Security of a UUP Web Search Protocol with Privacy Preserving

Fuh-Gwo Jeng[1], Bing-Jian Wang[2], Tzung-Her Chen*[2]
[1]*Department of Applied Mathematics, National Chiayi University*
[2]*Department of Computer Science and Information Engineering, National Chiayi University*

Abstract

Generally, a search engine will keep a record of a user about the websites he ever went and the past searches he had submitted to improve its performance. Similar to a spy tracking and tracing the footpath, a search engine will inevitably violate user's privacy as the record will reveal the user's personal information or the institution he works for. To protect user's privacy, Castellà-Roca et al. proposed a protocol called Useless User Profile (UUP), in which it provided a distorted user profile for a web search engine such that the web search engine cannot generate a real profile of a certain individual. One of the significant advantages lies on that their protocol requires no change in the server side and the server is not required to collaborate with the user. However, to claim security guarantee of new image cryptosystems is meaningful only when the cryptanalysis is taken into consideration. The UUP protocol was claimed to be secure; however, a potential collusion attack is pointed out. In order to benefit the advantages and contribution of Castellà-Roca et al.'s scheme, this paper redesigns a security-improved version by simple modification to remove the possible security concern. Precisely, to correct the shortcoming, the authors suggest the user's query be encrypted firstly by means of the server's public key and then each answer also be encrypted by a session key.

Keywords: privacy preserving, web search engine, private information retrieval

具隱私保護的 UUP 網頁搜尋引擎協定之安全探討

鄭富國 [1]　　王炳兼 [2]　　陳宗和*[2]

[1]嘉義大學應用數學系
[2]嘉義大學資訊工程學系

摘　要

一般而言，網路搜尋引擎會紀錄使用的使用動態名為改進效能，然而這樣也違反使用者的個人隱私。因此，Castellà-Roca 等提出 UUP 協定。該協定下，使用者不再需要提供完整的使用資訊，伺服器端得到的並不是某位使用者完整的使用資訊。值得注意的，伺服器端不用修改且不用與使用者配合修改。然而，基於宣稱安全的一個新的密碼技術必須經過安全分析後才有意義，本研究發現該協定有共謀攻擊的可能，當群組中的欺騙者人數高達 n-1 時，最後一位使用將可能受到共謀欺騙。為了維持 Castellà-Roca 等提出 UUP 協定的優點，本文對其進行小而簡單的修改以補強其安全性。主要改進的設計在於使用者先以伺服器的公開金鑰對查詢內容加密，而伺服器回應的內容則以一把會議金鑰加密後傳回。

關鍵詞：隱私保護，網頁搜尋引擎，個人資訊取得

文稿收件日期 106.9.20;文稿修正後接受日期 107.1.30; *通訊作者
Manuscript received September 20, 2017; revised January 30, 2018; * Corresponding author

1. Introduction

In order to protect the confidentiality of sensitive data in outsourcing cloud-computing environments, a well-defined encryption technique is used to encrypt the private and sensitive data stored in the cloud. With issuing a keyword searching on the encrypted data, it unavoidably faces the security problem of how to process the key search without revealing any sensitive information. Especially, the server maintaining the database of encrypted data is not always trusty.

Web search engines can help users to receive a great amount of data they want. However, their private search profiles are possible to be disclosed after submitting queries to a web search engine. This problem of protecting user's privacy can be viewed as a Private Information Retrieval (PIR) problem [8,9,10,11,12]. Generally, a user in a PIR protocol can retrieve a certain amount of data from the database of a server while the server has no idea about which data requested by the user.

In 2004, a public-key encryption with keyword search (PEKS) is first proposed by Boneh et al. [19]. Inspired by Boneh et al.'s scheme, Hwang and Lee [20] proposed another PEKS scheme for multi-receiver in which the concept of proxy re-encryption was later applied in keyword search by Shao et al. [21] and by Yau and Phan [22] as well. Furthermore, Baek et al. [24], on the one hand, demonstrated that outside attackers could perform the test process by collecting the transmitted ciphertexts and trapdoors in the PEKS scheme. Consequentially, attackers are potentially able to further construct the relationship between encrypted data and the given trapdoors of known keywords.

Therefore, Baek et al., on the other hand, proposed their public-key encryption scheme with designated tester (dPEKS) to remove the security problem. Byun et al. [25] presented that Boneh et al.'s design of trapdoors in PEKS suffers off-line keyword-guessing attacks. In such a way, attackers can choose the keywords 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. Unhappily, although Baek et al.'s dPEKS scheme achieves tester designating, the trapdoor's structure is identical to that in PEKS's. In such way, Baek et al.'s dPEKS scheme cannot prevent off-line keyword-guessing attacks. In 2010, Rhee et al. [26] enhanced the trapdoor security so as to prevent from off-line keyword-guessing attacks existing Baek et al.'s dPEKS scheme [24]. Yet, Wang et al. pointed out the trapdoor design was still on the risk of keyword-guessing attacks especially by malicious servers [27]. After that, there are further searchable encryption schemes taking realistic applications into account, for example, a conjunctive subset keywords search proposed by Zhang et al. [23].

Chor et al. [1, 2] firstly introduced the private information retrieval problem and proposed a protocol. In their protocol, several servers share the same database and these servers are not allowed to communicate to each other. But as mentioned above, Castellà-Roca et al. required one server, the web search engine, and one database in their case. Thus, they looked forward to the single-database PIR protocol proposed firstly by Kushilevitz and Ostrovsky [3]. The single-database PIR schemes are more suitable to apply on web search engines. "However, they suffer from some fundamental problems that make their use unfeasible in communications between a user and a web search engine," Castellà-Roca et al. summarized [5] as follows.

(1) The single-database PIR schemes are not suited to deal with large databases. With PIR in mind, the single database is usually modeled as a vector. Upon retrieving the value of the ith component of the vector, users wish to keep the index i hidden from the server holding the database. Supposing that the database contains n items, a PIR scheme aims to guarantee maximum server-uncertainty on the ith record retrieved by a user. It seems to be done by accessing to all records in the database. If some user only accesses to a part of them, the server easily lean to know the real interest of this user. And the cost of accessing all records implies a computational complexity of $O(n)$.

(2) Upon accessing a record in the database, it is reasonably assumed that the user knows its physical location. This assumption is not always realistic because the database is managed by the server. Instead, the user can submit a query consisting on keywords.

(3) Thirdly, it is assumed that the server, holding

the single database, collaborates with users in the PIR protocol. However, the assumption is not realistic since a server has no motivation to protect the privacy of users. In fact, users should take care of their own privacy by themselves instead of expecting any collaboration from the web search engine.

Consequently, Castellà-Roca et al. proposed the UUP protocol to protect the users' privacy by providing a distorted user profile for a web search engine so that the web search engine cannot generate a real profile of a certain individual. Briefly speaking, there is a central node in their scheme grouping n users who submit a query each and shuffling all queries and finally distributing the queries fairly. When a user receives the assigned query, he submits it to the web search engine and waits for the real answer for his own real query. The answers of the n queries from the search engine are broadcast to all the group users; therefore, a user figures out his answer and ignores the others.

Their scheme improves the performance of existing proposals in terms of the computational cost and communication overhead. To avoid a web search engine profiling a real search record of a certain individual, Castellà-Roca et al. applies the technologies of encryption, re-masking and permutation to achieve their goal and make sure their scheme secure. The UUP protocol is proven able to prevent any attack from a dishonest user, a dishonest central node and a dishonest search engine, i.e. three entities in their protocol, under the assumptions that all the group users follow their protocol and no collusions happen between two of the three entities in their scheme.

In addition to the above-mentioned advantages, the main contribution of Castellà-Roca et al.'s UUP protocol is that the UUP protocol does not require any change in the server side and the server is not required to collaborate with the user.

As security is always the concern for new cryptosystems such as the abovementioned PEKS, dPEKS, etc., all proposed cryptosystems must undergo the scrutiny of the scientific community [13-17]. Unhappily, taking the higher security into account, the UUP scheme cannot avoid the insider collusion attacks, in which the group users plan together to cheat the n th user's privacy profile. It is obvious that the $n-1$ group users can collaborate to analyze the queries they had submitted and the answers they had gotten, and further they can infer the nth user's authentic search profile.

In order to benefit the advantages and contribution of Castellà-Roca et al.'s scheme, it is worthwhile to re-design the improved version. As C. A. R. Hoare said, "*There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult.*" [18]

In this paper, the authors firstly show that the potential security weakness, i.e. collusion attacks, exists in Castellà-Roca et al.'s scheme. Precisely, their scheme is secure only if the number of dishonest users is less than n-1. Unfortunately, if there are n-1 dishonest users in the same group, the n-th user encounters the risk of cheating by the others. This security weakness comes from that the n-1 group users can collaborate to analyze the queries they had submitted. Upon they had gotten the responses, and further they can infer the n-th user's authentic search profile.

Secondly, a "small and simple" modification to Castellà-Roca et al.'s scheme is proposed. To correct the shortcoming, the authors suggest the user's query be encrypted firstly by means of the server's public key and then each answer also be encrypted by a session key. Inheriting the contribution from Castellà-Roca et al.'s UUP scheme, the main contribution of this paper is to further enhance the security to avoid the collusion attacks.

The rest of the paper is organized as follows. A review of the UUP protocol is given in Section 2. A security improvement and the security analysis are given in Section 3. The conclusions are given in Section 4.

2. Review of useless user profile (UUP) protocol

The main idea of Castellà-Roca et al.'s scheme relies on that each user who intends to submit a query will not send her/his own but a query of another user instead. Simultaneously, her/his query is submitted by just another group user. Considering privacy concerns, the key design relies on that users do not know which query issued by each user based on the assumption that each user submits very different

kinds of queries. There is no clue about that those queries are liable to a certain person.

With the help of *n*-out-of-*n* threshold ElGamal encryption [6] and ElGamal re-masking operation, Castellà-Roca et al. proposed the UUP protocol to protect user's privacy by providing a distorted user profile for a web search engine so that the web search engine cannot generate a real profile of a certain individual. On the basis of privacy requirements, their scheme achieves the objective because the link between the wanted query a user submitted originally and the true answer the user had is distorted.

The scenario in the UUP protocol contains the three entities:

- Users (U): Users in the group are the individuals who intend to submit queries to the web search engine but still keep protecting their own privacy in mind.
- The central node (C): The central node takes the responsibility to keep in touch all group users intending to submit their query. That is, it groups users in order to execute the UUP protocol.
- The web search engine (W): This web search engine holds the database but is not always trustworthy. It does not guarantee to preserve users' privacy.

Upon considering the privacy requirements of users, the UUP protocol should satisfies the following properties:

- U_i must not link a certain query with U_j who has generated it.
- C must not link a certain query with U_i who has generated it.
- W must be unable to construct a reliable profile of a certain user U_i.

There are four sub-protocols in their scheme. They are group setup, group key generation, anonymous query retrieval, and query submission and retrieval. The purpose of each sub-protocol and how it works are described in the following.

2.1 Group setup

Assume user wants to submit a query to the web search engine. Firstly, he has to send a message to the central node C for asking to be a group member. The central node receives all the requests from users. As soon as it collects n requests, it sets up a new user group $\{U_1, \dots, U_n\}$ and notifies the n users which group they belong to. A communication channel among them is built up at the same time such that they can talk to each other without the interference of the central node.

2.2 Group key generation

First, all the users $\{U_1, \dots, U_n\}$ in the same group agree on a large prime p where $p = 2q + 1$ and q is a prime too. Then they choose a generator element $g \in Z_q^*$ of the multiplicative group.

Next, user U_i randomly generates his private key $\alpha_i \in Z_q^*$ and publishes $y_i = g^{\alpha_i} \bmod q$. Note that each user should keep his private key secret. Finally, all the users $\{U_1, \dots, U_n\}$ execute altogether the n-out-of-n threshold ElGamal encryption to generate their group public key y, where $y = \prod_{i=1}^{n} y_i = g^{\alpha} \bmod q$, and $\alpha = \alpha_1 + \cdots + \alpha_n$.

2.3.Anonymous query retrieval

Firstly, user U_i (for $i = 1, \dots, n$) generates a random value $r_i \in Z_q^*$ and encrypts his query m_i with the group key by means of the standard ElGamal encryption function [7], i.e. $E_y(m_i, r_i) = (g^{r_i}, m_i \cdot y^{r_i}) \bmod q = (c1_i, c2_i) = c_i^0$ Next, user U_i (for $i = 1, \dots, n$) sends his cryptogram c_i^0 to the others in his group. In the end of the sending process, each of the group holds the ordered cryptograms $\{c_1^0, \dots, c_n^0\}$.

Then, user U_1 re-masks the cryptograms $\{c_1^0, \dots, c_n^0\}$, which he already holds, to get a re-encrypted version. Then, user U_1 randomly permutes the re-encrypted version to obtain a re-ordered version of cryptograms. Finally, he sends the re-ordered version of cryptograms to user U_2. Note that it is assumed that the group members are set in order from the first to the nth. Following this way, each of the other users U_i (for $i = 1, \dots, n$) will wait for the re-ordered version of the cryptograms from his immediate predecessor and then goes on the processes of re-masking the cryptograms and randomly permuting the re-encrypted version so as to get a re-ordered version of the cryptograms and finally sending them to the next group member. In the end, User U_n has to broadcast the last

result of the cryptograms $\{e^n_{\sigma(1)}, \dots, e^n_{\sigma(n)}\}$ to all of the group members.

Let $\{e^n_{\sigma(1)}, \dots, e^n_{\sigma(n)}\}$ denote as $\{c_1, \dots c_n\}$. To decrypt the value c_i, user U_i has to require all the other group members to take part by sending their corresponding shares called $(c1_i)^{\alpha_j}$ from user U_i, where $j = 1, \dots, n$ and $j \neq i$. Finally, user U_i can retrieve the query m^i by computing:

$$m^i = \frac{c2_i}{c1_i^{\alpha_i}(\prod_{j \neq i} c1_i^{\alpha_j})} \ mod\ q.$$

Note that the value c_i is correspondent to the query m^i, but the query m^i could be generated by one of the other group members.

2.4 Query submission and retrieval

Once user U_i retrieves the query m^i, he submits it to the web search engine *W*. As soon as he gets the response a^i from the web search engine, he broadcasts it to the other group members. Finally, each user figures out the exact answer from those responses to match his original query.

2.5 Security analysis

Castellà-Roca et al. proposed the UUP protocol by applying the technologies of encryption, re-masking and permutation to preserve the users' privacy when they submit queries to a web search engine. As they defined, a successful attacker is able to know the certain query submitted by a certain user. Their scheme is proven able to prevent any attack from a dishonest user, a dishonest central node and a dishonest search engine, i.e. three entities in their protocol, under the assumptions that all the group users follow their protocol and no collusions happen between two of the three entities in their scheme. And the attackers from external entities cannot get more information than those from the internal entities. Hence, Castellà-Roca et al. perform the security analysis for the internal entities as follows.

2.5.1 Dishonest user

User U_a is supposed to be dishonest. In the end of the cryptogram-sending process, he gets the original ordered cryptograms, which contains all the queries from the group members. To decrypt the cryptograms, user U_a has to require all the other group members to take part by sending their corresponding shares called $(c1_i)^{\alpha_j}$. Provided that all the other group members contribute their secret keys $(\alpha_1, \dots, \alpha_n)$, he is not able to decrypt the cryptograms $\{c_1^0, \dots, c_n^0\}$. Therefore, their scheme is secure if there is one dishonest user in a group.

2.5.2 Dishonest central node

The job of a central node *C* is to receive the user's request of being a part of a group and to set up a new group if the number for a group is met. Once a communication channel among the group members is established, it will leave them alone and has no business with the group members any more. Therefore, the central node cannot link any query to any user.

2.5.3 Dishonest web search engine

User U_i submits the assigned query m^i. When the web search engine receives the query, it makes a link between the query m^i and user U_i. Obviously, the web search engine builds a distorted user profile. Because of the re-masking operation and permutation steps, there is less possibility of the query m^i in correspondence to the original query m_i submitted by user U_i. Therefore, the search engine has a useless profile of user U_i.

3. Security improvement

From the security analysis stated above, their scheme is proven able to prevent any attack from a dishonest user, a dishonest central node and a dishonest search engine, and even more, their scheme can be secure if the number of dishonest users is less than *n*-1.

But if there are *n*-1 dishonest users in the same group, the *n*-th user will encounter the risk of being cheated by them. It is obvious that the *n*-1 group users can collaborate to analyze the queries they had submitted and the answers they had gotten, and further they can infer the *n*-th user's authentic search profile. Please note that each of the group members can get all the answers to their queries in the final step. For avoiding this kind of collusion attack, we make some security improvements on the UUP protocol.

3.1 The proposed improvement

The proposed improved version encompasses four sub-protocols. The first two, i.e., group setup, and group key generation, are the same as those in Castellà-Roca et al.'s scheme. Thus, they are omitted here. The other two, including anonymous query retrieval, and query submission and retrieval, are described as follows. Figure 1 demonstrates the operations in the anonymous query retrieval phase.

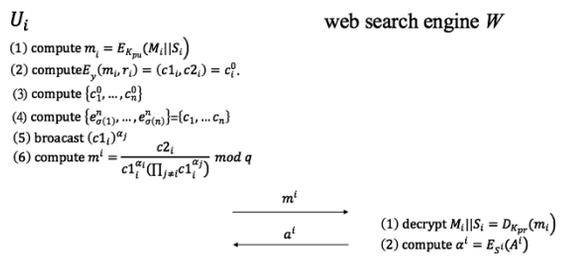

Figure 1: The operations in anonymous query retrieval

3.1.1.Anonymous query retrieval

First, we define M_i to be the real query submitted by user U_i and S_i to be a secret key selected by user U_i. We redefine m_i to be a ciphertext encrypted by the public key of a web search engine, where

$$m_i = E_{K_{pu}}(M_i||S_i). \quad (1)$$

User U_i (for $i = 1, \dots, n$) generates a random value $r_i \in Z_q^*$ and encrypts his query m_i mentioned above with the group key by means of $E_y(m_i, r_i) = (g^{r_i}, m_i \cdot y^{r_i}) \ mod\ q = (c1_i, c2_i) = c_i^0$. Next, user U_i (for $i = 1, \dots, n$) sends his cryptogram c_i^0 to the others in his group. In the end of the sending process, each of the group holds the ordered cryptograms $\{c_1^0, \dots, c_n^0\}$. Then, user U_1 re-masks the cryptograms $\{c_1^0, \dots, c_n^0\}$ to get a re-encrypted version. Then, user U_1 randomly permutes the re-encrypted version to obtain a re-ordered version of cryptograms. Finally, he sends the re-ordered version of cryptograms to user U_2. Following the way in Section 2.3, in the end, User U_n has to broadcast the last result of the cryptograms $\{c_1, \dots c_n\}$ to all of the group members.

To decrypt the value c_i, user U_i has to require all the other group members to take part by sending their corresponding shares $(c1_i)^{\alpha_j}$. Finally, user U_i retrieves the query m^i by computing:

$$m^i = \frac{c2_i}{c1_i^{\alpha_i}(\prod_{j\neq i} c1_i^{\alpha_j})} \ mod\ q \quad \text{where} \quad m^i = E_{K_{pu}}(M^i||S^i).$$

3.1.2 Query submission and retrieval

Once user U_i retrieves the query m^i, (s)he submits it to the web search engine *W*. When the search engine receives the query, it uses its secret key to decrypt it so as to get M^i and S^i. The answer A^i to query M^i is encrypted by the selected secret key S^i.

We denote the encrypted answer as

$$\alpha^i = E_{S^i}(A^i). \quad (2)$$

Similar to the concept of the UUP protocol, the web search engine has no idea about which user is the original generator of the query M^i and selected secret keyS^i. Thus, user U_i receives a^i from the web search engine and broadcasts it to the rest of the group members.

At last, user U_i uses his selected secret key S_i to decrypt all of the encrypted answers to figure out the real answer to his real query.

3.2 Security analysis

The security analysis of (*n*-1)-collusion-attack-free is given first.

Definition 1 (Collusion-attack-free).
(*n*-1)-collusion-attack-free is defined as if (*n*-1) dishonest users in a group with *n* participants has no feasible way to infer the *n*th user's authentic search profile by analyzing the queries they had submitted and the answers they had gotten.

Proposition 1: The improved scheme is (*n*-1)-collusion-attack-free.

Poof.

Each participant in a group has all $m_i = E_{K_{pu}}(M_i||S_i)$ encrypted using the public key of a web search engine by Eq. (1) and all $\alpha^i = E_{S^i}(A^i)$ encrypted using participant's secret key from the search engine by Eq. (2). In such a way, even if *n*-1 collusion attackers in a group have no feasible way to deduce the *n*th user's authentic search profile without the keys to decrypt all m_i and a^i. Precisely, the *n*-1 group

participants can not collaborate to infer the *n*-th user's authentic search profile as each of the group members can only obtain their individual answer to their queries in the final step.

Thus, the improved scheme (*n*-1)-collusion-attack-free. □

Proposition 2: The improved scheme is secure even if there are dishonest users in the group.
Poof.
Suppose U_a is dishonest. In the end of the cryptogram-sending process, (s)he has containing all the queries. In order to decrypt the cryptograms, U_a must ask all the other group members to take part in by sending the shares $(c1_i)^{\alpha_j}$. With all the other group members contribute their secret keys $(\alpha_1, \dots, \alpha_n)$, (s)he is not able to decrypt the cryptograms $\{c_1^0, \dots, c_n^0\}$. Therefore, their scheme is secure if there is one dishonest user in a group. Once there are more than one dishonest users, the improved scheme is still secure by **Proposition 1.** □

Proposition 3: The improved scheme is secure even if the central node is not honest in the group.
Poof.
The proof is the same as that in Section 2.5.2 and thus omitted here. □

Proposition 4: The improved scheme is secure even if the web search engine is not honest in the group.
Poof.
The proof is the same as that in Section 2.5.3 and thus omitted here. □

4. Conclusion

In this paper, only the web search engine can read the query M_i as it can use its secret key to decrypt the ciphertext m_i; however, there is no link between the real query and the real generator of the query. Moreover, only the original query generator can decrypt and figure out the real answer and read it. The collaboration of *n-1* group users only can derive a profile of encrypted answers for the *n*th group member. The improvement relies on the redesigned that the user's query is encrypted by means of the server's public key and then each answer is encrypted by a session key. Therefore, the security improvement proposed here can achieve the privacy requirements as Castellà-Roca et al. stated and further it can avoid the collusion attack from group members as well.

Acknowledgment

This work was partially supported by Ministry of Science and Technology, Taiwan, R.O.C., under contract by MOST 105-2221-E-415 -012 -.

References

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," Proceedings of the 36th Annual Symposium on Foundations of Computer Science, pp. 41-50, 1995.

[2] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," Journal of the ACM, Vol.45, Issue 6, pp. 965-981, 1998.

[3] E. Kushilevitz, and R. Ostrovsky, "Replication is not needed: single database, computationally-private information retrieval," Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, pp. 364-373, 1997.

[4] R. Ostrovsky, and W.E. Skeith III, "A survey of single-database PIR: techniques and applications," Lecture Notes in Computer Science, Vol. 4450, pp. 393-411, 2007.

[5] J. Castella-Roca, A. Viejo, and J. Herrera-Joancomarti, "Preserving user's privacy in web search engines," Computer Communications, Vol. 32, Issues 13-14, pp. 1541-1551, 2009.

[6] Y. Desmedt, and Y. Frankel, "Threshold cryptosystems," Advances in Cryptology, Lecture Notes in Computer Science, Vol. 335, pp. 307-315, 1990.

[7] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," IEEE Transactions on Information Theory, Vol. 31, pp. 469-472, 1958.

[8] S. Yekhanin, "Private information retrieval," Communications of the ACM, Vol. 53, pp. 68-73, 2010.

[9] S. Jarecki, C. Jutla, H. Krawczyk, M. Rosu, and M. Steiner, "Outsourced Symmetric Private Information Retrieval," Proceedings

of the 2013 ACM SIGSAC conference on Computer & communications security, pp. 875-888, 2013.

[10] E. Unal and E. Savas, "On Acceleration and Scalability of Number Theoretic Private Information Retrieval," IEEE Transactions on Parallel and Distributed Systems, Vol. 27, pp. 1727-1741, 2016.

[11] H. Sun, and S. A. Jafar, "The capacity of private information retrieval," to appear IEEE Transactions on Information Theory, 2017.

[12] Z. Li, C. Ma, D. Wang, and G. Du, "Toward single-server private information retrieval protocol via learning with errors," Journal of Information Security and Applications, available online 13 December, 2016.

[13] J. Liu, and J. Bi, "Cryptanalysis of a Fast Private Information Retrieval Protocol," Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography, pp. 56-60, 2016.

[14] T. Yang, B. Yu, H. Wang, J. Li, and Z. Lv, "Cryptanalysis and improvement of Panda-public auditing for shared data in cloud and internet of things," Multimedia Tools and Applications, available online 8 December, 2015.

[15] Y. Lu, L. Li, H. Peng, and Y. Yang, "Cryptanalysis and improvement of a chaotic maps-based anonymous authenticated key agreement protocol for multi-server architecture," Security and Communication Networks, Vol. 9, pp. 1321-1330, 2016.

[16] J. Jia, J. Liu, and H. Zhang, "Cryptanalysis of cryptosystems based on general linear group," China Communications, Vol. 13, pp. 217-224, 2016.

[17] F.G. Jeng, W.L. Huang, and T.H. Chen, "Cryptanalysis and improvement of two hyper-chaos-based image encryption schemes," Signal Processing: Image Communication, Vo. 34, pp. 45-51, 2015.

[18] https://en.wikiquote.org/wiki/C._A._R._Hoare.

[19] D. Boneh, G.D. Crescenzo, R. Ostrovsky, G. Persiano, "Public key encryption with keyword search," Proceedings of EUROCRYPT'04. LNCS, Vol. 3027, 2004, pp. 506–522.

[20] Y.H. Hwang, P.J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," Proceedings of Pairing 2007, LNCS, Vol. 4575, 2007, pp. 2–22.

[21] J. Shao, Z.F. Cao, X.H. Liang, H. Lin, "Proxy re-encryption with keyword search," Information Sciences, Vol. 180, Issue 13, 1 July 2010, pp. 2576-2587.

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

[23] B. Zhang, F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search," Journal of Network and Computer Applications, Vol. 34, Issue 1, January 2011, pp. 262-267.

[24] J. Baek, R. Safavi-Naini, W. Susilo, "Public key encryption with keyword search revisited," Proceedings of ACIS'06, 2006.

[25] J.W. Byun, H.S. Rhee, H.A. Park, D.H. Lee, "Off -line keyword guessing attacks on recent keyword search schemes over encrypted data," Proceedings of SDM'06. LNCS, Vol. 4165, 2006, pp. 75–83.

[26] H.S. Rhee, J.H. Park, W. Susilo, D.H. Lee, "Trapdoor security in a searchable public-key encryption scheme with a designated tester," Journal of Systems and Software, Vol. 83, Issue 5, May 2010, pp. 763-771.

[27] B.J. Wang., T.T. Chen., and F.G. Jeng, "Security improvement against malicious server's attack for a dPEKS Scheme," Proceedings of International Journal of Information and Education Technology, Vol. 1, Issue 4, 350, 2011.