PEKS scheme could perform the test process by collecting the transmitted ciphertexts and trapdoors. Thus the attacker could further construct the relationship between encrypted data and the given trapdoors of known keywords. Therefore, Baek et al. [6] proposed their public-key encryption scheme with designated tester (dPEKS) to solve the problem. In Baek et al.'s scheme, the keyword encryption function includes the server's public key such that, in the test processing, the server's private key is needed. Hence, their scheme equips the property of designated tester.

In the same year, Bynum et al. (2006) [7] pointed out that the design of trapdoors in PEKS scheme was insecure against off-line keyword-guessing attacks. Because an attacker can choose a keyword to test whether the captured trapdoor includes the guessed keyword with the receiver's public key and bilinear map operation, the interested keyword of the receiver is revealed. Unfortunately, although Beak et al.'s dPEKS scheme [6] achieves tester designating, the trapdoor's structure is the same with that in PEKS's. Inheritably, their scheme cannot prevent off-line keyword-guessing attacks.

Therefore based on Beak et al.'s dPEKS scheme [6], Rhee et al. (2010) [8] enhanced the trapdoor security so as to prevent from off-line keyword-guessing attacks. The enhanced scheme keeps the property of designated tester and redefines the trapdoor function. The redefined trapdoor includes the server's public key and a random parameter. That is, before doing test, the server's private key is also needed, and the random parameter makes the off-line keyword-guessing attacks of outside attacker impossible.

Rhee et al. [8] claimed that their dPEKS scheme with a new trapdoor function was secure against keyword-guessing attacks. Yet, we would like to point out their trapdoor design was still on the risk of keyword-guessing attacks especially by malicious servers in this paper. In this paper, the authors present an improved dPEKS scheme to avoid this drawback.

II. Cryptanalysis of Rhee Et Al.'s Dpeks Scheme

In this section, we would like to review the Rhee et al.'s dPEKS scheme [8] first and then try to point out their security problems.

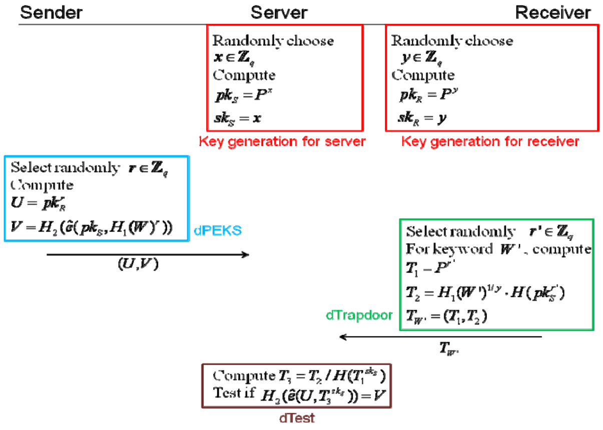

Sender
Server
Receiver
Randomly choose $x \in \mathbb{Z}_q$
Compute $pk_S = P^x$
$sk_S = x$
Key generation for server
Randomly choose $y \in \mathbb{Z}_q$
Compute $pk_R = P^y$
$sk_R = y$
Key generation for receiver
Select randomly $r \in \mathbb{Z}_q$
Compute $U = pk_R^r$
$V = H_2(\hat{e}(pk_S, H_1(W)^r))$
dPEKS
(U,V)
Select randomly $r' \in \mathbb{Z}_q$
For keyword W', compute
$T_1 - P^{r'}$
$T_2 = H_1(W')^{1/y} \cdot H(pk_S^{r'})$
$T_{W'} = (T_1, T_2)$
dTrapdoor
$T_{W'}$
Compute $T_3 = T_2 / H(T_1^{sk_S})$
Test if $H_2(\hat{e}(U, T_3^{sk_S})) = V$
dTest

Fig. 1. The process of Rhee et al.'s scheme

Manuscript received October 18, 2011; revised October 30, 2011. This work was supported in part by National Science Council under Grant No. NSC 99-2221-E-415-017 and NSC 99-2221-E-415-008.

B. J. Wang and T. H. Chen are with Department of Computer Science and Information Engineering, National Chiayi University, No.300 Syuefu Rd., Chiayi City 60004, Taiwan (R.O.C.) (e-mail: brucewang24@yahoo.com.tw, thchen@mail.ncyu.edu.tw)

F. G. Jeng is with Department of Applied Mathematics, National Chiayi University, No.300 Syuefu Rd., Chiayi City 60004, Taiwan (R.O.C.) (e-mail: fgjeng@mail.ncyu.edu.tw).

A. Rhee Et Al.'S Dpeks Scheme

1) Global setup

Determine two cyclic groups G_1 and G_2 with prime order p, and their admissible bilinear paring function $\hat{e}: G_1 \times G_1 \rightarrow G_2$. Define three hash functions as $H:\{0,1\}^* \rightarrow G_1$, $H_1:\{0,1\}^* \rightarrow G_1$ and $H_2: G_2 \rightarrow \{0,1\}^{\lambda}$, where λ is a security parameter. Choose a random generator $P \in G_1$.

2) Key generation for server and receiver

The server (resp. receiver) generates his private key by randomly choosing $sk_S = x \in \mathbb{Z}_p$ (resp. $sk_R = y \in \mathbb{Z}_p$) and the corresponding public key by computing $pk_S = P^x$ (resp. $pk_R = P^y$).

1) dPEKS: $(pk_R, pk_S, W) \rightarrow (U, V)$

The sender adopts receiver's and server's public keys, pk_R and pk_S, to compute the dPEKS cipher text by $(U,V) = (pk_R{}^r, H_2(\hat{e}(pk_S, H_1(W)^r))$, where W is keyword, and $r \in \mathbb{Z}_p$ is randomly chosen. Then the cipher text (U,V) is sent to server for receiver's search later.

3) dTrapdoor: $(pk_S, sk_R, W') \rightarrow T_{W'}$

When the receiver intends to process the search for keyword W', he has to generate a trapdoor for the keyword by computing $T_{W'} = (T_1, T_2) = (P^{r'}, H_1(W')^{1/y} \cdot H(pk_S{}^{r'}))$, where $r' \in \mathbb{Z}_p$ is randomly chosen. Then the receiver sends $T_{W'}$ to the server for searching process.

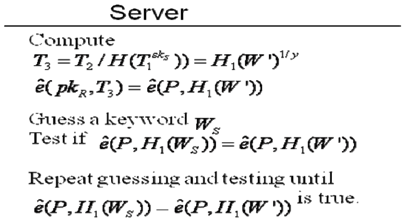
4) dTest: $((U,V), sk_S, T_{W'}) \rightarrow Boolean$

After the server receives the trapdoor $T_{W'}$ from the receiver, the server is able to test whether the keyword W' exists in some cipher text (U,V) or not. First, the server computes $T_3 = T_2 / H(T_1^{sk_S}))$. Second, the server checks if $H_2(\hat{e}(U, T_3^{sk_S}))$ is equal to V. If yes, the server sends the search result to the receiver.

The process of Rhee et al.'s scheme is illustrated in Fig. 1.

B. Security Problem

In the test phase of Rhee et al.'s dPEKS scheme, the server can compute $T_3 = T_2 / H(T_1^{sk_S})) = H_1(W')^{1/y}$. Accordingly, we found that a malicious server can forwardly compute $\hat{e}(pk_R, T_3) = \hat{e}(P^y, H_1(W')^{1/y}) = \hat{e}(P, H_1(W'))$. Then the malicious server can perform a keyword-guessing attack with $\hat{e}(P, H_1(W'))$ to guess which keyword the receiver is interested in. The process of malicious server's keyword guessing attacks is illustrated in Fig. 2.



Server

Compute

$T_3 = T_2 / H(T_1^{sk_S})) = H_1(W')^{1/y}$

$\hat{e}(pk_R, T_3) = \hat{e}(P, H_1(W'))$

Guess a keyword W_S

Test if $\hat{e}(P, H_1(W_S)) = \hat{e}(P, H_1(W'))$

Repeat guessing and testing until

$\hat{e}(P, II_1(W_S)) - \hat{e}(P, II_1(W'))$ is true.

Fig. 2. The process of malicious server's keyword guessing attacks

III. The Proposed Scheme

In this section, the improvement of Rhee et al.'s scheme is describe as follows.

To prevent the risk of keyword-guessing attacks from a malicious server, receiver's public key was redefined, and a random number u is introduced into the redefined public key computed by the receiver.

1) Global setup

Determine two cyclic groups G_1 and G_2 with prime order p, and their admissible bilinear paring function $\hat{e}: G_1 \times G_1 \rightarrow G_2$. Define three hash functions as $H:\{0,1\}^* \rightarrow G_1$, $H_1:\{0,1\}^* \rightarrow G_1$ and $H_2: G_2 \rightarrow \{0,1\}^{\lambda}$, where λ is a security parameter. Choose a random generator $P \in G_1$.

2) Key generation for server

The server generates his private key by randomly choosing $sk_S = x \in \mathbb{Z}_p$ and the corresponding public key by computing $pk_S = P^x$.

3) Key generation for receiver

The receiver generates his private key by randomly choosing $sk_R = y \in \mathbb{Z}_p$ and computes his public key by $pk_R = (pk_{R1}, pk_{S1}) = (P^{uy^2}, pk_S^u) = (P^{uy^2}, P^{xu})$, where $u \in \mathbb{Z}_p$ is random.

4) dPEKS: $(pk_{R1}, pk_{S1}, W) \rightarrow (U, V)$

The sender adopts pk_{R1} and pk_{S1} to compute the dPEKS ciphertext by $(U,V) = (pk_{R1}^r, H_2(\hat{e}(pk_{S1}, H_1(W)^r))$, where W is keyword, and $r \in \mathbb{Z}_p$ is randomly chosen. Then the ciphertext (U,V) is sent to server for receiver's search later.

5) dTrapdoor: $(pk_S, sk_R, W') \rightarrow T_{W'}$

When the receiver intends to process the search for keyword W', he has to generate a trapdoor for the keyword by computing $T_{W'} = (T_1, T_2) = (P^{r'}, H_1(W')^{1/sk_R^2} \cdot H(pk_S{}^{r'}))$, where $r' \in \mathbb{Z}_p$ is randomly chosen. Then the receiver sends $T_{W'}$ to the server for searching process.

6) dTest: $((U,V), sk_S, T_{W'}) \rightarrow Boolean$

After the server receives the trapdoor $T_{W'}$ from the receiver, the server is able to test whether the keyword W' exists in some ciphertext (U,V) or not. First, the server computes $T_3 = T_2 / H(T_1^{sk_S})$. Second, the server checks if $H_2(\hat{e}(U, T_3^{sk_S}))$ is equal to *V*. If yes, the server sends the search result to the receiver.

The process of the proposed scheme is illustrated in Fig. 3.

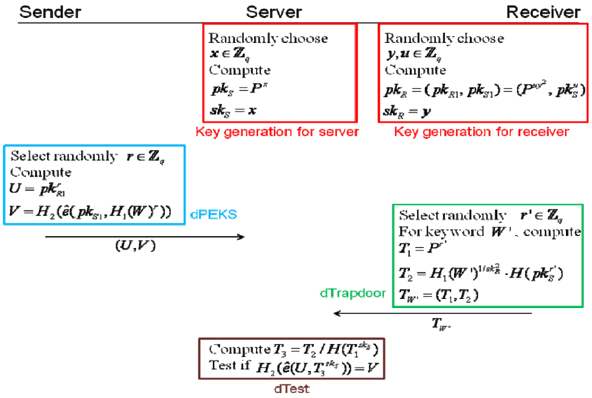

Sender
Server
Receiver
Randomly choose $x \in \mathbb{Z}_q$ Compute $pk_S = P^x$ $sk_S = x$
Key generation for server
Randomly choose $y, u \in \mathbb{Z}_q$ Compute $pk_R = (pk_{R1}, pk_{S1}) = (P^{uy^2}, pk_S^u)$ $sk_R = y$
Key generation for receiver
Select randomly $r \in \mathbb{Z}_q$ Compute $U = pk_{R1}^r$ $V = H_2(\hat{e}(pk_{S1}, H_1(W)^r))$
dPEKS
(U,V)
Select randomly $r' \in \mathbb{Z}_q$ For keyword W', compute $T_1 = P^{r'}$ $T_2 = H_1(W')^{1/sk_R^2} \cdot H(pk_S^{r'})$ $T_{W'} = (T_1, T_2)$
dTrapdoor
$T_{W'}$
Compute $T_3 = T_2 / H(T_1^{sk_S})$ Test if $H_2(\hat{e}(U, T_3^{sk_S})) = V$
dTest

Fig. 3. The process of the proposed schemecorrectness and security analysis

A. Correctness

To prove the design of dPEKS and trapdoor for testing are correct and feasible, the detailed computation of T_3 and test processing are shown as follows.

$$T_3 = T_2 / H(T_1^{sk_S}) = T_2 / H(T_1^x)$$

$$= [H_1(W')^{1/y^2} \times H((P^x)^{r'})] / H(P^{xr'})$$

$$= H_1(W')^{1/y^2}$$

$$H_2(\hat{e}(U, T_3^{sk_S})) = H_2(\hat{e}(pk_{R1}^r, H_1(W')^{x/y^2}))$$

$$= H_2(\hat{e}(P^{ruy^2}, H_1(W')^{x/y^2}))$$

$$= H_2(\hat{e}(P^{ru}, H_1(W')^x))$$

$$= H_2(\hat{e}(P^{xu}, H_1(W')^r))$$

$$= H_2(\hat{e}(pk_{S1}, H_1(W')^r))$$

If $W' = W$,

$$H_2(\hat{e}(pk_{S1}, H_1(W')^r)) = H_2(\hat{e}(pk_{S1}, H_1(W)^r)) = V$$

B. Security Analysis

The security analysis of proposed scheme is described as follows.

Proposition 1: Designated tester

The only designated server can perform the test processing.

Proof:

In the dPEKS function, $V = H_2(\hat{e}(pk_{S1}, H_1(W)^r)$, and trapdoor function, $T_2 = H_1(W')^{1/sk_R^2} \cdot H(pk_S{}^{r'})$, the server's public key is included, so in test phase the corresponding server's private key is needed as the test functions, $T_3 = T_2 / H(T_1^{sk_S})$ and $H_2(\hat{e}(U, T_3^{sk_S}))$. Hence, an outside attacker cannot perform the test processing without server's private key.

Proposition 2: Prevent off-line keyword guessing attacks

According to dPEKS or trapdoor functions, an outside attacker cannot perform keyword-guessing attacks.

Proof:

In dPEKS function, it includes a random parameter r. Hence, without knowing r, the attacker cannot perform keyword-guessing attacks.

In trapdoor function, it uses another random parameter, r', and private key of receiver, $sk_R = y$. Hence, without these two parameters, the attacker cannot perform keyword-guessing attacks.

Proposition 3: Prevent malicious servers

According to the trapdoor function, a malicious server cannot perform keyword-guessing attacks.

Proof:

In the test phase of our improvement, when a malicious server computes T_3, he will get $H_1(W')^{1/y^2}$. But the server doesn't have receiver's private key y so he is unable to compute y^2 for getting $H_1(W')$. Furthermore, if the malicious server computes $e(pk_{R1}, T_3) = e(P^{uy^2}, H_1(W')^{1/y^2}) = e(P^u, H_1(W'))$, the server finally gets $e(P^u, H_1(W'))$. However, since the random parameter u is determined by receiver, the malicious server cannot perform keyword-guessing attacks anymore without u.

Proposition 4: Controlled searching

Without receiver's help, the server cannot perform test processing.

Proof:

When generating the trapdoor for searching, $T_2 = H_1(W')^{1/sk_R^2} \times H(pk_S^{r'})$ needs the private key of receiver, that is, before performing test processing, the private key of receiver is needed for computing trapdoor. Hence, the server is unable to generating trapdoor and doing test phase without receiver's help.

Proposition 5: Hidden queries

According to the trapdoor sent by receiver, the server cannot know any information about the keyword which the receiver interested in.

Proof:

The receiver only sends trapdoor to the server. Then because $T_1 = P^{r'}$ and $T_2 = H_1(W')^{1/sk_R^2} \times H(pk_S^{r'})$ are just exponent and result of hash value, the server cannot know any information about keyword by analyzing the trapdoor, that is, the server can only test whether the keyword included in the trapdoor exists in some ciphertext of dPEKS.

Proposition 6: Query isolation

According to the operating trapdoor sent by receiver, the server cannot link to another passed trapdoor which has the same keyword.

Proof:

In each new search processing, receiver selects a new random parameter, r', to compute trapdoor so $T_1 = P^{r'}$ and $T_2 = H_1(W')^{1/sk_R^2} \times H(pk_S^{r'})$ values are different in each new searching even that the same keywords were used in the past search processing.

Furthermore, the security comparison of the proposed scheme and the related schemes is shown in TABLE I.

TABLE I: SECURITY COMPARISON

	Boneh et al. [1]	Baek et al. [6]	Rhee et al. [8]	Proposed scheme
Designated tester	Not provided	Provided	Provided	Provided
Off-line keyword guessing attacks	Insecure	Insecure	Secure	Secure
Malicious servers' attacks	Insecure	Insecure	Insecure	Secure

IV. CONCLUSION

In this paper, we have pointed out the Rhee et al.'s dPEKS scheme [8] with the weakness against off-line keyword-guessing attacks by malicious servers. Hence, we modify their scheme and propose an improved dPEKS scheme to prevent the attacks from malicious servers. And the improvement is demonstrated to prevent attacks from the malicious servers.

ACKNOWLEDGMENT

This work was partially supported by National Science Council under Grant No. NSC 99-2221-E-415-017 and NSC 99-2221-E-415-008.

REFERENCES

[1] D. Boneh, G. D. Crescenzo, R. Ostrovsky and G. Persiano, "Public key encryption with keyword search," in *Proc. EUROCRYPT'04. LNCS Conf.*, 2004, pp. 506-522.

[2] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Proc. Pairing 2007, LNCS Conf.*, 2007, vol. 4575, pp. 2-22.

[3] J. Shao, Z. F. Cao, X. H. Liang and H. Lin, "Proxy re-encryption with keyword search," *Information Sciences*, vol. 180, issue 13, pp. 2576-2587, July 1, 2010.

[4] W. C. Yau, R. C.-W. Phan, S. H. Heng and B. M. Goi, "Proxy re-encryption with keyword search: new definitions and algorithms," *Communications in Computer and Information Science*, vol. 122, pp. 149-160, 2010.

[5] B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search," *Network and Computer Applications*, vol. 34, issue 1, pp. 262-267, Jan. 2011.

[6] J. Baek, R. Safavi-Naini and W. Susilo, "Public key encryption with keyword search revisited," in *Proc. ACIS'06 Conf.*, 2006.

[7] J. W. Byun, H. S. Rhee, H. A. Park and D. H. Lee, "Off -line keyword guessing attacks on recent keyword search schemes over encrypted data," in *Proc. SDM'06. LNCS Conf.*, 2006, vol.4165, pp. 75–83.

[8] 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," *Systems and Software*, vol. 83, issue 5, pp. 763-771, May 2010.

Wang BingJian was born in Kaohsiung, Taiwan in October, 1987. He received his B.S. degree in Department of Applied Mathematics from National Chiayi University, Taiwan, in 2010. He was studying the M.S. in Department of Computer Science and Information Engineering from National Chiayi University. His research interest is visual cryptography, information hiding, digital rights management and network security.

Chen TzungHer was born in Tainan, Taiwan, Republic of China, in 1967. He received the B.S. degree in Department of Information and Computer Education from National Taiwan Normal University in 1991 and the M.S. degree in Department of Information Engineering from Feng Chia University in 2001. In 2005, he received his Ph.D. degree in Department of Computer Science from National Chung Hsing University.He has been with Department of Computer Science and Information Engineering at National Chiayi University as Professor since August 2011. His research interests include information hiding, multimedia security, digital rights management, network security. He is an honorary member of the Phi Tau Phi Scholastic Honor Society.

Jeng FuhGwo He received the B.S. degree in Department of Applied Mathematics from National Chung Hsing University, Taiwan, in 1986 and the M.S. degree in Department of Computer and Information Science from National Chiao Tung University, Taiwan, in 1991. In 2006, he received his Ph.D. degree in Department of Computer Science from National Chung Hsing University.He is currently an associate professor in the department of Applied Mathematics at National Chiayi University. His research interests include information security and computer graphics.