A New Efficient k-out-of-n Oblivious Transfer Scheme by means of Common Cipher

Chih-Hung Wang and Chi-Shin Lin
Department of Computer Science and Information Engineering
National Chiayi University, Taiwan 600
wangch@mail.ncyu.edu.tw

Abstract*-The oblivious transfer protocol has a critical problem on the sender's communication complexity. For this reason, we present a significant component called common cipher, which conceals all sender's secrets. The sender can derive the corresponding decryption keys from a common cipher and different encryption keys. The advantage of the technology of common cipher is to greatly reduce the sender's communication cost. Therefore, we employ our common cipher to develop an efficient* OT_k^n*. Our result is superior to all previous solutions in regard to the sender's communication complexity. In our* OT_k^n *protocol, the sender cannot determine which k secret messages the chooser received even if the sender has unlimited computational power, and the chooser cannot get the other* $n-k$ *secret messages if the factorization problem is hard. When* $k=1$ *, we particularly suggest an efficient solution.*

Keywords: Cryptography, Oblivious Transfer, RSA, Network Security.

1. Introduction

The Oblivious Transfer protocol (OT for short) was first introduced by Rabin [19] in 1981. Rabin's OT is a two-party protocol in which the sender delivers a one bit secret *b* to the chooser. In this case, the chooser has $\frac{1}{2}$ probability to get this secret, and on the other hand, the sender does not learn whether the chooser receives it. Until now, OT schemes for various types have been proposed (e.g., [2,5,6,8,13,14,15,18,22,24]).

The chooser wants to get anticipated secret information instead of according to probability, so the 1-out-of-2 OT $\left(OT_1^2\right)$ is developed. The sender transmits two secrets (m_0, m_1) to the chooser. Assume that the chooser wants to acquire the secret m_α. After oblivious transfer, the sender cannot know α, and the chooser cannot get $m_{1-\alpha}$. In order to provide a wide range of applications for electronic commerce, many papers extended the original protocol to 1-out-of-*n* OT $\left(OT_1^n\right)$. OT_1^n is a natural extension of OT_1^2, that is to say the sender have *n* secrets and the chooser can only select one. Similarly, *k* -out-of- *n* $\left(OT_k^n\right)$ is very easy to understand. The chooser can select *k* secret messages under the situation of $1 \le k < n$.

OT is an important cryptography primitive. It can play a key role to design a lot of cryptographic protocol, including electronic trading [1,10], private information retrieval [4] (PIR), digital signature [23], oblivious search [17] and so forth.

In the recent literature, the authors have proposed many efficient OT_1^n [6,16,20,21,22] and OT_k^n [13,18,24] protocols. Security assumption of these schemes is mainly based on discrete logarithm problem, computational Diffie-Hellman problem [9] and RSA problem [9]. No matter whether these OT protocols have good improvement in computation cost, they still need to take $O(n)$ complexity in the sender's communication. For this reason, we present a significant component called common cipher. In other words, we can hide a lot of secret messages in one cipher. This concept is extended from Maurer and Yacobi's identity-based non-interactive public-key distribution scheme [11]. The sender can derive the corresponding decryption keys from the common cipher and different encryption keys.

In our OT_k^n protocol, the sender cannot determine which *k* secret messages the chooser received even if the sender has unlimited computational power, and the chooser cannot get the other $n-k$ secret messages if the factorization problem is hard. When $k=1$, we particularly suggest an efficient solution.

Road-Map. The remainder of this paper is organized as follows. In Section 2 we present some definitions and the security requirements for our OT schemes. In Section 3, we introduce our common cipher technique and discuss its practicality. We describe our efficient OT_k^n protocol in Section 4. In Section 5, we compare the overheads with other OT schemes. The concluding remarks are given in Section 6.

2. Preliminaries

2.1. k-out-of-n OT

OT_k^n is a transmission protocol for two parties, the sender and the chooser. The sender have secret messages $m_1, m_2, \ldots, m_n$. The chooser can choose k ones from these n secret messages. OT_k^n must satisfy the following requirements.

Completeness: If the chooser and the sender honestly follow the procedure of the protocol, the chooser can properly get k secret messages.

The chooser's privacy: The chooser's privacy is unconditionally secure. In other words the sender does not know which secret messages are chosen by the chooser.

The sender's privacy: Even if chooser is dishonest, he is still unable to learn other remaining $n-k$ secret messages or other meaningful combinations.

2.2. The RSA Problem

Suppose that $N = pq$ (p and q are all prime numbers), e is an integer and satisfies $\gcd(e,(p-1)(q-1))=1$, and $C \in Z_N^*$.

The RSA problem is to find the unique integer $m \in Z_N^*$ such that $m^e \equiv C \pmod N$. Under RSA assumption, someone is infeasible to solve RSA problem if the factorization of the composite number is difficult.

3. Common Cipher

The main idea of our common cipher is coming from Maurer and Yacobi's [11] identity-based non-interactive public-key distribution scheme. Their scheme [11], based on a novel trapdoor one-way function, allows trusted authority to calculate discrete logarithm under the situation of a given number modulo a publicly known composite number M while that is infeasible for an adversary not knowing the factorization of M. We are interested in such a novel trapdoor one-way function. Further, Lim and Lee [7] proposed some suggestions and modifications for Maurer-Yacobi's scheme to be fit to develop a lot of applications.

What is common cipher? It means to conceal a lot of secret messages in one cipher. The sender can derive the corresponding deciphering keys from this cipher and different encryption keys.

Only the sender knows decryption keys of these secret messages, so it is safe to publish the common cipher. Now we carefully describe the construction steps of common cipher in the following.

3.1. Building Common Cipher

Let $M = p_1 . p_2 . p_3$ be a factorization of M into three prime numbers, and these prime numbers are all odd numbers and different. The Euler phi-function $\phi(M) = (p_1-1)(p_2-1)(p_3-1)$. For $i = 1,2,\ldots,n$, e_i and d_i must satisfy $1 < e_i, d_i < \phi(M)$ and $\gcd(e_i, \phi(M)) = 1$. We assume that $m_1, m_2, \ldots, m_n$ are secret messages and the secret space is Z_M^*.

For $i = 1,2,\ldots,n$ and $j = 1,2,3$, common cipher C is built according to the following procedure:

(1) Compute $C, x_{i1}, x_{i2}, x_{i3}$ satisfying the equation $C \equiv m_i^{x_{ij}} \pmod{p_j}$ for $j = 1,2,3$.

(2) Apply the Chinese Remainder Theorem [9] to compute e_i from x_{i1}, x_{i2} and x_{i3} , such that $C \equiv m_i^{e_i} \pmod M$.

(3) Compute an unique value d_i such that $e_i d_i \equiv 1 \pmod{\phi(M)}$.

Remark 3.1 The reader can refer to [11] for the details of numbers of primes, range of each prime and so on. Maurer and Yacobi in [11] suggest choosing 3 to 4 prime factors of between 60 and 70 decimal digits. As selecting each prime on a proper range, the sender is feasible in calculating discrete logarithms x_{i1}, x_{i2}, x_{i3} modulo p_j to the base m_i. On the contrary, without knowing the factorization of M, an adversary is difficult to calculate the discrete logarithms x_{i1}, x_{i2}, x_{i3}.

4. Efficient OT_k^n with Common Cipher

Malkhi and Sella investigate the relationship between OT and blind signatures, and propose a very efficient OT_1^n protocol called blind OT [8] which relies on Chaum's RSA based blind signature scheme [3]. We find that if we combine blind OT with our common cipher, that is a major breakthrough in efficiency of OT. Therefore we propose an efficient OT_k^n in light of blind OT's essence. In our scheme, the chooser sends $O(\lceil k/2 \rceil)$ messages to the sender. The sender sends $O(\lceil k/2 \rceil)+1$ messages back to the chooser. As to the computation cost, the chooser performs $O(2\lceil k \rceil)$ exponentiations, $O(k)$ exclusive-or operations and $O(3\lceil k/2 \rceil)$ multiplications, and the sender performs $O(\lceil k/2 \rceil)$ exponentiations. Compared with other protocols, our

scheme attains the beneficial results in communication and computation complexity.

4.1. Proposed Protocol

Our scheme consists of an initialization phase and a transfer phase. Initialization phase is a pre-computation phase done by the sender, and the execution of OT protocol is in transfer phase. In order to guarantee the security and prevent the common module attack [12], the parameters of initialization phase must be renewed and published again after a transaction of OT. Hence we suppose that the sender can precompute a lot of parameters of initialization phase. It is a minor shortcoming that the public parameters cannot be reused. But the proposed protocol really can reduce the communication cost in transfer phase which will generally become a communication bottleneck. We remove this limitation in $k=1$ protocol.

We formally describe these phases below.

Initialization Phase:

The sender prepares p_1, p_2, p_3, $M = p_1 \cdot p_2 \cdot p_3$, the Euler phi-function $\phi(M)$ and n secret messages $m_1, m_2, \ldots, m_n$. For $i = 1,2,\ldots,n$. Initialization Phase goes on with the following steps:

(1) The sender uses the technology of common cipher to deal with m_i, and computes C, (e_i, d_i), chooses a random number $r \in Z^*_{\phi(M)}$.
Then she publishes C, M, $\varepsilon_i = e_i \oplus r$.

(2) The sender chooses a random number $b \in Z^*_{\phi(M)}$, and computes

$$o_{pri} = (b)^{-1} \cdot \left(\prod_{i=1}^{n} d_i \right) \bmod \phi(M),$$

$$e_{pri} = (b^2)^{-1} \left(\prod_{i=1}^{n} d_i \right)^2 \bmod \phi(M).$$

Then the sender computes and publishes the following values:

$$\alpha_1 = b. \prod_{i \in [1,n], i \neq 1} e_i \bmod \phi(M),$$

$$\alpha_2 = b. \prod_{i \in [1,n], i \neq 2} e_i \bmod \phi(M),$$

…,

$$\alpha_n = b. \prod_{i \in [1,n], i \neq n} e_i \bmod \phi(M).$$

Transfer Phase:

Suppose that the chooser selects k indexes $t_1, t_2, \ldots, t_k \in \{1, \ldots, n\}$.

(1) Chooser first picks $B \in Z^*_M$, then he needs to consider the following two situations:

Case 1 (k is even)

(a) Computes $B_{even} = B^{\left(b \cdot \prod_{i=1}^{n} e_i\right)^2} = B^{(e_1 \alpha_1)^2}$ and $\beta_i = \alpha_{t_{2i-1}} \alpha_{t_{2i}}$ for $i = 1,2,\ldots,\lfloor k/2 \rfloor$.

(b) Computes $X_i = C^{\beta_i} B_{even}$ for $i = 1,2,\ldots,\lfloor k/2 \rfloor$ and sends them to the sender.

Case 2 (k is odd)

(a) Computes $B_{odd} = B^{\left(b \cdot \prod_{i=1}^{n} e_i\right)} = B^{(e_1 \alpha_1)}$ and $\beta_i = \alpha_{t_{2i-1}} \alpha_{t_{2i}}$ for $i = 1,2,\ldots,\lfloor k/2 \rfloor$.

(b) Computes $X_i = C^{\beta_i} B_{even}$ for $i = 1,2,\ldots,\lfloor k/2 \rfloor$ and $Y_{\lceil k/2 \rceil} = C^{\alpha_k} B_{odd}$, and sends them to the sender.

(2) Sender computes $Z_i = (X_i)^{e_{pri}} \bmod M$ for $i = 1,2,\ldots,\lfloor k/2 \rfloor$.
If k is odd, then he must additionally compute $W_{\lceil k/2 \rceil} = \left(Y_{\lceil k/2 \rceil}\right)^{o_{pri}} \bmod M$. Finally, the sender sends r, Z_i for $i = 1,2,\ldots,\lfloor k/2 \rfloor$ and $W_{\lceil k/2 \rceil}$ to the chooser.

(3) Chooser decrypts the cipher according to the following equations:

$$e_{t_i} = \varepsilon_{t_i} \oplus r, \text{ for } i = 1,2,\cdots k.$$

$$m_{t_{2i-1}} = \left(Z_i \cdot B^{-1}\right)^{e_{t2i}} (\bmod M),$$

$$m_{t_{2i}} = \left(Z_i \cdot B^{-1}\right)^{e_{t2i-1}} (\bmod M), \text{ for } i = 1,2,\ldots,\lfloor k/2 \rfloor.$$

If k is odd, the chooser must additional decrypt $m_{t_k} = \left(W_{\lceil k/2 \rceil} \cdot B^{-1}\right) \bmod M$.

4.2. Security Analysis

Claim 1: *(Completeness) If the sender and the chooser follows the process of the protocol correctly, the chooser will receive exactly k of n secret messages retrieved from the common cipher.*

Utilizing the public values announced in the initialization phase, the chooser can hide k secret indexes $t_1, t_2, \ldots, t_k$ into $X_i = C^{\beta_i} B^{\left(b \cdot \prod_{i=1}^{n} e_i\right)^2} \bmod M$ for $i = 1,2,\ldots,\lfloor k/2 \rfloor$ ($Y_{\lceil k/2 \rceil} = C^{\alpha_k} B^{\left(b \cdot \prod_{i=1}^{n} e_i\right)} \bmod M$, if k is odd). There are two blind factors b and B are embedded into X_1, X_2, …, $X_{\lfloor k/2 \rfloor}$ (, $Y_{\lceil k/2 \rceil}$). Because only the sender knows e_{pri} and o_{pri}, therefore she can unblind b embedded inside these ciphers. Thus the result of unblinding becomes

$$Z_i = (X_i)^{e_{pri}} = \left(C^{\alpha_{t_{2i-1}} \alpha_{t_{2i}}} B^{\left(b \cdot \prod_{i=1}^{n} e_i\right)^2} \right)^{(b^2)^{-1} \left(\prod_{i=1}^{n} d_i \right)^2} \bmod M$$

for $i = 1,2,\ldots,\lfloor k/2 \rfloor$

($W_{\lceil k/2 \rceil} = \left(Y_{\lceil k/2 \rceil}\right)^{o_{pri}} = \left(C^{\alpha_{tk}} B^{\left(b.\prod_{i=1}^{n} e_i\right)}\right)^{(b)^{-1}.\left(\prod_{i=1}^{n} d_i\right)} \bmod M$). Obviously, the chooser can easily unblind $Z_1, Z_2, \ldots, Z_{\lfloor k/2 \rfloor}$ and $W_{\lceil k/2 \rceil}$, because he knows blind factor B and can calculate $B^{-1} \bmod M$. The chooser can decipher the secret messages $m_{t2i-1} = \left(Z_i . B^{-1}\right)^{e_{t2i}} = \left(\left(C^{d_{t2i-1} d_{t2i}} B\right) . B^{-1}\right)^{e_{t2i}}$, $m_{t2i} = \left(Z_i . B^{-1}\right)^{e_{t2i-1}} = \left(\left(C^{d_{t2i-1} d_{t2i}} B\right) . B^{-1}\right)^{e_{t2i-1}}$ for $i = 1,2,\ldots,\lfloor k/2 \rfloor$ ($m_{t_k} = \left(W_{\lceil k/2 \rceil} . B^{-1}\right) = \left(\left(C^{d_{tk}} B\right) . B^{-1}\right) (\bmod M)$, if k is odd).

Claim 2: *(Chooser's privacy) the sender cannot learn which k messages are choosen by the chooser.*

Because the sender cannot know the secret blind factor B selected by the chooser, so our scheme satisfies the chooser's privacy. Regard to detailed proof, readers can understand the property of information-theoretic blindness from Chaum's blind signature scheme [3].

Claim 3: *(Sender's privacy) the chooser cannot get more than k secret messages.*

As illustrated in Claim 1. The chooser utilizes blind factor B, so the sender does not know which secret messages the chooser wants to obtain. In our scheme, b is an important factor that can prevent the chooser from constructing the errorous X_i and $Y_{\lfloor k/2 \rfloor}$ to maliciously get more messages. By using $e_{pri} = \left(b^2\right)^{-1} \left(\prod_{i=1}^{n} d_i\right)^2$ and $o_{pri} = (b)^{-1} . \left(\prod_{i=1}^{n} d_i\right)$, the sender can guarantee that the chooser only can obtain two secret messages form Z_i, for $i = 1,2,\ldots,\lfloor k/2 \rfloor$ and one secret message from $W_{\lceil k/2 \rceil}$. In other words, the sender can guarantee the chooser finally can only get k secret messages at most.

4.3. When $k = 1$

Under the condition of $k = 1$, in order to reduce the amount of information published by the sender, we modify the protocol of Section 4.1 as follows.

Initialization Phase:

(1) This step is the same as one in Section 4.1 except that the sender does not disclose ε_i for $i = 1,2,\ldots,n$.

(2) The sender chooses a random number $b \in Z^*_{\phi(M)}$, and computes $e_{pri} = (b)^{-1} . \left(\prod_{i=1}^{n} d_i\right) \bmod \phi(M)$.

Then she computes and publishes following numbers:

$e_{pub} = \left(b . \prod_{i=1}^{n} e_i\right) \bmod \phi(M)$,

$\alpha_1 = b . \prod_{i \in [1,n], i \neq 1} e_i \bmod \phi(M)$,

$\alpha_2 = b . \prod_{i \in [1,n], i \neq 2} e_i \bmod \phi(M)$,

…,

$\alpha_n = b . \prod_{i \in [1,n], i \neq n} e_i \bmod \phi(M)$.

Transfer Phase:

Suppose the chooser selects a secret index $s \in \{1,\ldots,n\}$.

(1) The chooser randomly picks $B \in Z^*_M$, and computes $X = C^{\alpha_s} B^{e_{pub}} \bmod M$, then sends X to the sender.

(2) Sender computes $Y = X^{e_{pri}} \bmod \phi(M)$ and sends Y to the chooser.

(3) Chooser can decrypt the cipher by calculating $m_s = \left(Y . B^{-1}\right) \bmod M$.

5. Overhead Comparison

Transfer phase is the most important procedure in the OT protocol. Hence about overhead comparison, we focus on the transfer phase. All OT schemes that we compare with have good efficiency in communication and computation complexity. In addition, we omit to evaluate the precomputation costs in all OT schemes.

In our OT_k^n scheme, as employing common cipher, each of $X_1, X_2, \ldots, X_{\lfloor k/2 \rfloor}$ and $Z_1, Z_2, \ldots, Z_{\lfloor k/2 \rfloor}$ ciphers in our scheme consists of double secret messages. Consequently, the communication cost of the sender is $O(\lceil k/2 \rceil) + 1$, and the chooser is only $O(\lceil k/2 \rceil)$. We compare our OT_k^n scheme with the schemes of Ogata et al. [18], Wu et al. [24] (based on discrete logarithm) and Mu et al. [13] (efficient interactive OT_m^n scheme). The result is shown in Table 1 and Table 2.

In our OT_1^n (see Section 4.3) scheme, the chooser sends 1 message to the sender. Similarly the sender replies 1 message to the chooser. As to the computation cost, the chooser performs 1 exponentiation and 2 multiplications, and the sender performs 1 exponentiation. We compare our OT_1^n scheme with the schemes of Kurosawa et al. [6] (RSA-based scheme), Wu et al. [24] (for $k = 1$ case) and Tzeng [22] (based on random oracle model). Table 4 and Table 5 show the overhead comparisons.

Besides, Table 3 and Table 6 show the amount of published information within these OT schemes. Although our protocol needs to publish more public

values, nevertheless, our proposed scheme has a significant improvement on communication and computation complexity specially when k is a small value.

6. Concluding Remarks

We have proposed a useful technique called common cipher. In our scheme, though the sender will be time-consuming in building common cipher, the communication cost of the sender can be greatly reduced. In our proposed OT_k^n, some attack methods will be considered [12]. In order to guarantee the security of the protocol, the parameters of initial phase must be renewed and published again in each transaction. However, our proposed OT_1^n has no this problem. Our improvement really benefits the sender when the sender is required to deal with a lot of transactions at the same time. Moreover, our common cipher is specially suitable for the development of private information retrieval (PIR). In the environment of PIR, it is not necessary to restrict the chooser to only obtain k out of n messages.

Furthermore, we will work in the future to accelerate the construction of the common cipher and develop a more efficient OT_k^n.

Table 1. Communication complexity in OT_k^n

	Sender	Chooser
The proposed scheme	$\lceil k/2 \rceil$ group elements and 1 string	$\lceil k/2 \rceil$ group elements
Ogata et al. [18]	n strings n group elements	$(2k+1)$ group elements
Wu et al. [24]	k group elements	k group elements
Mu et al. [13]	n strings	$2n$ group elements

Table 2. Computation complexity in OT_k^n

	Sender	Chooser
The proposed scheme	$\lceil k/2 \rceil$ exponentiations	$2\lceil k \rceil$ exponentiations $\frac{3\lceil k \rceil}{2}$ multiplications k exclusive-or operations
Ogata et al. [18]	$4n$ exponentiations	$(3k+1)$ exponentiations
Wu et al. [24]	k exponentiations	$2k$ exponentiations
Mu et al. [13]	$2n$ exponentiations n multiplications	k exponentiations $2k$ multiplications k additions

Table 3. Amount of the public values in OT_k^n

	Number of public values
The proposed scheme	$2n+2$
Ogata et al. [18]	1
Wu et al. [24]	$n+1$
Mu et al. [13]	n

Table 4. Communication complexity in OT_1^n

	Sender	Chooser
The proposed scheme	1 group element	1 group element
Kurosawa et al. [6]	n strings + $\lceil \log_2 N \rceil$*	1 group element
Wu et al. [24]	1 group element	1 group element
Tzeng [22]	n strings 1 group element	1 group element

* $N = p.q$ with prime numbers p and q.

Table 5. Computation complexity in OT_1^n

	Sender	Chooser
The proposed scheme	1 exponentiation	1 exponentiation 2 multiplications
Kurosawa et al. [6]	1 exponentiation n divisions	1 exponentiation 1 multiplication
Wu et al. [24]	1 exponentiation	2 exponentiations
Tzeng [22]	3 exponentiations	2 exponentiations

Table 6. Amount of public values in OT_1^n

	Number of public values
The proposed scheme	$n+3$
Kurosawa et al. [6]	3
Wu et al. [24]	$n+1$
Tzeng [22]	3

References

[1] B. Aiello, Y. Ishai, and O. Reingold, "Priced Oblivious Transfer: How to Sell Digital Goods," Advances in Cryptology - Eurocrypt 2001, LNCS 2045, pp.119-135, 2001.

[2] M. Bellare and S. Micali, "Non-Interactive Oblivious Transfer and Applications," Advances in Cryptology - Crypto '89, LNCS 435, pp.547-557, 1990.

[3] D. Chaum, “Blind Signatures for Untraceable Payments,” Advances in Cryptology - Crypto ’82, pp.199-203, 1982.

[4] G.D. Crescenzo, T. Malkin, and R. Ostrovsky, “Single Database Private Information Retrieval Implies Oblivious Transfer,” Advances in Cryptology - Eurocrypt 2000, LNCS 1807, pp.122-138, 2000.

[5] J.A. Garay and P. Mackenzie, “Concurrent Oblivious Transfer,” Proceedings of the 41st IEEE Symposium on Foundations of Computer Science, pp.314-324, 2000.

[6] K. Kurosawa and Q.V. Duong, “How to Design Efficient Multiple-Use 1-out-n Oblivious Transfer,” IEICE Trans. Fundamentals, vol. E87-A, no. 1, pp.141-146, 2004.

[7] C.H. Lim and P.J. Lee, “Modified Maurer-Yacobi’s scheme and its applications,” Advances in Cryptology - Asiacrypt ’92, LNCS 718, pp.308-323, 1992.

[8] D. Malkhi and Y. Sella, “Oblivious Transfer Based on Blind Signatures,” Technical report, Leibniz Center For Research in Computer Science: Report 2003/31, 2003.

[9] W. Mao, *Modern Cryptography: Theory and Practice*, Prentice Hall, New Jersey, 2003.

[10] S. Matsuo and W. Ogata, “Matching Oblivious Transfer: How to Exchange Valuable Data,” IEICE Trans. Fundamentals, vol. E86-A, no. 1, pp.189-193, 2003.

[11] U.M. Maurer and Y. Yacobi, “Non-interactive Public-Key Cryptography,” Advances in Cryptology - Eurocrypt ’91, LNCS 547, pp.498-507, 1991.

[12] J.H. Moore, “Protocol failures in cryptosystems,” Proceedings of the IEEE, vol. 76, issue: 5, pp.594-602, 1988.

[13] Y. Mu, J. Zhang, and V. Varadharajan, “m out of n Oblivious Transfer,” Australasian Conference on Information Security and Privacy (ACISP) 2002, LNCS 2384, pp.395-405, 2002.

[14] 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.

[15] M. Naor and B. Pinkas, “Distributed Oblivious Transfer,” Advances in Cryptology - Asiacrypt 2000, LNCS 1976, pp.205-219, 2000.

[16] M. Naor and B. Pinkas, “Efficient Oblivious Transfer Protocols,” Proceedings of the 12th Annual Symposium on Discrete Algorithms (SODA), pp.448-457, 2001.

[17] W. Ogata and K. Kurosawa, “Oblivious Keyword Search,” Technical report, Cryptology ePrint Archive: Report 2002/182, 2002.

[18] W. Ogata and R. Sasahara, “k out of n Oblivious Transfer without Random Oracles,” IEICE Trans. Fundamentals, vol. E87-A, no. 1, pp.147-151, 2004.

[19] M. Rabin, “How to Exchange Secrets by Oblivious Transfer,” Technical Report TR-81, Aiken Computation Lab., Harvard University, 1981.

[20] C. Tobias, “Practical Oblivious Transfer Protocols,” Information Hiding (IH) 2002, LNCS 2578, pp.415-426, 2002.

[21] W.G. Tzeng, “Efficient 1-out-n oblivious transfer schemes,” Public Key Cryptography (PKC) 2002, LNCS 2774, pp.159-171, 2002.

[22] W.G. Tzeng, “Efficient 1-out-of-n Oblivious Transfer Schemes with Universally Usable Parameters,” IEEE Transactions on Computers, vol. 53, no. 2, pp.232-240, 2004.

[23] H. Wang and J. Pieprzyk, “Efficient One-Time Proxy Signatures,” Advances in Cryptology - Asiacrypt 2003, LNCS 2894, pp.507-522, 2003.

[24] Q.H. Wu, J.H. Zhang, and Y.M. Wang, “Practical t-out-n Oblivious Transfer and Its Applications,” International Conference on Information and Communications Security (ICICS) 2003, LNCS 2836, pp.226-237, 2003.