A New Design of Oblivious Transfer for Private Information Retrieval

Hui-Feng Huang
Graduate School of Computer Science and Information Technology
National Taichung Institute of Technology Taichung 404, Taiwan, R.O.C.

Summary
In a t-out-n oblivious transfer, the receiver can only receive t messages out of n messages sent by the sender; and the sender has no idea about which ones have been received. Majority of the existing of previous efficient oblivious transfer schemes require t calls of 1-out-n oblivious transfer to construct the t-out-n oblivious transfer. Its computational requirements and bandwidth consumption are quite demanding. Therefore, to guarantee the quality of electronic commercial service, an efficient t-out-n oblivious transfer scheme is urgently desired. Based on the RSA, we propose a new t-out-n oblivious transfer protocol for private information retrieval in this paper. In our method, only several modular multiplications and t hash functions are performed by a user (receiver) to obtain t messages. It is very suitable for the limited computation capacities of receivers such as smart cards or mobile units.

Key words:
Private information retrieval, security, RSA

1. Introduction

Rabin proposed the two-party oblivious transfer (OT) scheme in the cryptographic scenario [1]. For 1-out-*n* OT, Alice (sender) has *n* secrets $M_1, M_2, \ldots, M_n$ and would like to reveal one of them to a receiver (Bob) at the receiver's choice. Again, Bob does not want Alice to know which secret he chooses. Also Bob cannot obtain other $n-1$ secrets. The oblivious transfer has found many applications in cryptographic studies, such as fair electronic contract signing, oblivious secure computation, private information retrieval (PIR), etc [3][4] [5]. The *t*-out-*n* OT is a natural extension of 1-out-n OT, in which the sender has *n* secrets $M_1, M_2, \ldots, M_n$, and is willing to reveal *t* of them to a receiver at the receiver's choice. Also the receiver cannot obtain other $n-t$ secrets for each protocol.

The *t*-out-*n* oblivious transfer has many applications. One application is for private information retrieval (PIR), in which the user wants to query some data blocks (*t* blocks) from a database, but the user does not want the database manager (DBM) to know which data blocks (*t* blocks) he is interested in [3]. The regular PIR does not restrict the user to obtain only one data block of the database. So far, most of efficient previous 1-out-*n* OT schemes cannot easily construct a *t*-out-*n* OT scheme [2][6][7][9][11]. These previous constructions require *t* calls of 1-out-*n* OT to form *t*-out-*n* OT. Their computation complexity is $O(nt)$ modular exponentiations for Alice (sender) and $O(t)$ modular exponentiations for Bob (receiver). In Tzeng's scheme [9][11], the sender needs $2nt$ modular exponentiations and the receiver needs $2t$ modular exponentiations. The computational requirements and bandwidth consumption rates are quite demanding and likely to bottleneck in many applications. Recent technological advances have allowed users to carry portable communication devices. They offer affordable mobile networking capabilities while using very little power. These above mentioned oblivious transfer schemes [2][6][7][9][11] require several modular exponentiations for a user or receiver to obtain secrets. Since these computations are time-consuming, these schemes are impractical for the situations where computation capacities of users are limited to smart-cards or mobile units. Recently, Mu et al. [12] proposed a novel non-interactive oblivious transfer scheme that can reduce many communication loads. By non-interactive we mean that Bob does not need to communicate with Alice during an oblivious transfer process. However, Mu et al.'s method is not a secure OT [13].

We consider the situation for the oblivious transfer protocol, the sender usually possesses more computation capacity than the client (receivers). In many computer environments, the client will be using a mobile station or a smart card. For example, with private information retrieval (PIR), the database manager (sender) possesses more computation capacity than the users (receivers). Mobile clients (receivers) and smart card users are greatly restricted in their computation capacity. To guarantee the quality of communication services for a growing mobile clientele, it is more urgent to construct low-computation for the receivers than for the senders. Clearly, the design of user (receiver) efficient *t*-out-*n* oblivious transfer will play an important role for many applications in the future. Based on the RSA cryptography [8], we propose an efficient *t*-out-*n* OT scheme in this paper. In our method, only several modular multiplications and *t* hash functions are required for the receiver to obtain *t* secrets; and the computational complexity for the sender is $n+t$ modular exponentiations. Moreover, we only need one call (two rounds) to construct *t*-out-*n* oblivious transfer. Compared

Manuscript received September 5, 2008.
Manuscript revised September 20, 2008.

with existing oblivious transfer schemes [2][6][7][9][11], our scheme can reduce many computations and communications for the sender and the receiver. With just some modular multiplications for a receiver to perform, it is especially suitable for mobile receivers and smart-card users.

The rest of this paper is organized as follows: In the next section, we propose a new efficient *t*-out-*n* oblivious transfer scheme. The security and performance of the scheme is examined in Section 3. Finally, a brief conclusion is given in Section 4.

2. The Proposed Scheme

Before we propose a new *t*-out-*n* oblivious transfer scheme, we first describe that a *t*-out-*n* OT should satisfy the following requirements [10]:

Correctness: If both the sender and the receiver perform the protocol, the receiver obtains *t* secrets after executing the protocol with the sender.

Receiving ambiguity: After executing the protocol with the receiver, the sender will not know which *t* secrets the receiver has received.

Sending privacy: After executing the protocol with the sender for every time, the receiver only gets t messages for each time.

Let the sender possess a string of *n* secrets $M_1, M_2, ..., M_n$ and be willing to reveal *t* secrets of them to the receiver at the receiver's choice for every time. Based on the RSA [8], the sender randomly chooses two large primes *p* and *q*, then computes $N = pq$ and $\phi(N) = (p-1)(q-1)$. The sender also calculates a private key *d* so that $ed = 1 \mod \phi(N)$. Next, he randomly picks up *n* parameters $y_1, y_2, ..., y_n$. Then, the sender publishes *N*, *e*, $y_1, y_2, ..., y_n$ and keeps (*d*, *p*, *q*) secretly. The detail of the protocol is depicted as follows.

Step1. At first, the receiver chooses *t* random secrets $s_1, s_2, ..., s_t \in Z_N^*$ and *t* published parameters $y_{r1}, y_{r2}, ..., y_{rt} \in \{y_1, y_2, ..., y_n\}$. Then, he calculates

$c_1 = s_1^e \times h(y_{r1}, x), c_2 = s_2^e \times h(y_{r2}, x), ..., c_t = s_t^e \times h(y_{rt}, x) \mod N$ and sends

$c_1, c_2, ..., c_t$ to the sender. Here, the integer *x* is randomly selected by the system when the receiver calls the protocol.

Step 2. After receiving $c_1, c_2, ..., c_t$, the sender computes

$z_1 = h(y_1, x)^d \times M_1, z_2 = h(y_2, x)^d \times M_2, ..., z_n = h(y_n, x)^d \times M_n \mod N$ and

$w_1 = c_1^{-d}, w_2 = c_2^{-d}, ..., w_t = c_t^{-d} \mod N$. Then, the sender sends $\{w_1, w_2, ..., w_t\}$ and

$\{z_1, z_2, ..., z_n\}$ to the receiver.

Step 3. After receiving $\{w_1, w_2, ..., w_t\}$ and $\{z_1, z_2, ..., z_n\}$, the receiver obtains *t* messages by computing $M_{r1} = s_1 w_1 z_{r1}$, $M_{r2} = s_2 w_2 z_{r2}$, ..., $M_{rt} = s_t w_t z_{rt} \mod N$, where $z_{rj} \in \{z_1, z_2, ..., z_n\}$ for j =1,2,…,*t*, and $M_{r1}, M_{r2}, ..., M_{rt} \in \{M_1, M_2, ..., M_n\}$.

In Step 1, for the security and low-computation, given any valid message *m*, the system could generate smaller *e* so that the value m^e is greater than *N*. Then, it is suitable for mobile receivers and smart-card users (receiver). We could consider adding some important information such as date, time, or the identity number of the sender or the receiver before the message is encrypted by the sender. In addition, to provide the security of the proposed scheme, the random number *x* should not be re-sued. To achieve sending security, an authenticated channel from the sender to the receiver (or the receiver to the sender) is required. The authenticated channel can be achieved with authentication techniques and we omit it in the description. We give the following theorem to examine the correctness of the proposed method.

Theorem 1: In the proposed method, for each $j = 1,2,...,t$, the receiver can obtain the message $M_{rj} = s_j w_j z_{rj} \mod N$ and $M_{rj} \in \{M_1, M_2, ..., M_n\}$, where $z_{rj} = h(y_{rj}, x)^d \times M_j$ and $w_j = c_j^{-d} \bmod N$.

Proof: According to the proposed method, the receiver can calculate $c_j = h(y_{rj}, x) s_j^e \mod N$ for $j = 1,2,...,t$, where $y_{r1}, y_{r2}, ..., y_{rt} \in \{y_1, y_2, ..., y_n\}$ and *x* is randomly selected by the system when executing the protocol. On the other hand, the sender computes $z_i = h(y_i, x)^d \times M_i \mod N$ For $i = 1,2,...,n$ and $w_j = c_j^{-d} \mod N$ for $j = 1,2,...,t$. Hence, we have $w_j = c_j^{-d} = h(y_{rj}, x)^{-d} s_j^{-1} \mod N$ for $j = 1,2,...,t$. Since these s_j's are randomly chosen by the receiver for $j = 1,2,...,t$, the receiver can recover the validating message by computing

$M_{rj} = s_j w_j z_{rj} = s_j h(y_{rj}, x)^{-d} s_j^{-1} h(y_{rj}, x)^d M_{rj} = M_{rj} \mod N$ for $M_{rj} \in \{M_1, M_2, ..., M_n\}$ and $j = 1,2,...,t$.

3. Discussions

In this section, we examine the security properties and the performance of the proposed *t*-out-*n* oblivious transfer protocol.

3.1 Secrecy

In the proposed scheme, it provides the cipher $z_i = h(y_i, x)^d M_i \mod N$ for $i = 1,2,...,n$. Here, $y_1, y_2, ..., y_n$ are published parameters and the integer *x* is randomly given by the system when the receiver calls this system. If an intruder can easily get the integer *d*, then he can derive the message M_i. However, it is very difficult to obtain the sender's secret key *d* from the corresponding public key *e*. The security is based on the RSA cryptography [8]. Therefore, an attacker cannot decrypt secrets M_i from $z_i = h(y_i, x)^d M_i$. On the other hand, if the receiver *R* had obtained some secret message M_i, then he could compute

$$h(y_i, x)^d = \frac{z_i}{M_i} \mod N. \quad (1)$$

Now, assume another receiver R' would like to get *t* secrets from these *n* secrets $M_1', M_2', ..., M_n'$, according to our method, the sender computes these cipher $z_1' = h(y_1, x')^d \times M_1', z_2' = h(y_2, x')^d \times M_2', ..., z_n' = h(y_n, x')^d \times M_n' \mod N$ and sends them to the receiver R'. Thus, we have $M_i' = \frac{z_i'}{h(y_i, x')^d} \mod N$. Here, if the random number $x = x'$, then the receiver *R* could derive the receiver R''s message M_i' by the Equation (1). In this situation, our scheme is not secure. Hence, for the security of our scheme, the random number *x* should be used only one time.

3.2 Receiver's Ambiguity

In the proposed method, the receiver chooses *t* random secrets $s_1, s_2, ..., s_t \in Z_N^*$ and *t* published parameters $y_{r1}, y_{r2}, ..., y_{rt} \in \{y_1, y_2, ..., y_n\}$. Next, he computes $c_1 = s_1^e \times h(y_{r1}, x), c_2 = s_2^e \times h(y_{r2}, x), ..., c_t = s_t^e \times h(y_{rt}, x) \mod N$ and sends $c_1, c_2, ..., c_t$ to the sender. Assume $s_1 = s_2 = ,..., = s_t$, then we have

$$\frac{c_2}{c_1} = \frac{h(y_{r2}, x)}{h(y_{r1}, x)}, \quad \frac{c_3}{c_1} = \frac{h(y_{r3}, x)}{h(y_{r1}, x)}, ..., \quad \frac{c_t}{c_1} = \frac{h(y_{rt}, x)}{h(y_{r1}, x)} \mod N. \quad (2)$$

Under this condition, given *j*, the sender can compute $\frac{h(y_i, x)}{h(y_j, x)} \mod N$ for $i = 1,2,...,n$ and check with the Equation (2). Then, he could know which information the receiver wants to retrieve if he does an exhausted search within *n* times for $j = 1,2,...,n$. Thus, for the security of our scheme, these secrets s_j should be different for $j = 1,2,...,t$. Therefore, after receiving $c_1, c_2, ..., c_t$, the sender computes

$z_1 = h(y_1, x)^d \times M_1, \quad z_2 = h(y_2, x)^d \times M_2, ...,$ $z_n = h(y_n, x)^d \times M_n \mod N$ and

$w_1 = c_1^{-d}, w_2 = c_2^{-d}, ..., w_t = c_t^{-d} \mod N$; and

he sends $\{w_1, w_2, ..., w_t\}$ and

$\{z_1, z_2, ..., z_n\}$ to the receiver. However, without knowing *t* secrets s_j for $j = 1,2,...,t$, the sender cannot derive which message the receiver has received, where the integer s_j is randomly chosen by the receiver. Hence, after receiving $\{w_1, w_2, ..., w_t\}$ and $\{z_1, z_2, ..., z_n\}$, the receiver obtains *t* messages by computing $M_{r1} = s_1 w_1 z_{r1}, \quad M_{r2} = s_2 w_2 z_{r2}, ...,$ $M_{rt} = s_t w_t z_{rt} \mod N$ for every time, where $y_{r1}, y_{r2}, ..., y_{rt} \in \{y_1, y_2, ..., y_n\}$. Therefore, after executing our protocol with the receiver, the sender shall not know which *t* secrets the receiver has chosen each time. This means that our *t*-out-*n* oblivious transfer scheme satisfies the receiving ambiguity requirement.

3.3 Sender's Privacy

After executing our protocol with the sender, the receiver can get *t* secrets $M_{r1}, ..., M_{rt}$, where $r1, ..., rt \in \{1,2,\ldots,n\}$ are his choice for each time. In our method, it has the cipher $z_i = h(y_i, x)^d M_i \mod N$ for $i = 1,2,...,n$. However, based on the RSA cryptography, the receiver cannot easily derive *d* from his known message M_j, where $j \in \{1,2,...,n\}$. Without

knowing *d*, the receiver gains no information about other $_{M_i}, i \notin \{r1\ , r2, ..., rt\}$ for each time. This means that the proposed *t*-out-*n* oblivious transfer scheme provides the sending privacy property.

3.4 Performance

The complexity of most efficient previous *t*-out-*n* oblivious transfer schemes [2][6]7][9][11], which require *t* calls of 1-out-*n* oblivious transfer, is *O*(*nt*) modular exponentiations for the sender and *O*(*t*) modular exponentiations for the receiver. For example, in Tzeng's scheme [11], it requires two rounds for each call. Hence, it takes 2*t* rounds to construct *t*-out-*n* oblivious transfer in Tzeng's scheme. Again, the sender needs to send *nt* elements to the receiver and the receiver needs to send *t* elements to the sender. According to their computational complexity, the sender needs 2*nt* modular exponentiations and the receiver needs 2*t* modular exponentiations. However, in our scheme, we require just one call for constructing *t*-out-*n* oblivious transfer. Only two rounds are required for our method, in total the sender sends *n*+*t* elements to the receiver and the receiver sends *t* elements to the sender.

For convenience, we make comparisons between our scheme and Tzeng's scheme [11] in Table 1. The following notations are used to analyze the computational complexity. T_e means the time for one exponentiation computation. T_h stands the time for one hash function computation. Finally, T_m defines the time for one modular multiplication computation. Note that the time for computing modular addition and subtraction are ignored, since it is much smaller than T_e, T_h , and T_m . We summarize the comparisons of our *t*-out-*n* oblivious transfer scheme with Tzeng's scheme in Table 1. In our scheme, the computational complexity for the sender and the receiver are $(n+t)T_e + n(T_h + T_m)$ and $(3t + t|e|)T_m + tT_h$, respectively. Here, the integer $|e|$ represents the bit-length of the public key *e*. As shown in Table 1, our scheme is more efficient than Tzeng's scheme for both the sender and the receiver.

In addition, for the security and low-computation, given any valid message *m*, the system could generate smaller *e* such that the value m^e is greater than *N*. Hence, in our scheme, the receiver requires only several modular multiplications and *t* hash functions for the computations. Therefore, in comparison with previous oblivious transfer schemes [2][6][7][9][11], our scheme will reduce many computations and communications for both the sender and the receiver.

Table 1 Computational cost Comparisons of two schemes

	Tzeng's scheme	**Our scheme**
Computation cost for a sender	$2ntT_e + ntT_i + 2ntT_m$	$(n+t)T_e + n(T_h + T_m)$
Computation cost for a receiver	$2tT_e + tT_i + tT_m$	$(3t + t\|e\|)T_m + tT_h$
Communication load for the protocol	2*t* rounds	2 rounds

In addition, for the security and low-computation, given any valid message *m*, the system could generate smaller *e* such that the value m^e is greater than *N*. Hence, in our scheme, the receiver requires only several modular multiplications and *t* hash functions for the computations. Therefore, in comparison with previous oblivious transfer schemes [2][6][7][9][11], our scheme will reduce many computations and communications for both the sender and the receiver.

4. Conclusions

Based on the RSA cryptography, we have proposed a new *t*-out-*n* oblivious transfer scheme. The proposed protocol requires two rounds of communication. In our method, only several modular multiplications and *t* hash functions are performed for a receiver to obtain *t* secrets for each protocol. It is very suitable for mobile clients because no time-consuming computations are needed, such as modular exponentiation and inverse computations.

References

[1] M. Rabin, "How to Exchange Secrets by Oblivious Transfer", *Technical Report TR-81*, Aiken Computation Laboratory, Harvard University, 1981.

[2] G. Brassard and C. Crepeau, "Oblivious Transfer and Privacy Amplification", *Proceedings Advances in Cryptology* (*Eurocrypt'97*), pp. 334-346, 1997.

[3] B. Chor, O. Goldreich, E. Kushilevitz, and M. Susdan, "Private Information Retrieval", *Journal of the ACM*, vol. 45, no. 6, pp. 965-982, 1998.

[4] G. Di Crescenzo, T. Malkin, and R. Ostrovsky, "Single Database Private Information Retrieval Implies Oblivious Transfer", *Proceedings Advances in Cryptology (Eurocrypt'00)*, pp. 122-138, 2000.

[5] S. Even, O. Goldreich, and A. Lempel, "A Randomized Protocol for Signing Contracts", *Communications of ACM*, vol. 28, pp. 637-647, 1985.

[6] M. Naor and B. Pinkas, "Efficient Oblivious Transfer Protocols", *Proceedings 12th Ann. Symp. Discrete Algorithms*, pp. 448-457, 2001.

[7] J. P. Stern, "A New and Efficient All-or-Nothing Disclosure of Secrets Protocol", *Proceedings Advances in Cryptology* (*Asiacrypt'98*), pp. 357-371, 1998.

[8] R.L. Rivest, A. Shamir and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, *Communications of ACM*, vol. 21, no. 2, pp. 120-126, February 1978.
[9] W. Tzeng, “Efficient 1-out-of-*n* Oblivious Transfer Schemes”, *PKC’02*, pp. 159-171, 2002.
[10] Q. H. Wu, J. H. Zhang, and Y. M. Wang, “Practical t-out-n Oblivious Transfer and Its Applications”, *Information and Communications Security, ICICS 2003*, LNCS 2836, pp. 226-237, 2003.
[11] W. G. Tzeng, “Efficient 1-out-n Oblivious Transfer Schemes with Universally Usable Parameters”, *IEEE Transactions on Computers*, vol. 53, no. 2, pp. 232-240, 2004.
[12] Y. Mu, J. Zhang, V. Varadharajan, and Y. X. Lin, “Robust Non-Interactive Oblivious Transfer,” *IEEE Communications Letters*, vol. 7, no. 4, pp.153-155, 2003.
[13] H. F. Huang, C. C. Chang, and J. S. Yeh, "Enhancement of Non-Interactive Oblivious Transfer Scheme," *Fourth International Conference on Information and Management Sciences (IMS)*, Kunming & Dali & Shangrila, China, pp. 196-199, Jul. 2005.

Hui-Feng Huang received her Ph.D. degree in Computer Science and Information Engineering from National Chung Cheng University. Her first degree is Bachelor of Mathematics from Fu Jen Catholic University and Master of Mathematics from National Taiwan University. Currently, she is currently an associate professor at the Department of Information Management in National Taichung Institute of Technology. Her research interests focus on the areas of cryptography and information security, network security, algorithm, and electronic commerce etc.