ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2016, T. 45, Nr. 3

A Searchable Hierarchical Conditional Proxy Re-Encryption Scheme for Cloud Storage Services

Cheng-Chi Lee

Fu Jen Catholic University, Department of Library and Information Science,
510 Jhongjheng Rd., Sinjhuang Dist., New Taipei City 24205, Taiwan, R.O.C.,
Asia University, Department of Photonics and Communication Engineering,
Wufeng Shiang, Taichung 413, Taiwan, R.O.C.,
e-mail: cclee@mail.fju.edu.tw

Chun-Ta Li

Tainan University of Technology, Department of Information Management,
529 Jhong Jheng Road, Tainan 710, Taiwan, R.O.C.,
e-mail: th0040@mail.tut.edu.tw

Chin-Ling Chen

Chaoyang University of Technology, Department of Computer Science and Information Engineering,
168, Jifeng E. Rd., Wufeng District, Taichung 41349, Taiwan, R.O.C.,
e-mail: clc@mail.cyut.edu.tw

Shih-Ting Chiu

Fu Jen Catholic University, Department of Library and Information Science,
510 Jhongjheng Rd., Sinjhuang Dist., New Taipei City 24205, Taiwan, R.O.C.,
e-mail: shihting0404@hotmail.com

crossref http://dx.doi.org/10.5755/j01.itc.45.3.13224

Abstract. As cloud technologies thrive, researches in the field of cloud storage have switched their focus from encryption-decryption techniques that help data owners protect their privacy and data confidentiality to the application of searching techniques on encrypted data while maintaining high level security and privacy of outsource data. To begin with, Song et al. offered some practical techniques for searches on encrypted data. After that, Weng et al. presented their conditional proxy re-encryption scheme where the data owner can decide which ciphertext satisfies a certain keyword condition set and then can have the retrieved data re-encrypted by the semi-trusted proxy server. The basic concepts of the above schemes are indeed quite innovative and do lead the way towards the solutions to the major practical cloud storage application problems; however, of all the researches that follow, none has had both searching on encrypted data and conditional proxy re-encryption combined. In this paper, we propose a new scheme for cloud storage services that integrates keyword search with conditional proxy re-encryption. This say, with a newly added keyword or new proxy, the cloud service provider is able to generate a hierarchical key. As far as data security is concerned, our scheme provides proven data owner authentication, re-delegation, and chosen-ciphertext security. The superior performance of the proposed scheme has been established by comparing it with related works, and our security analysis as well as BAN logic correctness check also offered solid proof that the new scheme is both secure and practical.

Keywords: re-encryption; hierarchical conditional proxy re-encryption; proxy re-encryption; cloud storage; security.

1. Introduction

Nowadays, due to the amazing mobility and convenience the thriving Internet and related wireless technologies have brought, more and more people have fallen into the habit of keeping their data in cloud storage instead using traditional portable storage devices such as USB flash drives. As people get more and more dependent on cloud storage services, cloud servers have to handle larger and larger amount of data where the confidentiality has to be considered. In other words, how to provide satisfactory mobility and convenience without sacrificing data security and confidentiality in cloud environment is the main concern. Currently, when a data owner wants to store some sensitive data in cloud storage, he/she needs to encrypt the data before uploading them to the cloud storage so as to maintain data secrecy. After uploading the data to cloud storage, he/she can then access them wherever Internet connection is available; in other words, he/she can either access the data at home or office where cabled connection is ready, or he/she can use a mobile device such as a smart phone or tab with Wi-Fi when he/she is out somewhere. Of course there can also be cases where a person (the data owner) has the data uploaded to the cloud storage and then another person (the authorized data user) accesses the data stored. However, oftentimes a data owner can have a huge amount of data uploaded to cloud storage. How can he/she access a certain part or certain parts of the data stored in cloud, then? In the past, there were two ways to get the job done [14]:

1. The user downloads all his/her data from cloud. Since the data are in encrypted form, after the downloading, the user must decrypt all the data. Now the data are in plaintext format, and the user can finally search through them and pick out the part or parts he/she desires. Just as it appears, this process makes a lot of problems for the user.
2. The user sends his/her secret key to the cloud server. With the user's secret key, the cloud server decrypts all the data uploaded by the user and finds the part or parts of the data that the user desires. In this design, the user has no choice but to totally trust the cloud server, which can be a serious security problem if the cloud server has malicious purposes.

To deal with the above problems, Song et al. [32] were the first to raise the concept of searching on encrypted data and named it the method of keyword search. In their method, the data owner can encrypt the data with some keywords, and the user can later access a certain part of the encrypted data that contains a specified keyword without having to download all the encrypted data, decrypt them all, and then do the searching. This way, the user can easily retrieve the part of the data that is needed without leaking any information. Here is a scenario to illustrate the concept of keyword search on encrypted data: Suppose Alice wants to store some data in cloud storage. She generates the ciphertext of the data. To make the data easy to access, Alice also sets the keyword "October" for the data. After generating the ciphertext of the keyword "October", Alice sends all the encrypted data to cloud storage. Later, when Bob, an authorized user, wants to retrieve the data that contains the keyword "October", he first generates the trapdoor of the keyword "October" and then sends this trapdoor to the cloud server as an access request. Upon receiving the request, the cloud server searches through the encrypted data and finds the data that contains the keyword "October" without decrypting the ciphertext. After that, the cloud server returns the corresponding ciphertext to Bob.

However, in real-world practice, there are always risks when the cloud user has to fully trust the cloud service provider. In other words, there is no way the data owner should hand his/her private key over to the server. To solve this problem, Blaze et al. [3] presented the concept of proxy re-encryption which allows the delegated semi-trusted server to re-encrypt the ciphertext by using a re-encryption key without learning any information about the plaintext. There is a scenario to illustrate the concept of proxy re-encryption: Alice uses her public key to encrypt the data and uploads the encrypted data to the server. Alice has some data for Bob, but she does not want Bob to have her private key. Without Alice's private key, Bob cannot decrypt the data. In order for Bob to be able to decrypt the ciphertext by using his own private key, Alice exploits her pubic key and Bob's public key to generate a new key for the server called a re-encryption key. With this key, the server can re-encrypt the ciphertext without getting the plaintext. Then Bob can use his private key to decrypt the ciphertext without getting Alice's private key.

Later in 2009, the notion of conditional proxy re-encryption was brought up by Weng et al. [36]. As the name suggests, by applying conditional proxy re-encryption, the data owner is enabled to decide which ciphertext satisfies a certain keyword condition set that can be re-encrypted by the proxy. Then, in 2012, Fang et al. took a step further and proposed a hierarchical conditional proxy re-encryption scheme [15]. Inspired by Fang et al., in this paper, we shall propose a searchable hierarchical conditional proxy re-encryption scheme we have designed for cloud storage. As the name reveals, the aim of our new scheme is to combine keyword search and conditional proxy re-encryption. Our scheme has the following properties:

1. Searching data without decrypting the ciphertext
   The CSP (Cloud Server Provider) does not need to decrypt the ciphertext; all the CSP does with the data in cloud storage is search on the encrypted data with a keyword in encrypted format to find the data the user needs.
2. User authentication
   The CSP can confirm the user's real identity with the trapdoor sent from the user.

3. Data owner authentication
   The CSP can utilize the ciphertext uploaded by the data owner and some public parameters to verify the legality of the data owner's identity and the ciphertext.
4. Re-delegation
   The CSP can utilize its re-encryption key to derive the sub-re-encryption key for the newly added keyword or for their children.
5. Chosen-ciphertext security
   Our scheme is based on Fang et al.'s design [15]; by the same token, our scheme provides the same level of chosen-ciphertext security on the first and the second ciphertext.

The rest of this paper is organized as follows. In Section 2, we shall review some related works dealing with keyword search and proxy re-encryption. Then, in Section 3, we will offer some preliminaries. In Section 4, we shall present the details of our new searchable hierarchical conditional proxy re-encryption scheme for cloud storage, followed by the results of a number of analyses on the proposed scheme's features, performance, security, and correctness in Section 5. Finally, the conclusions will be in Section 6.

2. Related works

In this section, some related works dealing with keyword search on encryption data as well as some proxy re-encryption and conditional proxy re-encryption schemes will be quickly reviewed.

2.1. Keyword search on encrypted data

To make searching on encrypted data possible, Song et al. [32] first proposed a secure keyword search scheme in 2000. After that, many researchers have focused on how to design secure, efficient schemes for searches on encrypted data [2, 4, 6, 9, 17, 19, 20, 21, 22, 24, 25, 27, 28, 31, 39, 40]. In 2004, Boneh et al. [4] proposed the idea of public key encryption with keyword search (PEKS), which allows the server to search through the stored data for the parts that contain certain keywords without decrypting the ciphertext. Golle et al. [17] proposed a conjunctive keyword search mechanism that allows the user to search with a conjunction of multiple keywords. Later, Park et al. [27] proposed an efficient public encryption scheme with conjunctive keyword search. On the other hand, to avoid the use of pairing operations, in 2006, Khader [20] proposed a public key encryption scheme with keyword search based on K-Resilient IBE. In 2008, Baek et al. [2] extended the PEKS into a secure channel free public key encryption scheme with keyword search (SCF-PEKS), which does not include any secure channel between the user and the server. Then, in 2009, Liu et al. [24] proposed an efficient privacy preserving keyword search (EPPKS) scheme to improve the performance of PEKS, while Rhee et al. [28] brought up the concept of trapdoor indistinguishability and proposed a new scheme to mend the weakness they found in Baek et al.'s SCF-PEKS. In 2012, Liu et al. [25] improved Liu et al.'s EPPKS and proposed a new keyword search scheme called Secure and Privacy-preserving Keyword Search (SPKS) that can do searches on encrypted data with the server in charge of the re-encryption of the ciphertext.

2.2. Proxy re-encryption

A proxy re-encryption (PRE) scheme allows the delegated semi-trusted server to re-encrypt the ciphertext by using its re-encryption key without learning any information about the plaintext. The concept of proxy re-encryption was proposed by Blaze et al. [3] in 1998. Later on, the pairing operation was commonly used in schemes of this kind [1, 8, 11, 18, 23, 35]. In 2007, Ateniese et al. proposed an identity-based proxy re-encryption scheme where the ciphertext can be transformed from one identity to another [1]. In addition, Chu and Tzeng [11] also proposed an identity-based proxy re-encryption scheme without random oracles. Finally, due to the fact that the pairing operation consumes too much communication resources, in recent years, some PRE schemes have been proposed to avoid the use of the pairing operation [10, 13, 26, 29].

2.3. Conditional proxy re-encryption

Firstly, type-based proxy re-encryption (TB-PRE) is a design where the data owner can categorize his/her ciphertext into different subsets and then delegate the decryption right of each subset to a specific delegator. In 2008, Tang [33] first proposed the construction of TB-PRE, providing fine-grained delegation and enabling the semi-trusted server to re-encrypt ciphertext of a specific type by using a re-encryption key. Since then, quite a big portion of research endeavors in the field of study have been dedicated to the development of TB-PRE schemes [12, 15, 16, 30, 34, 36, 37]. Among the schemes, Seo et al.'s TB-PRE scheme offered proven security against the standard-model chosen ciphertext attack and achieved proxy invisibility [30]. Since by definition TB-PRE means that the data owner can categorize the ciphertext into different subsets, TB-PRE is also referred to as conditional proxy re-encryption (C-PRE), where a condition is equivalent to a type [30]. Weng et al. [36] presented a kind of conditional proxy re-encryption where the data owner can assign some specific ciphertext to match a certain keyword condition set that can be re-encrypted by the semi-trusted proxy server. Later, Weng et al. [37] pointed out that Weng et al.'s scheme [36] had failed to achieve chosen ciphertext attack security (CCA-security), and so they proposed a new C-PRE scheme to fix that problem. In addition, Fang et al. [16] also proposed an anonymous conditional proxy re-encryption scheme without random oracle. Chu et al. [12] presented a conditional proxy broadcast re-encryption scheme where the proxy can delegate decryption rights to a set of users at a time. In 2010, Vivek et al. [34] improved

the performance of Weng et al.'s [37] C-PRE scheme and proposed a more efficient construction for C-PRE. In 2012, Fang et al. proposed a hierarchical conditional proxy re-encryption (HC-PRE) scheme that enhanced the concept of C-PRE by allowing more general re-encryption key delegation patterns [15].

To this day, no scheme proposed has had both ideas of searching on encrypted data and conditional proxy re-encryption combined. Inspired by Fang et al. [15], in this paper, we propose a new scheme that puts together keyword search and conditional proxy re-encryption.

3. Preliminaries

In this section, we shall review bilinear pairing [5], give some complexity assumptions in our scheme, and then introduce the idea of hierarchical conditional proxy re-encryption [37].

3.1. Bilinear pairing

Let $\mathbb{G}_1$and $\mathbb{G}_2$ be two cyclic group with prime order p, and g is the generator of group $\mathbb{G}_1$. Suppose we have $a, b \in \mathbb{Z}_q$ and a bilinear map e: $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Then there are some notable properties as follows [5]:

1. Bilinearity for all $a, b \in \mathbb{Z}_q$ and $P, Q \in \mathbb{G}_1$,$e(aP, bQ) = e(P, Q)^{ab}$.
2. Computability. There in always an efficient polynomial time algorithm to compute $e(P, Q) \in \mathbb{G}_2$, for any $P, Q \in \mathbb{G}_1$.
3. Non-degeneration. There is always such a pair of P and $Q \in \mathbb{G}_1$ that satisfies $e(P, Q) \neq 1$.

3.2. Hierarchical conditional proxy re-encryption

Here is the hierarchical conditional proxy re-encryption design proposed by Weng et al. in 2009 [37]. In their scheme, there are eight algorithms: setup, key generation, re-encryption key generation, level 2 encryption, level 1 encryption, re-encryption, level 2 decryption, and level 1 decryption. Figure 1 gives a rough idea of how the system works, and the algorithms are as follows:

- Setup: The setup algorithm is executed by a trusted party with the input being the security parameter 1^K and the output the global parameters GP.
- KeyGen: The key generation algorithm produces the public key pk_i and secret key sk_i for the user i.
- RKeyGen: The re-encryption key generation algorithm takes the secret key sk_i, the conditional keyword w, and the other public key pk_j as input and then outputs the re-encryption key $rk_{i \rightarrow j}^{w}$.
- Enc2: Level 2 encryption algorithm intakes the public key pk, the plaintext $m \in \mathcal{M}$ and the conditional keyword w and then outputs level 2 ciphertext CT. Here $\mathcal{M}$ is the message space.
- Enc1: Level 1 encryption algorithm takes the public key pk and the plaintext $m \in \mathcal{M}$ as input and then outputs level 1 ciphertext CT. Notice that this ciphertext cannot be encrypted by any other user.
- ReEnc: The re-encryption algorithm intakes the second ciphertext CT and the re-encryption key $rk_{i \rightarrow j}^{w}$.
- Dec2: Level 2 decryption algorithm takes the second ciphertext CT and the secret key sk as input and then outputs the message m.

Dec1: Level 1 decryption algorithm intakes the first ciphertext CT and the secret key sk and then outputs the message m.

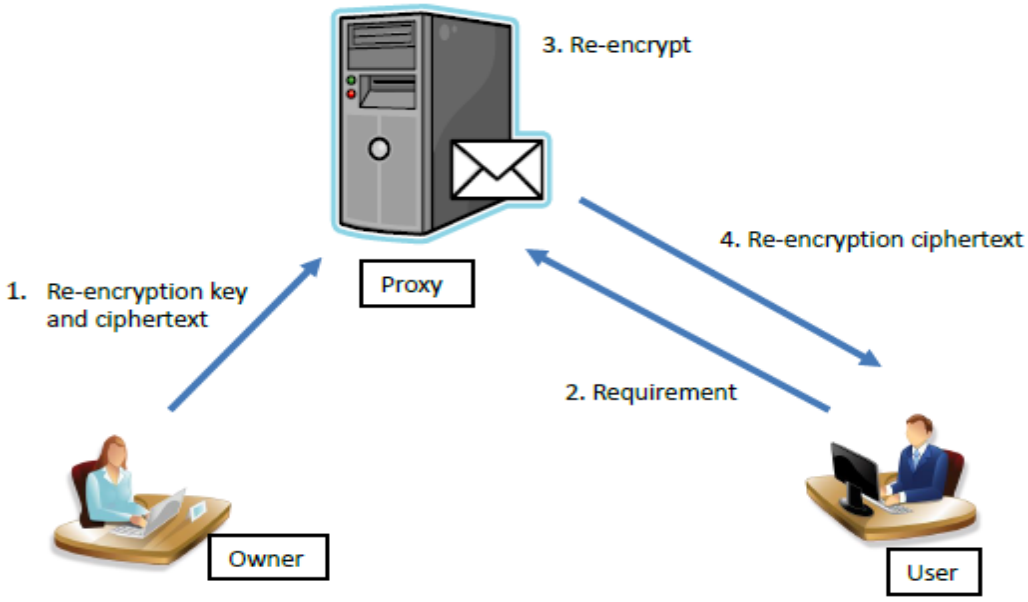

Figure 1. Hierarchical conditional proxy re-encryption

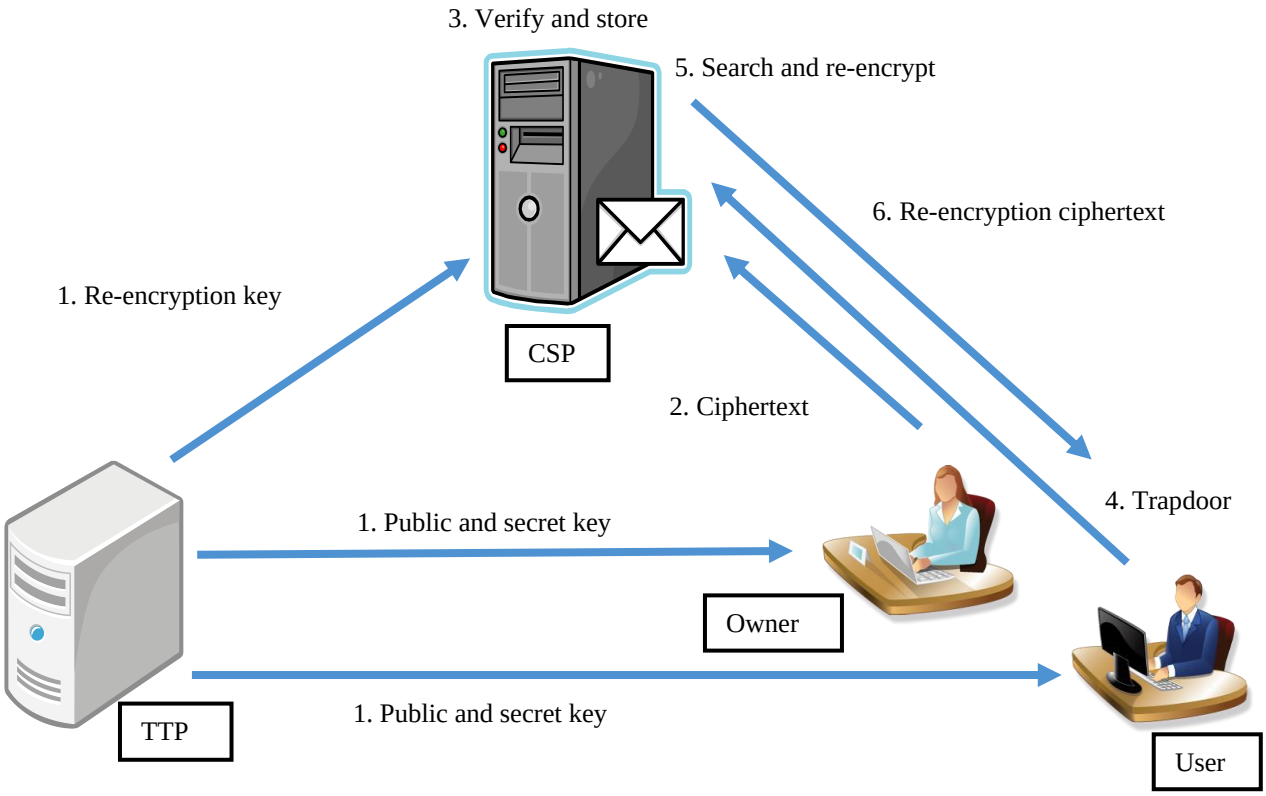

Figure 2. Searchable hierarchical conditional proxy re-encryption

4. The proposed scheme

In this section, we shall present our searchable hierarchical conditional proxy re-encryption scheme. We will first illustrate the framework of our scheme and then give detailed descriptions to all the phases of our scheme.

4.1. Framework of our scheme

In this subsection, we shall first introduce the participants in our scheme and then the phases. There are four kinds of participants in our scheme: the trusted third party (TTP), the cloud service provider (CSP), the data owner, and the users. The role each participant plays is shown as follows.

1. Trusted third party (TTP): The trusted third party is responsible for generating the public key and the secret key for the user and the data owner and also generating the re-encryption key for the cloud server provider.
2. Cloud service provider (CSP): The function of CSP is to accept and store the ciphertext sent by the data owner. Upon receiving the retrieval request from the user, CSP searches through the stored data and finds what the user wants. Besides that, CSP is able to re-encrypt the ciphertext and uses a re-encryption key to generate a hierarchical key for a newly added keyword.
3. Data owner: The data owner generates ciphertext on two different levels. One does not contain the keyword vector, while the other contains the keyword vector set by the data owner.
4. Users: When a user wants to retrieve some data that contains a certain keyword, the user needs to generate the trapdoor of the keyword and then send it to CSP as a request. Then, when the user receives the re-encrypted ciphertext that CSP returns, he/she can use his/her secret key to decrypt it.

There are 11 phases in our scheme: setup, key generation, re-encryption key generation, level 1 encryption, level 2 encryption, verification, trapdoor generation, keyword searching, re-encryption, level 1 decryption, and level 2 decryption. The flowchart of our scheme is shown in Figure 2, and the function of each phase is as follows:

- Setup: In this phase, the security parameter λ is the input, the bilinear map is set, and then the system public parameters are outputted.
- KeyGen: In this phase, the system public parameters are inputted, and the public key and the secret key for the data owner and the user are outputted.
- Re-keyGen: In this phase, the inputs are the user's secret key, the data owner's secret key and a conditional keyword vector, and then the output is the re-encryption key for CSP. When a new keyword is added to the conditional keyword vector,

CSP can use the current re-encryption key to generate a new re-encryption key. This is called hierarchical key derivation.

- Enc1: In order to have the message encrypted, the data owner inputs the message along with his/her public key and then gets the first level ciphertext for CSP.
- Enc2: To encrypt the message with a conditional keyword vector, the data owner inputs the message along with his/her public key and a conditional keyword vector. The output is the second level ciphertext for CSP.
- Verify: Upon receiving the ciphertext, CSP determines whether the ciphertext is truly sent by the data owner and has not been tampered by a malicious attacker.
- Trapdoor: In order to retrieve the data which contains a certain keyword, the user generates the trapdoor of the keyword vector and then sends it to CSP.
- Search: To search for the data the user requests, CSP inputs the ciphertext, the user's public key and the trapdoor.
- ReEnc: When CSP finds the data that the user requests, CSP uses the re-encryption key to encrypt the ciphertext.
- Dec1: The user inputs his/her secret key and the first level ciphertext to decrypt the ciphertext.
- Dec2: The user inputs his/her secret key and the second level ciphertext to decrypt the ciphertext.

4.2. Our scheme

In this subsection, we look into the details of the phases in our scheme. Table 1 lists the notations used in our scheme.

Table 1. Notations used in our scheme

Notations	Descriptions
p	A prime order
g	A generator of G_1
G_1, G_2	Multiplicative cyclic groups of prime order p
e	Bilinear map $e: G_1 \times G_1 \to G_2$
H_1, H_2, H_3, H_4	One-way hash functions
w_i	Keyword
L	The maximum length of keyword vector
m	The message, $m \in \mathcal{M}$
$\oplus$	XOR operation

- Setup: With a security parameter λ inputted, set (p, g, G_1, G_2, e) as bilinear map parameters. Then, $\mathcal{M} = \{0,1\}^{k_1}$ is set as the message space, and there are four one-way hash functions $H_1: \{0,1\}^* \to Z_p^*, H_2: G_2 \to \{0,1\}^{k_1}, H_3: \{0,1\}^* \to G_1^*$, and $H_4: \{0,1\}^* \to Z_p^*$. Let the conditional keyword vector be $W = (w_1, w_2, \dots, w_k) \in \{0,1\}^*$, where k is the length of W. Generate the random numbers $g_1, g_2, h_1, h_2, \dots, h_L \in G_1$. The system public parameters are $(p, g, G_1, G_2, e, g_1, g_2, h_1, \dots, h_L, k_1, L, H_1, H_2, H_3, H_4)$.
- KeyGen: Generate a random number $x_i \in Z_p^*$ for user i and then compute $X_i = g^{x_i}$. Set the public key as $pk_i = X_i$ and secret key as $sk_i = x_i$ for user i.
- Re-keyGen: Given the data owner's secret key sk_i, the conditional keyword vector $W = (w_1, w_2, \dots, w_k)$, and the user's secret key sk_j, select a random number $r \in Z_p^*$ and compute

$$a_0 = g_2^{x_i - x_j} \left(\prod_{l=1}^{k} h_l^{H_4(pk_i, w_l)} g_1 \right)^r_{l \in \{k+1, \dots, L\}}, \quad (1)$$

$$a_1 = g^r, \quad (2)$$

$$b = (b_l = h_l^r)_{l \in \{k+1, \dots, L\}}. \quad (3)$$

The re-encryption key for CSP is $rk_{i,W,j} = (a_0, a_1, b)$. When CSP needs to generate a new re-encryption key for a new keyword vector $W = (w_1, w_2, \dots, w_k, w_{k+1})$, CSP picks a random number $t \in Z_p^*$ and then computes

$$a_0' = a_0 b_{k+1}^{H_4(pk_i, w_{k+1})} \left(\prod_{l=1}^{k+1} h_l^{H_4(pk_i, w_l)} g_1 \right)^t_{l \in \{k+2, \dots, L\}}, \quad (4)$$

$$a_1' = a_1 g^t, \quad (5)$$

$$b' = (b_l = h_l^t)_{l \in \{k+2, \dots, L\}}. \quad (6)$$

The hierarchical re-encryption key is $rk_{i,W_{k+1},j} = (a_0', a_1', b')$, which is properly distributed to W_{k+1} for $r' = r + t$.

- Enc1: Data owner chooses a random number $R \in G_2^*$ and then computes

$$s = H_1(m, R), \quad (7)$$

$$B = g^s, \quad (8)$$

$$D = e(X_i, g_2)^s R, \quad (9)$$

$$E = m \oplus H_2(R). \quad (10)$$

The first level ciphertext is $CT_i = (B, D, E)$.

- Enc2: To encrypt the message with the conditional keyword vector $W = (w_1, w_2, \dots, w_k)$, data owner chooses $R \in G_2^*$ and then computes

$$s = H_1(m, R), \quad (11)$$

$$B = g^s, \quad (12)$$

$$C = \left(\prod_{l=1}^{k} h_l^{H_4(pk_i, w_l)} g_1 \right)^s, \quad (13)$$

$$D = e(X_i, g_2)^s R, \quad (14)$$

$$E = m \oplus H_2(R), \quad (15)$$

$$F = H_3(B, C, D, E)^s. \quad (16)$$

The second level ciphertext is $CT_i = (B, C, D, E, F)$.

- Verify: After receiving the ciphertext, CSP checks out

$$e\left(\prod_{l=1}^{k} h_l^{H_4(pk_i, w_l)} g_1, B \right) =? e(C, g), \quad (17)$$

$$e(H_3(B, C, D, E), B) =? e(F, g). \quad (18)$$

If both check out, CSP accepts and stores the ciphertext.

- Trapdoor: When the user wants to retrieve a part of the stored data that contains the conditional keyword vector $W = (w_1, w_2, \dots, w_k)$, he/she computes the trapdoor of the conditional keyword vector as

$$T_{w_j} = \left(\prod_{l=1}^{k} h_l^{H_4(X_i, w_l)} g_1\right)^{x_j} \quad (19)$$

and then sends it to CSP.

- Test: When receiving the trapdoor from the user, CSP tests to see whether $e(B, T_{w_j})$ is equal to $e(pk_j, C)$ or not. If the result is positive, CSP re-encrypts the ciphertext and then sends it to the user.
- ReEnc: After finding the data that the user requests, CSP re-encrypts the ciphertext by computing

$$D' = \frac{e(a_1, C)}{e(a_0, B)} \cdot D. \quad (20)$$

The re-encrypted ciphertext, namely $CT_j = (B, D', E)$, is then sent to the user.

- Dec1: To decrypt the re-encrypted first level ciphertext $CT_j = (B, D', E)$, the user uses his/her secret key $sk_j$and computes

$$R = \frac{D'}{e(B, g_2)^{x_j}}, \quad (21)$$

$$m = E \oplus H_2(R), \quad (22)$$

$$s = H_1(m, R). \quad (23)$$

After computing R, m and s, the user checks $B = ?\ g^s$. If it checks out, then the message m is returned.

- Dec2: To decrypt the re-encrypted second level ciphertext $CT_j = (B, C, D', E, F)$ containing the conditional keyword vector, the user uses his/her secret key sk_j and computes

$$R = \frac{D'}{e(B, g_2)^{x_j}}, \quad (24)$$

$$m = E \oplus H_2(R), \quad (25)$$

$$s = H_1(m, R). \quad (26)$$

After computing $R, m,$ and s, the user checks

$$B = ?\ g^s, \quad (27)$$

$$C = ?\ \left(\prod_{l=1}^{k} h_l^{H_4(pk_i, w_l)} g_1\right)^s, \quad (28)$$

$$F = ?\ H_3(B, C, D', E)^s. \quad (29)$$

If all check out, then the message m is returned.

5. Analysis of the proposed scheme

In this section, we shall first show how our new scheme compares with Zhao et al.'s [40], Liu et al.'s [25], Fang et al.'s [15], and Seo et al.'s scheme [30] in terms of function as well as performance. Then, we will analyze the security of our scheme and confirm the correctness with a BAN logic [7, 38] check.

5.1. Comparisons

In this subsection, we compare the functions and performance of our scheme with those of Zhao et al.'s, Liu et al.'s, Fang et al.'s, and Seo et al.'s scheme. Of all the schemes compared, Zhao et al.'s, and Liu et al.'s focus on secure keyword search , while Fang et al.'s, and Seo et al.'s focus on conditional proxy re-encryption.

5.1.1. Function comparison

Before looking into the comparison results, let's define some abbreviations we use. Expressions such as AuthID Pro, User Auth, Owner Auth, Searching, and P-Re are used to indicate authorized identity protection, user authentication, data owner authentication, search on encrypted data, and proxy re-encryption, respectively. The comparison results are given in Table 2. As the table reveals, Zhao et al.'s, and Liu et al.'s both fall short of offering data owner authentication, which means vulnerability to the modification attack where the attacker sends fake ciphertext to CSP and the user never receives the data he/she requests. On the other hand, although Fang et al.'s and Seo et al.'s are under the protection of data owner authentication, they are both incapable of supporting searches on encrypted data. In contrast, our scheme offers both data owner authentication but also searching on encrypted data.

Table 2. Function comparison of our scheme and other schemes

	AuthID Pro	User Auth	Owner Auth	Searching	P-Re
Zhao et al.'s	v	v	x	v	x
Liu et al.'s	v	v	x	v	v
Fang et al.'s	v	v	v	x	v
Seo et al.'s	v	v	v	x	v
Our scheme	v	v	v	v	v

5.1.2. Performance comparison

For the performance comparison, we use Encrypt, Trapdoor, Verification, Test, and Re-encryption as abbreviations for conditional encryption, trapdoor generation, verification of data owner, keyword test, and proxy re-encryption, respectively. Note that conditional encryption includes conditional encryption, type-based encryption, and keyword encryption. In addition, we define P as a map-to-point hash function operation, E as a pairing operation, and M as a multiplication operation in G_1. The performance comparison results are given in Table 3. According to the running time calculations in millisecond given in [41], the running time of one map-to-point hash function operation is 3.04 *ms*, one pairing operation is 20.04 *ms*, and one multiplication operation is 2.21 *ms*. Check the table, we can see that the performance of our scheme is not the best. However, our scheme can support the all functions in Table 2.

Table 3. Performance comparison of our scheme and other schemes

	Encrypt	Trapdoor	Verification	Test	Re-encryption
Zhao et al.'s	$1P + 2E + 3M$	$4P + 1E + 3M$	–	$1P + 4E + 2M$	–
Liu et al.'s	$1P + 1E$	$1P$	–	$1E$	$1P + 2E + 2M$
Fang et al.'s	$3P + 1E + 3M$	–	$2E + 1M$	–	$2E$
Seo et al.'s	$1E + 4M$	–	$1E + 2M$	–	$1M$
Our scheme	$3P + 1E + 3M$	$0P + 0E + 0M$	$2E + 1M$	$1E$	$2E$

1. CSP can verify the data owner's identity.
   To determine the legitimacy of the data owner, CSP utilizes the ciphertext B, C, D, E, F, data owner's public key, and the keyword vector to verify the data owner's identity. Because the data owner uses the public key to generate the ciphertext, CSP can confirm the data owner's identity by checking out the ciphertext.
2. CSP can verify that the sender of the ciphertext is an authorized data owner.
   To avoid mistakenly accepting tampered ciphertext from a malicious attacker, CSP must check the integrity of the ciphertext. When CSP verifies the data owner's identity, the ciphertext is examined at the same time. If any part of the ciphertext is tampered, it cannot pass the verification.
3. CSP can verify the user's identity.
   Upon receiving the trapdoor of the keyword vector from a user as a searching request, CSP must check the user's identity to make sure he/she is properly authorized. CSP utilizes the ciphertext B, C, D, E, F, the data owner's public key, the user's pubic key and the keyword vector to verify the user's identity. Only a legitimate user owns the secret key that can be used to generate the trapdoor. In fact, CSP can verify the user's identity and search for the data the user requests as the same time.
4. The user can verify whether the ciphertext is tampered.
   Upon receiving the re-encrypted ciphertext, the user verifies the integrity of the re-encrypted ciphertext to determine whether it has been tampered by a malicious attacker. The user exploits his/her secret key to decrypt the re-encrypted ciphertext. After decrypting the re-encrypted ciphertext, the user exploits the re-encrypted ciphertext and the plaintext to check the integrity of the ciphertext. Only CSP has the re-encryption key and thus can have the ciphertext re-encrypted, and only the legitimate user can exploit his/her secret key to recover the integral plaintext.
5. Our scheme can achieve chosen-ciphertext security.
   Based on Fang et al.'s design [15], our scheme inherits the chosen-ciphertext security on the first and the second ciphertext.

5.3. Correctness analysis

In this subsection, we use the BAN logic [7, 38] to check the correctness of the data owner verification, user verification, and ciphertext verification of our scheme. The BAN logic is a well-accepted method to analyze the correctness of cryptographic protocols. Before applying the BAN logic, let's define some notations, goals and assumptions as follows.

5.3.1. Notations

Here we deal with the syntax and notations of the BAN logic. Assume that A and B are some specific participators, and X is the formula (statement). The basic rules of language are as follows [7, 38]:

1. $A|\equiv X$ means A believes that formula X is ture.
2. $A|\equiv B$ means A believes B's action.
3. $A|\Rightarrow X$ means A has complete control over formula X.
4. $A \lhd X$ means A holds or sees formula X.
5. $\#(X)$ means formula X is fresh and has not been used before.
6. $\overset{K_A}{\longmapsto} A$ means K_A is the public key for A and K_A^{-1} is the private key for A.
7. $\frac{Rule\ 1}{Rule\ 2}$ means $Rule\ 2$ is derived from $Rule\ 1$.

5.3.2. Goals

The roles and the goals in our scheme are as follows. First, there are four roles in our scheme: the trusted third party (TTP), the data owner ($Owner$), the cloud service provider (CSP), and the user ($User$). Then, there are three goals to be achieved. In the BAN logic language, the three goals are:

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

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

$G3.User|\equiv CSP \lhd rk$

$G1$ means in verification phase CSP needs to make sure that the sender of the ciphertext is $Owner$ and that the ciphertext has not been tampered by an attacker. So CSP must believe that $Owner$ holds his/her private key so that he/she can create the ciphertext. $G2$ means in the test phase CSP needs to verify $User$'s identity to determine that the trapdoor is permissible by believing that $User$ holds his/her private key so that he/she can

create the trapdoor. $G3$ means $User$ needs to determine that the re-encrypted ciphertext has not been tampered by an attacker; in other words, $User$ needs to believe that CSP holds the re-encryption key rk to generate the re-encrypted ciphertext.

5.3.3. Assumptions

With the goals set, now let's state our assumptions as follows:

$A1.CSP|\equiv \overset{K_{owner}}{\mapsto} Owner$

$A2.User|\equiv \overset{K_{Owner}}{\mapsto} Owner$

$A3.CSP|\equiv \overset{K_{User}}{\mapsto} User$

$A4.Owner|\Rightarrow K_{owner}^{-1}$

$A5.User|\Rightarrow K_{User}^{-1}$

$A6.CSP|\Rightarrow rk$

$A7.CSP|\Rightarrow W$

5.3.4. Verification of the data owner

The data owner verification process in the verification phase is checked with the BAN logic as follows:

Message 1: $Owner \longrightarrow CSP: CT_i = (B, C, D, E, F)$

$V1.\ CSP \lhd B, C, D, E, F$

$V2.\frac{CSP \lhd w_l, CSP \lhd B}{CSP \lhd C}$

$V3.\frac{CSP \lhd C, CSP \lhd D, CSP \lhd E}{CSP \lhd F}$

$V4.\frac{CSP|\equiv F}{CSP|\equiv (B,D,E)}$

$V5.\frac{CSP|\equiv D, CSP\left|\equiv \overset{K_{owner}}{\mapsto} Owner\right.}{CSP|\equiv Owner \lhd K_{owner}^{-1}}$

When CSP receives the ciphertext from $Owner$, CSP can exploit the information to determine the correctness. From formula $V5$, we can infer that our scheme does achieve the goal we set. By formula $V5$, CSP believes that $Owner$ holds the private key to create the ciphertext.

5.3.5. Verification of the user

The correctness of user verification in the test phase is verified with the BAN logic as follows:

Message 1: $User \longrightarrow CSP: T_{w_j}$

$V1.\ CSP \lhd T_{w_j}$

$V2.\frac{CSP|\equiv W, CSP|\equiv (B,C), CSP\left|\equiv \overset{K_{User}}{\mapsto} User\right.}{CSP|\equiv T_{w_j}}$

$V3.\frac{CSP|\equiv T_{w_j}}{CSP|\equiv User \lhd K_{User}^{-1}}$

When CSP receives the trapdoor, CSP can exploit the ciphertext sent from $Owner$ and $User$'s public key to determine the correctness. Formula $V3$, we can infer that our scheme achieves the goal we set for $Test$ phase. By formula $V3$, CSP believes that $User$ holds the private key to create the trapdoor.

5.3.6. Verification of the ciphertext

In this subsection, we examine the correctness of the re-encrypted ciphertext verification process in the decryption phase (including $Dec1$ and $Dec2$) with the BAN logic. The details are as follows:

For $Dec1$:

Message 1: $CSP \longrightarrow User: CT_j = (B, D', E)$

$V1.\ User \lhd B, D', E$

$V2.\frac{User \lhd (B,D'), User \lhd K_{User}^{-1}}{User \lhd R}$

$V3.\frac{User \lhd E, User \lhd R}{User \lhd m}$

$V4.\frac{User \lhd (m,R)}{User \lhd s}$

$V5.\frac{User|\equiv (s,B)}{User|\equiv (R,m)}$

$V6.\frac{User|\equiv R}{User|\equiv D'}$

$V7.\frac{User|\equiv D'}{User|\equiv CSP \lhd rk}$

When $User$ receives the ciphertext, he/she exploits all information contained in it to determine that the re-encrypted ciphertext is truly sent by CSP and has not been tampered by an attacker. By formula $V7$, $User$ believes CSP holds the re-encryption key that can be used to re-encrypt the ciphertext, and therefore we can infer that our scheme achieves the goal we set for phase $Dec1$.

For $Dec2$:

Message 1: $CSP \longrightarrow User: CT_j = (B, C, D', E, F)$

$V1.\ User \lhd B, C, D', E, F$

$V2.\frac{User \lhd (B,D'), User \lhd K_{User}^{-1}}{User \lhd R}$

$V3.\frac{User \lhd E, User \lhd R}{User \lhd m}$

$V4.\frac{User \lhd (m,R)}{User \lhd s}$

$V5.\frac{User|\equiv (s,B)}{User|\equiv (C,R,m)}$

$V6.\frac{User|\equiv (R,m)}{User|\equiv (D',E)}$

$V7.\frac{User|\equiv D'}{User|\equiv F}$

$V8.\frac{User|\equiv F}{User|\equiv CSP \lhd rk}$

When $User$ receives the ciphertext that contains the keyword, he/she exploits all information contained in it to determine whether the re-encrypted ciphertext sent from CSP has been tampered by an attacker. By formula $V8$, $User$ believes that CSP holds the re-encryption key for the re-encryption of the ciphertext. Therefore, we can infer that our scheme achieves the goal we set for phase $Dec2$.

In this paper we presented two multistart tabu search implementations for the MAX - CUT problem. The algorithm can quickly find solutions that are competitive with those found by most successful algorithms described in the literature. For 6 benchmark graphs the solutions of weight larger than the best known value were produced.

6. Conclusions

In this paper, we have presented a searchable hierarchical conditional proxy re-encryption scheme for cloud storage services. Not only does our new scheme support hierarchical proxy re-encryption but it also allows CSP to do keyword searching on the encrypted data. If a new keyword is added, our scheme can exploit the current re-encryption key to generate a new re-encryption key for the newly added keyword. The correctness of our new scheme has been proven by a BAN logic examination. Compared with similar schemes, our scheme shows superiority in terms of function, performance, and security. So far, quite a number of new schemes including ours can support the generation of new re-encryption keys for when new keywords are added. In the future, we hope to develop a new re-encryption key that can handle keyword reduction.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable suggestions and comments. This research was partially supported by the Ministry of Science and Technology, Taiwan, R.O.C., under contract no.: MOST 105-2221-E-030-012.

References

[1] **G. Ateniese, K. Fu, M. Green, S. Hohenberger.** Improved proxy re-encryption schemes with applications to secure distributed storage. In: *Proc. of the 12th Annual Network and Distributed System Security Symposium*, 2005, pp. 29-44.

[2] **J. Baek, R. Safiavi-Naini, W. Susilo.** Public key encryption with keyword search revisited. In: *Proceedings of the 8th International Conference on Computational Science and Its Applications (ICCSA'08)*, 2008, 5072, pp. 1249-1259.

[3] **M. Blaze, G. Bleumer, M. Strauss.** Divertible protocols and atomic proxy cryptography. In: *Proc. of EUROCRYPT 1998*, LNCS, Springer, Heidelberg, 1998, Vol. 1403, pp. 127-144.

[4] **D. Boneh, G. D. Crescenzo, R. Ostrovsky, G. Persiano.** Public key encryption with keyword search. In: *Proceedings of 2004 International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'04)*, 2004, Vol. 3027, pp. 506-522.

[5] **D. Boneh, M. Franklin.** Identity-based encryption from the weil pairing. *Advances in Cryptology-CRYPTO 2001, Lecture Notes in Computer Science*, 2001,Vol. 2139, pp. 213-229.

[6] **D. Boneh, B. Waters.** Conjunctive, subset, and range queries on encrypted data. In: *Proceedings of TCC 2007, Lecture Notes in Computer Science*, 2007, Vol. 4392, pp. 535-554.

[7] **M. Burrows, M. Abadi, R. Needham.** A logic of authentication. *ACM Transactions Computer Systems*, 1990, Vol. 8, No. 1, 18-36.

[8] **R. Canetti, S. Hohenberger.** Chosen-ciphertext secure proxy re-encryption. In: *Proc. of the 14th ACM Conference on Computer and Communications Security*, ACM New York, NY, USA, 2007, pp. 185-194.

[9] **Y. C Chang, M. Mitzenmacher.** Privacy preserving keyword searches on remote encrypted data. In: *Proceedings of ACSN 2005, Lecture Notes in Computer Science*, 2005, Vol. 3531, pp. 442-455.

[10] **S. Chow, J. Weng, Y. Yang, R. Deng.** Efficient unidirectional proxy re-encryption. In: *Proc. of AFRICACRYPT 2010*, LNCS, Springer, Heidelberg, 2010, Vol. 6055, pp. 316-332.

[11] **C. Chu, W. Tzeng.** Identity-based proxy re-encryption without random oracles. In: *Proc. of ISC 2007*, LNCS, Springer, Heidelberg, 2007, Vol. 4779, pp. 189-202.

[12] **C. Chu, J. Weng, S. Chow, J. Zhou, R. Deng.** Conditional proxy broadcast reencryption. In: *Proc. of ACISP 2009*, LNCS, Springer, Heidelberg, 2009, Vol. 5594, pp. 327-342.

[13] **R. Deng, J. Weng, S. Liu, K. Chen.** Chosen-cipertext secure proxy re-encryption without pairings. In: *Proc. of CANS 2008*, LNCS, Springer, Heidelberg, Vol. 5339, 2008, pp. 1-17.

[14] **C. I. Fan, S. Y. Huang.** Controllable privacy preserving search based on symmetric predicate encryption in cloud storage. *Future Generation Computer Systems*, 2013, Vol. 29, No. 7, 1716-1724.

[15] **L. Fang, W. Susilo, C. Ge, J. Wang.** Hierarchical conditional proxy re-encryption. *Computer Standards & Interfaces*, 2012, Vol. 34, No. 4, 380-389.

[16] **L. Fang, W. Susilo, J. Wang.** Anonymous conditional proxy re-encryption without random oracle. *ProvSec, Lecture Notes in Computer Science*, Springer, 2009, Vol. 5848, pp. 47-60.

[17] **P. Golle, J. Staddon, B. Waters.** Secure conjunctive keyword search over encrypted data. In: *Proc. of the 2nd International Conference on Applied Cryptography and Network Security (ACNS'04)*, Yellow Mountain, China, Springer-Verlag, 2004, Vol. 3089, pp. 31-45.

[18] **M. Green, G. Ateniese.** Identity-based proxy re-encryption. In: *Proc. of ACNS 2007*, LNCS, Springer, Heidelberg, 2007, Vol. 4521, pp. 288-306.

[19] **S. T. Hsu, C. C. Yang, M. S. Hwang.** A study of public key encryption with keyword search. *International Journal of Network Security*, 2013, Vol. 15, No. 2, 71-79.

[20] **D. Khader.** Public key encryption with keyword search based on k-resilient IBE. *Computational Science and Its Application-ICCSA 2006*, Lecture Notes in Computer Science, Springer, Heidelberg, 2006, Vol. 3982, pp. 298-308.

[21] **C. C. Lee, S. T. Hsu, M. S. Hwang.** A study of conjunctive keyword searchable schemes. *International Journal of Network Security*, 2013, Vol. 15, 311-320.

[22] **J. Li, Q. Wang, C. Wang, K. R. N. Cao, W. Lou.** Fuzzy keyword search over encrypted data in cloud computing. In: *Proceedings of the 29th conference on Information communications (INFOCOM'10)*, San Diego, California, USA, IEEE, 2010, pp. 1-5.

[23] **B. Libert, D. Vergnaud.** Unidirectional chosen-ciphertext secure proxy reencryption. In: *Proc. of PKC 2008*, LNCS, Springer, Heidelberg, 2008, Vol. 4939, pp. 360-379.

[24] **Q. Liu, G. Wang, J.** Wu. An efficient privacy preserving keyword search scheme in cloud computing. In: *Proceedings of the 2009 International Conference on Computational Science and Engineering*, 2009, pp.715-720.

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

[26] **T. Matsuda, R. Nishimaki, K. Tanaka.** CCA proxy re-encryption without bilinear maps in the standard model. In: *Proc. of PKC 2010*, LNCS, Springer, Heidelberg, 2010, Vol. 6056, pp. 261-278.

[27] **D. J. Park, K. Kim, P. J. Lee.** Public key encryption with conjunctive field keyword search. *Lecture Notes in Computer Science*, 2004, Vol. 3325, pp. 73-86.

[28] **H. S. Rhee, J. H. Park, W. Susilo, 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*, 2009, pp. 376-379.

[29] **H. S. Rhee, W. Susilo, H. J. Kim.** Secure searchable public key encryption scheme against keyword guessing attacks. *IEICE Electronics Express*, 2009, Vol. 6, No. 5, 237-243.

[30] **J. W. Seo, D. H. Yumb, P. J. Lee.** Proxy-invisible CCA-secure type-based proxy re-encryption without random oracles. *Theoretical Computer Science*, 2013, Vol. 491, 83-93.

[31] **E. Shi, J. Bethencourt, T-H. H. Chan, D. Song, A. Perrig.** Multi-dimensional range query over encrypted data. In: *Proceedings of IEEE Symposium on Security and Privacy*, 2007, pp. 350-364.

[32] **D. Song, D. Wagner, A. Perrig.** Practical techniques for searches on encrypted data. In: *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, 2000, pp.44-55.

[33] **Q. Tang.** Type-based proxy re-encryption and its construction. In: *Proc. of INDOCRYPT 2008*, LNCS, Springer, Heidelberg, 2008, Vol. 5365, pp. 130-144.

[34] **S. S. Vivek, S. Sharmila Deva Selvi, V. Radhakishan, C. Pandu Rangan.** Conditional proxy re-encryption-a more efficient construction. In: *Proc. of CNSA 2011*, 2011, CCIS 196, pp. 502-512.

[35] **J. Weng, M. Chen, Y. Yang, R. Deng, K. Chen, F. Bao.** CCA-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles. *SCIENCE CHINA Information Sciences*, 2010, Vol. 53, 593-606.

[36] **J. Weng, R. H. Deng, C. Chu, X. Ding, J. Lai.** Conditional proxy re-encryption secure against chosen-ciphertext attack. In: *Proc. of the 4th International Symposium on ACM Symposium on Information, Computer and Communications Security (ASIACCS 2009)*, 2009, pp. 322-332.

[37] **J. Weng, Y. Yang, Q. Tang, R. H. Deng, F. Bao.** Efficient conditional proxy reencryption with chosen-ciphertext security. In: *Proc. of the 12th International Conference on Information Security (ISC 2009)*, 2009, pp. 151-166.

[38] **J. Wessels.** Application of BAN-logic. *CMG FINANCE B.V.*, 19 April 2001.

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

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

[41] **D. He, J. Chen, J. Hu.** An ID-based proxy signature schemes without bilinear pairings. *Annals of Telecommunications*, 2011, Vol. 66, 657-662.

Received September 2015.