MORE ROBUST PRIVATE INFORMATION RETRIEVAL SCHEME

Chun-Hua Chen[1,2], Gwoboa Horng[1]
[1]*Institute of Computer Science, National Chung-Hsing University, No. 250,Kuo Kuang Road,Taichung, Taiwan, R.O.C.*

[2]*Department of Electronic Engineering, Chienkuo Technology University, No. 1, Chieh Sou North Road, Changhua City, Taiwan, R. O. C.*

Keywords: Private information retrieval (PIR), Mutual authentication, DSA algorithm, Secure coprocessor (SC).

Abstract: In e-commerce, the protection of users' privacy from a server was not considered feasible until the private information retrieval (PIR) problem was stated and solved. A PIR scheme allows a user to retrieve a data item from an online database while hiding the identity of the item from a database server. In this paper, a new PIR scheme using a secure coprocessor (SC) and including mutual authentication by DSA signature algorithm for protecting the privacy of users, is proposed. Because of using only one server and including the mutual authentication process in the proposed scheme, it is more efficient and more robust (secure) in the real e-commerce environment compared with previous PIR solutions. In addition, a security analysis (proof) for the proposed scheme and comparisons to other PIR schemes are given.

1 INTRODUCTION

1.1 Motivation

Nowadays, knowledge about user preferences is important and valuable. This information may often play a negative role if it is used against the user. The assumption, that the server will not employ user preferences against the user, has been taken as an assumption for a long time. However, there is no reason for such an assumption. The solutions for the private information retrieval (PIR) problem would make it possible for a user to keep his preferences private from everybody including the server. The thought mentioned above is very reasonable in the e-commerce environment. The following two examples are given:

(1) Patent Databases:

About the patent database query, if the patent server knows which patent the user is interested in, this will cause a lot of problems. Imagine that some scientist discovers a science formula, for example "$H_2 + O_2 \Rightarrow H_2O$". Naturally, he wants to patent it, because it may be valuable in the industry. But first, he checks at an international patent database to see whether the same or similar patent already exists. If the user's privacy is not secret to the server, the administrator of that server will know the scientist's query. Then the administrator of that server may gain a lot of profit from the information. PIR schemes solve this problem, the user may query a patent and the server will not know which patent the user just queried.

(2) Pharmaceutical Databases:

Usually, pharmaceutical companies are specialized either in inventing drugs or in gathering information about the basic components and their properties. The process of synthesizing a new drug requires information on several basic components from this pharmaceutical database. To hide the plans of the company, drug designers buy the entire pharmaceutical database. These big expenses can be avoided if the designers use a PIR scheme to query only the information about a few basic components needed.

1.2 Private Information Retrieval

Formally, private information retrieval (PIR) is a general problem for private retrieval of the i-item out of an n-item database stored at the server. "Private" means that the server does not know about i, that is, the server does not learn which item the client is interested in, in the process of the query. Initial research of PIR was done by Chor et al. (Chor, 1995), and then it became the topic of a significant amount of research work. By replicating databases

Chen C. and Horng G. (2006).
MORE ROBUST PRIVATE INFORMATION RETRIEVAL SCHEME.
In *Proceedings of the International Conference on Security and Cryptography*, pages 297-302
DOI: 10.5220/0002097102970302

on separated servers and limiting the communication's capability of replicated database servers (that is, the servers cannot collude), the PIR scheme (Chor, 1995, 1998) is able to protect the users' privacy.

The communication complexity (between the user and the server) of retrieving one out of n bits is one way to measure the costs of PIR schemes. It has been proven in (Chor, 1998) that the communication complexity in information-theoretic privacy of one-server scheme is O(n). The "n" is the size of the database. Through using the k-server scheme, the communication complexity of a PIR scheme was improved to $O(n^{1/k})$ (Chor, 1995). Some subsequent studies of PIR were focused on reducing the complexity. Ambainis improved the communication complexity to $O(n^{1/(2k-1)})$ in (Ambainis, 1997). Beimel et al. (Beimel, 2002, FOCS) broke the barrier $O(n^{1/(2k-1)})$ of communication complexity for information-theoretic PIR. The server computations of all the above-mentioned protocols are at least O(n). Beimel et al. proposed the protocol of PIR with pre-processing (Beimel, 2004). Before the execution of the protocol, the server may compute and store the information regarding the database. Later on, this information should enable the server to answer the query of the user with more efficient computation. The server's computation complexity of this protocol (using k server) is $O(n / (\log^{2k-2} n))$.

The standard definition of PIR schemes (Chor, 1998) raises a simple question – what happens if some servers crash during the operation? Current systems do not guarantee availability of servers at all times for many reasons, e.g., crash of server or communication problems. Beimel et al. proposed several robust PIR schemes in (Beimel, 2002) to solve the problem. Yang et al. presented a fault-tolerant scheme in (Yang, 2002) to tolerate malicious server failures. These PIR schemes use an organization including L replicated copies of a database ($L > k \geq 2$) in computer network. It results in heavy overheads for managing these database servers, including keeping them with one accord. It is not practical from an implementation viewpoint.

From a mathematical viewpoint, the PIR schemes mentioned above are excellent research work. But from the implementation viewpoint, the existing PIR schemes have some limitations and constraints in their practical feasibility in real-world applications.

1.3 Results

We address the PIR problem of heavy overheads for managing multiple servers mentioned in section 1.2. A new one-server PIR scheme, with mutual authentication between the user and the server, is proposed to provide privacy protection for online users in the e-commerce environment. The major contributions of this paper are as follows:

(1) The proposed scheme is more practical (more robust and more efficient) than previous PIR schemes in the e-commerce environment. Some comparisons are provided in Section 4.

(2) The proposed scheme has mutual authentication and key agreement process, which makes it more robust in security than that in (Smith, 2001; Asonov, 2003). The analysis of security is provided in Section 3.

2 RELATED WORK

2.1 Computational Private Information Retrieval

To improve the communication complexity, Chor et al. introduced the notation of a computational PIR (CPIR) scheme (Chor, 1997) that lowers the privacy security (from information-theoretic security to computational security) for improving the complexity of a PIR scheme. Kushilevitz et al. proposed a CPIR scheme (Kushilevitz, 1997) based on the quadratic residuosity assumption with $O(n^{\varepsilon})$ communication complexity. Cachin et al. proposed a CPIR scheme (Cachin, 1999) with the poly-logarithm communication complexity —O(logn) which is based on the Φ-Hiding Assumption : essentially the difficulty of deciding whether a small prime divides $\Phi(m)$, where m is a big composite integer of unknown factorizing.

Although CPIR schemes break the O(n) communication complexity of one server, the computation of the server is still O(n). In addition, CPIR schemes of one server can only deal with one bit per query. This is the most serious flaw of CPIR schemes.

2.2 Private Information Retrieval Using a Secure Coprocessor (SC)

Smith et al. (Smith, 2001) used a secure coprocessor (SC) in their PIR solution. An SC is a temper-proof device with small memory in it; it is designed to prevent anybody (including the server) from accessing its memory. Unlike the previous PIR papers, which concentrated on the theory and mathematical model, Smith et al. focus on real world applications. The operations of Smith's scheme are shown in Fig. 1. The user encrypts the query "I need the i-th record" with a public key of the SC of the server, and sends it to the server. The SC receives

the encrypted query and decrypts it, and then reads all records from the database, but leaves in its memory the i-th record only. Finally, the SC in the server encrypts the record and sends it to the user. This PIR scheme conquers the problem of CPIR which can only deal with one bit per query, and improves the communication complexity to O(1), but the server's computation complexity is still O(n). Iliev et al. (Iliev, 2005) use the concept (PIR using secure coprocessor in server) on the topic: protecting client privacy with trusted computing at the server, because previous solutions usually put physically secure hardware on users' machines, potentially violating user privacy.

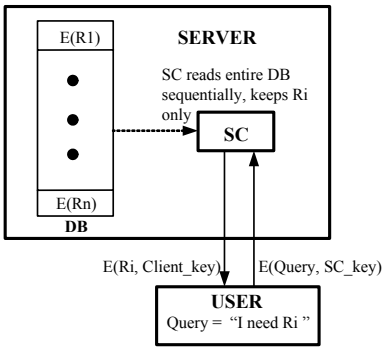

Figure 1: PIR scheme of Smith.

Asonov et al. (Asonov, 2003) proposed another PIR scheme using an SC. They improve Smith's scheme by shuffling the database offline (the shuffling algorithm can be found in (Asonov, 2003)). In the preprocessing phase, the SC computes a shuffled index by the algorithm described in (Knuth, 1981) and computes the random permutation of the records by the shuffled index and stores this permutation (include the shuffled index) in an encrypted form. In the processing phase, the operations of Asonov's are similar to those of Smith's, but improve the computation complexity to O (k), k is a constant. When the SC in the server receives the query "I need the i-th record" from the user, the SC does not need to read the entire database. Instead, the SC accesses the desired encrypted record directly, because the SC knows the shuffled index. Then the encrypted record is decrypted inside the SC, encrypted with the user's key and sent to the user. But for the reason of confusing the server, in the kth query, the SC must read previously accessed records, and one unread record. So, the server's computation complexity to is O (k), when k is a constant, that is O (1). The algorithm of processing kth query can be seen in (Asonov, 2003). The operations of Asonov's scheme are shown in Fig 2

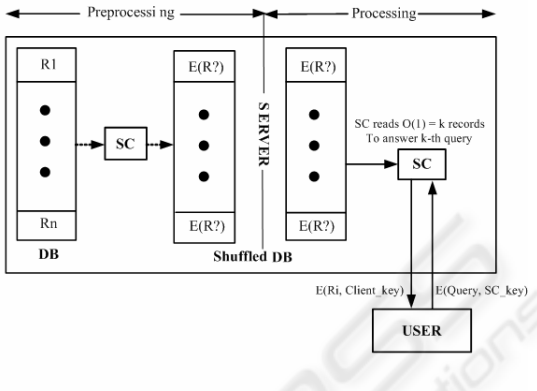

Figure 2: PIR scheme of Asonov.

Smith's PIR scheme and Asonov's PIR scheme make PIR solutions more practical, and the communication complexity of their schemes is O(1). But from the viewpoint of information security, there are some security leaks in the communication between the SC (in the server) and the user in their schemes. In this paper, a new PIR scheme is proposed, which considers the authentication and the key agreement between the SC and the user, is more robust (in security) than both Smith's PIR and Asonov's PIR schemes.

3 THE PROPOSED SCHEME

In this scheme, there are three phases: registering phase, preprocessing phase and online-query phase. Suppose that the public key and private key of the SC in the server are announced before the three phases started. The operations of the proposed scheme are shown in Fig 3.

Firstly, some symbols are defined before describing the scheme in detail. We use p and q as the symbols for a large prime number (512 ~ 1024-bit prime number p, 160-bit prime number q such that $q|p$-1). Let ID_u be the identification number of user U. Let x_u ($1 < x_u < q$-1) be the private key of user U, then y_u ($y_u = g^{xu} \bmod p$) be the public key of user U. The SC in Server S has a public key PK_{SC} and a corresponding private key SK_{SC}. Let $E_{PKsc}( )$ denote an encryption function with the public key PKsc, and $D_{SKsc}( )$ be the corresponding decryption function with the private key SKsc. Also, let $E()$ and $D()$ denote encryption and decryption function with a symmetric key. Let r_u be the random number chosen by user U and r_s be the random number

chosen by the SC in server S. Let K_{su} be the session key (a kind of symmetric key) in one PIR query and it is calculated by r_s ô r_u. We use *h* (.) as the symbol of some collision resistant hash function that map $\{0, 1\}^*$ to the set $\{1, 2, \ldots, q\text{-}1\}$. The framework figure of the proposed scheme is shown in Fig. 3.

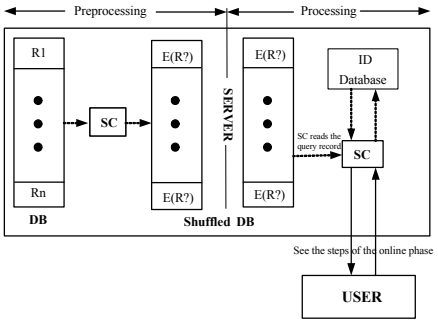

Figure 3: The proposed one-server PIR scheme.

1. Registering phase:
Before a legal user U can query the database on the server, he/she must register on Server S first.
(1) User U chooses an ID_U as the identification number of user U and *p, q* two big prime numbers (512 ~ 1024-bit prime number *p,* 160-bit prime number *q* such that $q|p\text{-}1$). Selects an ordered *q* primitive root *g* in Z_p^* and $g \neq 1$.
(2) User U chooses x_u as the private key and public key $y_u = g^{xu} \bmod p$.
(3) User U computes $C_1 = E_{PKsc}(ID_U, y_u)$ and sends C_1 to the SC in Server S.
(4) On receiving C_1, the SC decrypts C_1 with its private key SK_{SC} and then stores (ID_U, y_u) to the ID file in server S.
(5) The numbers *p, q* and *g* are published and can be used by a group of users.

2. Preprocessing phase:
The SC in server S executes the preprocessing phase periodically. The major function of the preprocessing phase is to produce a shuffled copy of DB in server S and a shuffled index in the SC. The shuffle function that provides a shuffled index is constructed in accordance with (Knuth, 1981), Sec. 3.4.2. The shuffling algorithm can be found in (Asonov, 2003).

3. Online-query phase:
(1) User U selects a random number r_u (a part of session key) and sends $C_2 = (ID_U, E_{PKsc}(r_u))$ to the SC in Server S.
(2) The SC in the server decrypts C_2 with its private key SK_{SC} to get ID_U and r_u.
(3) The SC selects a random number r_s (another part of session key) and calculates the session key K= Ksu = r_s ô r_u. And then sends $C_3=(r_s, E_K(r_u))$ to user U.
(4) User U calculates the session key K'=Kus = r_u ô r_s and decrypts the $E_K(r_u)$ with K'. If the result is equal to the r_u then user U sends $E_{K'}$(Query) to the SC, else stops the online-query phase because the server S (with the SC in it) does not pass the authentication by user U.
(5) User U selects a random number *k* in Zq, then calculates *r,s* and M where $M = E_{K'}(ID_U, r_s, r_u)$, $r=g^k \pmod p \pmod q$ and $s=k^{-1}\times(h(M)+ x_u r) \pmod q$. Then user U sends $C_4 = (r, s, M)$ to the SC.
(6) The SC in Server S calculates $t= h(M)\times s^{-1} \pmod q$ and $u=r\times s^{-1} \pmod q$. Then checks whether $1 \leqq r \leqq q\text{-}1$, $1 \leqq s \leqq q\text{-}1$ and $r = g^t\times(y_u)^u \pmod p \pmod q$. If the answer is correct then goes to step (7), else stops the online-query phase because user U does not pass the authentication by the SC of server S.
(7) The SC in server S reads the Ri from the shuffled database according to the shuffled index (detail algorithm can be seen in (Asonov, 2003)) and sends $E_K(Ri)$ to user U.
(8) User U decrypts $E_K(Ri)$ with K' to get the Ri which he/she queries.

4 SECURITY ANALYSIS OF THE PROPOSED SCHEME

In the following, Section 4.1 proves that the proposed scheme is a mutual authentication scheme between the user and the server. Section 4.2 proves that the proposed scheme is a secure scheme.

4.1 The Proposed Scheme is a Mutual Authentication Scheme

Lemma 1. The proposed scheme correctly authenticates a legal user U.
Proof. If user U is a legal user, he/she knows the private key x_u (including the the public key y_u). So, User U can calculates *r, s* and M in step (5) of the online-query phase, where $M = E_{K'}(ID_U, r_s, r_u)$, $r=g^k \pmod p \pmod q$, and $s=k^{-1}\times(h(M)+ x_u r) \pmod q$. Then user U sends $C_4 = (r, s, M)$ to the SC in server S which can be authenticated successfully by checking the correctness of the equations, $1 \leqq r \leqq q\text{-}1$, $1 \leqq s \leqq q\text{-}1$ and $r = g^t\times(y_u)^u \pmod p \pmod q$, where $t=h(M)\times s^{-1} \pmod q$ and $u=r\times s^{-1} \pmod q$. Thus the SC in server S successfully authenticates user U in step (6) of the online-query phase.

If an adversary *E* wants to impersonate some legal user U, but he/she does not know the private key x_u. He/she can get the information ID_U in some way. By the way, the public key y_u and the numbers *p*, *q* and *g* are published. Suppose *E* can successfully impersonate user U, that is, *E* can generate C4'

(C4'= (r, s', M)) where $s' = k^{-1} \times (h(M) + x_E \times r) \pmod q$ in step (5) of the online-query phase such that $r = g^{t'} \times (y_u)^{u'} \pmod p \pmod q$, where $t' = h(M) \times s'^{-1} \pmod q$ and $u' = r \times s'^{-1} \pmod q$. Then E can be authenticated successfully in step (6) of the online-query phase. Thus, from the verification formula ($r = g^{t'} \times (y_u)^{u'}$), we can get

$$g^{(h(M)+ x_u \times r) \times s^{r-1}} \equiv r \equiv g^{k} \equiv g^{(h(M)+ x_u \times r) \times s^{-1}}$$

(mod p) (mod q) ➔ $(h(M)+ x_u r) \times s'^{-1} \equiv (h(M)+ x_u r) \times s^{-1} \bmod q$ ➔ $s'^{-1} \equiv s^{-1} \bmod q$ ➔ $s'^{-1} = s^{-1}$ (because of $1 \leqq s, s' \leqq q-1$) ➔ $s' = s$. From the definition of s and s', we can get $k^{-1} \times (h(M)+ x_E \times r) \equiv k^{-1} \times (h(M)+ x_u \times r) \pmod q$ ➔ $(h(M)+ x_E \times r) \equiv (h(M)+ x_u \times r) \pmod q$ ➔ $x_E = x_u$ (because of $1 \leqq r, x_E, x_u \leqq q-1$). So, if the adversary E can generate correct s', then he/she knows x_u or he/she can guess x_E ($=x_u$) from y_u. Because E is not user U, he/she does not know the private key x_u. Thus he/she can guess x_E ($=x_u$) from y_u ($y_u = g^{xu} \bmod p$). This conclusion contradicts the intractable assumption of discrete logarithms problem. Therefore, if the SC in server S successfully authenticates the user U, then U knows the private key x_u. □

Lemma 2. The proposed scheme correctly authenticates Server S (with the SC in it).

Proof. If the SC in Server S knows the secret key SK_{SC}, then the SC can decrypt C_2 to obtain r_u and calculate the session key $Ksu = r_s \oplus r_u$. On receiving r_s, user U calculates the session key $Kus = r_u \oplus r_s$ using the r_u chosen by him/her. Thus, the session keys Ksu and Kus are the same value. So, in this situation, user U successfully authenticates Server S (with the SC in it).

With overwhelming probability, the SC knows the secret key SKsc, if user U authenticates the SC in Server S as legal. Namely, only the SC can decrypt C_2 to obtain r_u. This result is derived from the security of the encryption functions $E_{PKsc}( )$ which is assumed to be secure against the adaptive chosen ciphertext attack (Rackoff, 1991; Dolev, 1991; Bellare, 1998). Therefore, Server S is successfully authenticated by user U if and only if the SC in Server S knows the private key SKsc. □

Theorem 3. The proposed scheme is a mutual authentication scheme.

Proof. This can be derived immediately from Lemma 1 and Lemma 2. □

4.2 The Proposed Scheme is a Secure Scheme

The security of message transformation between the user and the server is analyzed in this section. Assume that an adversary can control over the communication channels and is told the previous session key. In the proposed scheme, the session key is used (once in some query) to protect the security of the message. The session key is produced by the process of key exchange. A key exchange scheme is secure if the following requirements are satisfied (Bellare, 1993; Canetti, 2001):

(1) If both participants honestly execute the scheme then the session key is K=Ksu = Kus.

(2) No one can calculate the session key except participants U and Server S.

(3) The session key is indistinguishable from a truly random number.

Lemma 4. The proposed scheme satisfies the first security requirement.

Proof. After mutual authentication, both participants have agreed on the random number $r_s \oplus r_u$ by Lemma 1 and Lemma 2. Therefore, $K = Ksu = r_s \oplus r_u = r_u \oplus r_s = Kus = K'$. □

Lemma 5. The proposed scheme satisfies the second security requirement.

Proof. The random number r_u is selected by user U and is encrypted by the encryption function $E_{PKsc}( )$. The encryption function $E_{PKsc}( )$ is secure and can only be decrypted by the SC in Sever S. The random number r_s is selected by the SC and is sent to user U in step (3) of the online-query phase. Therefore, only the participants U and the SC in Server S can calculate the session key $K(= Ksu = r_s \oplus r_u = r_u \oplus r_s = Kus = K')$. □

Lemma 6. The proposed scheme satisfies the third security requirement.

Proof. Because r_u, r_s are two random numbers selected by user U and the SC in Server S. The session key $K(= Ksu = r_s \oplus r_u = r_u \oplus r_s = Kus = K')$ is also a random number. □

Theorem 7. The proposed scheme is a secure scheme.

Proof. This can be derived immediately from Lemmas 4, 5 and 6. □

5 COMPARISONS AND CONCLUSIONS

In this paper, a one-server PIR scheme using a secure coprocessor (SC) is presented which avoids the large management overheads of multi-servers. The proposed scheme has an optimal communication complexity and an optimal computation complexity of O(1). And it has a mutual authentication process (by DSA algorithm) and a key agreement between the server and the user, which makes it more robust in security in the e-commerce environment.

The proposed scheme is a good scheme in private information retrieval. We think it can not only apply

in the e-commerce environment, but also other applications which need privacy in the internet.

ACKNOWLEDGEMENTS

We are grateful to Dr. Fuw-Yi Yang for useful discussions. And we also thank Prof. J. Buchmann in Darmstadt University of Technology (Germany) for providing good research environment. This paper was completed in the visiting time to Prof. J. Buchmann's research group.

REFERENCES

Ambainis, A., 1997. Upper bound on the communication complexity of private information retrieval. In *Proc. of 24th ICALP 97*, *Lecture Notes in Computer Science*, 1256, 401-407.

Asonov, D. & Freytag, J.-C., 2003. Almost optimal private information retrieval. In *Pre- and Postproc. of 2nd Workshop on Privacy Enhancing Technologies (PET2002)*, San Francisco, USA; *Lecture Notes in Computer Science*, 2482, 209-223.

Beimel, A. & Ishai, Y., 2001. Information-theoretic private information retrieval: a unified construction. *Electronic Colloquium on Computational Complexity*, TR01-15, 2001; Extended abstract in: *ICALP 2001, Lecture Notes in Computer Science*, 2076, 89-98.

Beimel, A. & Ishai, Y., 2002. Robust information-theoretic private information retrieval. In *Proc. of the 3rd Conference on Security in Communication Networks*, *Lecture Notes in Computer Science*, 2576, 326-341.

Beimel, A., Stahl, Y., Kushilevit, E. & Raymond, J. F., 2002. Breaking the barrier $O(n^{1/(2k-1)})$ for information-theoretic private information retrieval. In *Proc. of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS)*, 261-270.

Beimel, A., Ishai,Y. & Malkin, T., 2004. Reducing the servers computation in private information retrieval: PIR with preprocessing. *Journal of Cryptology*, 17, 125-151.

Bellare, M. & Rogaway, P., 1993. Entity authentication and key distribution, *Advances in Cryptology-CRYPTO'93, Lecture Notes in Computer Science*, 773, 232-249.

Bellare, M., Desai, A., Pointcheval, D. & Rogaway, P., 1998. Relations among notions of security for public key encryption schemes. *Advances in Cryptology CRYPTO'98*, *Lecture Notes in Computer Science*, 1462, 26-46.

Cachin, C., Micali, S. & Stadler, M., 1999. Computationally private information retrieval with polylogarithmic communication. *Eurocrypt 99, Lecture Notes in Computer Science*, 1592, 402-414.

Canetti, R. & Krawczyk, H., 2001. Analysis of key-exchange protocols and their use for building secure channels. *Advances in Cryptology-Eurocrypt'01, Lecture Notes in Computer Science*, 2045, 453-474.

Chor, B., Goldreich, O., Kushilevitz, E. & Sudan, M., 1995. Private information retrieval. In *Proc. of the 36 th IEEE Symposium on Foundations of Computer Science (FOCS)*, 41-50.

Chor, B., Goldreich, O., Kushilevitz, E. & Sudan, M., 1998. Private information retrieval. *Journal of ACM*, 45, 965-981.

Chor, B. & Gilboa, N., 1997. Computationally private information retrieval. In *Proc. of the twenty-ninth annual ACM symposium on Theory of computing (STOC)*, 304-313.

Dolev, D., Dwork, C. & Naor, M., 1991. Non-malleable cryptography. *In Proc. of the twenty third annual ACM symposium on theory of computing(STOC), ACM press*, 542-552.

Iliev, A. & Smith, S.W., 2005. Protecting client privacy with trusted computing at the server. *Security and Privacy Magazine, IEEE,* 3(2), 20-28.

Knuth, D. E., 1981. *The Art of Computer Programming,* vol. 2, Addison-Wesley. Second edition.

Kushilevitz, E. & Ostrovsky, R., 1997. Replication is not needed: single database, computationally-private information retrieval. *In Proc. of the 38th IEEE Symposium on Foundations of Computer Science (FOCS)*, 364-373.

Rackoff, C. & Simon, D., 1991. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. *Advances in Cryptology- Crypto'91, Lecture Notes in Computer Science*, 576, 433-444.

Smith, S.W. & Safford, D., 2001. Practical server privacy using secure coprocessors. *IBM System Journal*, 40(3), 683-695.

Yang, E. Y., Xu, J. & Bennett, K. H., 2002. Private information retrieval in the presence of malicious failures. *In Proc. of the 26th IEEE Annual International Computer Software and Applications Conference (COMPSAC)*, 805-810.