Efficient Oblivious Transfer Schemes*

Wen-Guey Tzeng
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan 30050
Email:tzeng@cis.nctu.edu.tw

Abstract

In this paper we propose a very efficient (string) OT_n^1 scheme for any $n \geq 2$. We build our OT_n^1 scheme from fundamental cryptographic techniques directly. It achieves optimal efficiency in the number of rounds and the total number of exchanged messages for the case that the receiver's choice is unconditionally secure. The computation time of our OT_n^1 scheme is very efficient, too. The receiver need compute 2 modular exponentiations only no matter how large n is, and the sender need compute $2n$ modular exponentiations. Furthermore, the system-wide parameters need not change during the lifetime of the system and are *universally usable*. That is, all possible receivers and senders use the same parameters and need no trapdoors specific to each of them. For our OT_n^1 scheme, the privacy of the receiver's choice is unconditionally secure and the privacy of the un-chosen secrets is at least as strong as the hardness of the decisional Diffie-Hellman problem.

We extend our OT_n^1 scheme to distributed oblivious transfer schemes. Our distributed OT_n^1 scheme takes full advantage of the research results of secret sharing and is conceptually simple. It achieves better security than Noar and Pinkas's scheme does in many aspects. For example, our scheme is secure against collusion of R and t-1 servers and it need not restrict R to contact at most t servers, which is difficult to enforce.

For applications, we present a method of transforming any single-database PIR protocol into a symmetric PIR protocol with only one extra unit of communication cost.

Keywords: oblivious transfer, distributed oblivious transfer, private information retrieval.

1 Introduction

Consider two parties of the sender S and the receiver R. S has n secrets $m_1, m_2, \ldots, m_n$ and is willing to disclose one of them (m_α) to R at R's choice α. However, R does not want to reveal to S its choice α and S does not want R to gain any information about other secrets m_i, $i \neq \alpha$. An oblivious transfer scheme is a cryptographic two-party protocol that provides a solution for the

*Research supported in part by National Science Council grant 89-2213-009-145 and MOE Excellence grant 89-E-FA04-1-4, Taiwan, ROC.

goal. Oblivious transfer has many flavors, such as, original oblivious transfer (OT), 1-out-2 oblivious transfer (OT_2^1) and 1-out-n oblivious transfer (OT_n^1).

Rabin [31] proposes the concept of oblivious transfer (OT) in the cryptographic scenario. In this case, S has only one secret (bit) m and would like to have R to get it with probability 0.5. On the other hand, R does not want S to know whether it gets m or not. For OT_2^1, S has two secrets m_1 and m_2 and would like to have R to get one of them at R's choice. Again, R does not want S to know which secret it chooses. OT_n^1 is a natural extension of OT_2^1 to the case of n secrets. Nevertheless, to construct OT_n^1 from OT_2^1 is not trivial. OT_n^1 is also known as "all-or-nothing disclosure of secrets (ANDOS)" in which R is not allowed to gain combined information of the secrets, such as, their exclusive-or. Essentially, all these flavors are equivalent in the information theoretic sense [9, 12, 15]. Oblivious transfer is a fundamental primitive for cryptography and secure distributed computation [23, 24] and has many applications, such as, private information retrieval (PIR), fair electronic contract signing, oblivious secure computation, etc [5, 14, 21].

A general approach for constructing OT_n^1 schemes is first to construct a basis OT_2^1 scheme (where m_1 and m_2 are bits) and then to construct the OT_n^1 scheme by (explicitly or implicitly) invoking the basis OT_2^1 scheme for many runs, typically, n or $\log_2 n$ runs [9, 11, 26]. Another approach is to build OT_n^1 schemes from basic techniques directly [29, 30, 33, 35]

In this paper we propose a very efficient OT_n^1 scheme for any $n \geq 2$ even when the secrets m_i's are strings. We build our OT_n^1 scheme from fundamental cryptographic techniques directly. It achieves optimal efficiency in the number of rounds and the total number of exchanged messages for the case that R's choice is unconditionally secure. The computation time of our OT_n^1 scheme is very efficient, too. R need compute 2 modular exponentiations only no matter how large n is, and S need compute $2n$ modular exponentiations. Furthermore, the system-wide parameters need not change during the lifetime of the system and are *universally usable*. That is, all possible R's and S's use the same parameters and need no trapdoors (eg. factoring of $N = pq$) specific to each of them. For our OT_n^1 scheme, the privacy of R's choice α is unconditionally secure and the privacy of the un-chosen secrets m_i, $i \neq \alpha$, is at least as strong as the hardness of the decisional Diffie-Hellman problem.

Our OT_n^1 scheme can be parallelized to construct an OT_n^k scheme, in which R can get k secrets among n secrets at its choice. Our OT_n^k schemes are very efficient, too.

Furthermore, we can combine our OT_n^1 scheme with any secret sharing scheme to form an efficient threshold (distributed) oblivious transfer scheme. The concept of threshold (distributed) oblivious transfer is formally proposed by Noar and Pinkas [28], though it was used in some other contents [32]. For distributed oblivious transfer, there are p servers additionally. Each server holds partial information about the secrets m_i's. If R contacts t (the threshold) or more servers, it can compute m_α of its choice; otherwise, it cannot get any information about the secrets. R's choice should not be revealed by a coalition of t' servers for some threshold $t' \leq p$. Our threshold OT_n^1 scheme takes full advantage of the research results of secret sharing and is conceptually simple. It achieves better security than Noar and Pinkas's scheme does in many aspects. For example, our scheme is secure against collusion of R and t-1 servers and it need not restrict R to contact at most t servers, which is difficult to enforce.

In the end we further extend threshold oblivious transfer to access-structure oblivious transfer (Γ-OT_n^k). Our Γ-OT_n^1 scheme is very efficient, too.

For applications, we present a method of transforming any single-database PIR protocol into a symmetric PIR (SPIR) protocol with only one extra unit of communication cost. As SPIR is equivalent to OT_n^1, this transformation provides a reduction from PIR to OT_n^1 with almost no extra communication cost. In particular, any computational PIR [25], in which the receiver's choice is computationally private, with efficient communication complexity can be transformed to an OT_n^1 scheme (with R's choice is computationally secure) with almost the same efficiency for communication complexity. Some communication-efficient single-database PIR schemes have been proposed [13, 25].

1.1 Previous work and comparison

Oblivious transfer has been studied in various flavors and security models extensively (cf. [1, 4, 7, 9, 11, 17, 21, 26, 30, 33, 35]). In particular, bit OT_2^1 (where m_1 and m_2 are bits) attracts much attention from researchers since it is the basis oblivious transfer scheme to which string OT_2^1 and OT_n^1 schemes are reduced. Most previous oblivious transfer schemes are based on hardness of factoring or quadratic residuosity problems except the one in [4], which is based on hardness of computing discrete logarithm.

The reduction approach is studied in [8, 9, 11, 15, 26]. For example, a k-bit string OT_2^1 scheme can be achieved by invoking βk runs of a bit OT_2^1 scheme for some β, $2 \leq \beta \leq 18$, [8, 9, 11]. In [26], a string OT_n^1 scheme is constructed by invoking $\log_2 n$ runs of a string OT_2^1 scheme.

The generic construction is studied in [21, 30, 33, 35, 29]. In particular, Stern [35] proposes a general construction for OT_n^1 based on any public-key encryption scheme that has some specific properties, such as, the property of "additive" homomorphism. The privacy of the receiver's choice of the scheme is computationally secure. Stern focuses on providing a zero-knowledge proof for verifying that n committed bits consist of exact one 1 and $n-1$ 0's. The numbers of rounds and exchanged messages are less efficient than ours with the same measuring parameters. For example, the scheme takes $O(\sqrt{\log_2 n})$ rounds if better efficiency for exchanged messages is desired.

Naor and Pinkas [29] proposes the first two-round oblivious transfer protocol, which achieves unconditional security for the sender's choice also. In comparison, the system parameter of our protocol is a constant, while theirs is $O(n)$. Furthermore, our protocol can be extended to threshold oblivious transfer easily and transfer any PIR protocol into a SPIR protocol without increasing communication complexity.

Distributed oblivious transfer has been studied in various contents under variant models, such as function evaluation [3] and private information retrieval [22]. Naor and Pinkas [28] identify the important attributes of distributed oblivious transfer. They propose a threshold (string) OT_2^1 scheme such that R and the involved servers need do polynomial evaluation only. Nevertheless, it comes with cost of privacy and simplicity. For example, a coalition of less than t servers can compute R's choice. One scheme (based on sparse polynomials) is not secure against collusion of R and a single server. Some schemes cannot prevent R from learning linear combination of secrets. Furthermore R cannot contact more than t servers; otherwise, the scheme is not secure.

In some sense, our schemes fall in the category of non-interactive oblivious transfer [4, 34], in which R selects a public key and S performs non-interactive oblivious transfer using R's public key. The schemes in [34] are based on the quadratic residuosity assumption. Each R uses a specific Blum integer N that is re-usable by the R. The privacy of R's choice is computationally secure and the privacy of the un-chosen secret is unconditionally secure. The bit OT_2^1 scheme is extended to the bit OT_n^1 scheme. The k-bit string OT_2^1 scheme invokes k runs of the bit OT_2^1 scheme. The number (size) of exchanged messages is not as efficient as ours. For example, if k is close to the security parameter, our k-bit string OT_2^1 scheme exchanges $O(k)$ bits and that of [34] exchanges $O(k^2)$ bits.

Transforming any PIR scheme to a symmetric PIR scheme has been studied in [19, 26]. Naor and Pinkas [26] show such a reduction using one call to the base PIR scheme and a logarithmic number of calls to a string OT_2^1 scheme. Crescenzo, etc [19], show a reduction using communication $poly(k)$ times of that of the base PIR scheme, where k is the security parameter. In comparison, our reduction uses only one extra communication cost.

2 1-out-n Oblivious Transfer

An OT_n^1 scheme is a two-party protocol in which the sender S possesses n (string) secrets $m_1, m_2, \ldots, m_n$ and would like to reveal one of them to the receiver R at R's choice. We assume that S is honest, that is, it won't send secrets that are not the same as claimed ones, either in content or in order. An OT_n^1 scheme should meet the following requirements:

1. Correctness: if both R and S follow the protocol, R gets m_α after executing the protocol with S, where α is its choice.

2. Receiver's privacy: after executing the protocol with R, S shall not get information about R's choice α.

3. Sender's privacy: after executing the protocol with S, R gets no information about other m_i's or their combinations, $i \neq \alpha$,

Let g and h be two generators in G_q, which is an order-q group, where q is prime. Let $x \in_R X$ denote that x is chosen uniformly and independently from the set X. We assume that the decisional Diffie-Hellman (DDH) problem over G_q is hard. That is, it is not possible to distinguish the following two distribution ensembles with a non-negligible advantage in polynomial time:

– $D = \{D_{G_q}\} = \{(g, g^a, g^b, g^{ab})\}_{G_q}$, where $g \in_R G_q \backslash \{1\}$ and $a, b \in_R Z_q$;

– $R = \{R_{G_q}\} = \{(g, g^a, g^b, g^c)\}_{G_q}$, where $g \in_R G_q \backslash \{1\}$ and $a, b, c \in_R Z_q$.

For simplicity, we omit the security parameter size(q) in the later arguments. Note that the DDH assumption is stronger than the discrete logarithm assumption. Typically, G_q is the set of quadratic residues of Z_p^*, where $p = 2q + 1$ is also prime. Any element in $G_q \backslash \{1\}$ is a generator of G_q.

The system-wide parameters are (g, h, G_q), which can be used by all possible senders and receivers. Assume that the discrete logarithm $\log_g h$ is unknown to all. As long as $\log_g h$ is not revealed, g and h can be used repeatedly. Our OT_n^1 scheme is as follows. Wlog, we assume that the secrets m_i's are all in G_q.

$\mathbf{OT_n^1}$ scheme:

- S's input: $m_1, m_2, \ldots, m_n \in G_q$; R's choice: α, $1 \le \alpha \le n$;

1. R sends $y = g^r h^\alpha$, $r \in_R Z_q$.
2. S sends $c_i = (g^{k_i}, m_i(y/h^i)^{k_i})$, $k_i \in_R Z_q$, $1 \le i \le n$;
3. By $c_\alpha = (a, b)$, R computes $m_\alpha = b/a^r$.

Correctness. Since $c_\alpha = (a, b) = (g^{k_\alpha}, m_\alpha(y/h^\alpha)^{k_\alpha})$, we have

$$b/a^r = m_\alpha(y/h^\alpha)^{k_\alpha}/(g^{k_\alpha})^r = m_\alpha(g^r h^\alpha/h^\alpha)^{k_\alpha}/(g^{k_\alpha})^r = m_\alpha.$$

Efficiency. The scheme takes only two rounds. This is optimal since at least R has to choose α and let S know, and S has to respond to R's request. R sends one message y to S and S sends n messages c_i, $1 \le i \le n$, to R. This is also optimal (within a constant factor of 2) by the argument for the lower bound $\Omega(n)$ of communication cost of the single-database PIR when R's choice is unconditionally secure [14].

For computation, R need do 2 modular exponentiations for y and m_α. S need do $2n$ modular exponentiations for c_i, in which g^{k_i} can be pre-computed, $1 \le i \le n$. This is very efficient, too.

Security. The above OT_n^1 scheme has the properties that the choice α of R is unconditionally secure and R gets no information about any other m_i, $i \ne \alpha$, if the DDH problem is hard.

Theorem 2.1 *For the OT_n^1 scheme, the choice α of R is unconditionally secure.*

Proof. For any α', there is r' that satisfies $y = g^{r'} h^{\alpha'}$. Therefore, S cannot get any information about R's α even if it has unlimited computing power. $\square$

Theorem 2.2 *For the OT_n^1 scheme, if R follows the protocol, it gets no information about m_i, $1 \le i \ne \alpha \le n$, assuming that the DDH problem is hard. That is, all c_i's, $1 \le i \ne \alpha \le n$, are computationally indistinguishable from a random $z = (g, h, a, b)$, $g, h \in_R G_q \backslash \{1\}$, $a, b \in_R G_q$, even if R knows the r and α in $y = g^r h^\alpha$.*

Proof. Since the DDH assumption is stronger than the DL assumption, R cannot compute two different pairs of (r, α) and (r', α') that both satisfy $y = g^r h^\alpha = g^{r'} h^{\alpha'}$. Otherwise, R computes $\log_g h = (r' - r)/(\alpha - \alpha')$. Therefore, R cannot get two secrets.

We show that c_i, $i \ne \alpha$, looks random assuming that the DDH problem is hard. Formally, we define the random variable of c_i as

$$C_i = (g, h, g^{k_i}, m_i(g^r h^{\alpha - i})^{k_i})$$

where $k_i \in_R Z_q$, $g, h \in_R G_q \backslash \{1\}$. Note that we treat g and h as random variables in C_i. Let $Z = (r_1, r_2, r_3, r_4)$, where $r_1, r_2 \in_R G_q \backslash \{1\}$ and $r_3, r_4 \in_R G_q$. We show that if C_i and Z are computationally distinguishable by distinguisher $\mathcal{A}$, D and R of the DDH problem are computationally distinguishable by the following $\mathcal{A}'$, which uses $\mathcal{A}$ as a procedure:

- Input: (g, u, v, w); (which is either from R or D)

1. If $u \neq 1$, let $h = u$; otherwise, output 1 if and only if $u = 1$;
2. Feed $(g, u, v, m_i v^r w^{\alpha-i})$ to A.
3. $A(g, u, v, m_i v^r w^{\alpha-i}) = 1$ if and only if output 1.

We can see that if $(g, u, v, w) = (g, g^a, g^b, g^{ab})$ is from D and $a \neq 0$,

$$(g, u, v, m_i v^r w^{\alpha-i}) = (g, h, g^b, m_i(g^r h^{\alpha-i})^b)$$

has the right form for C_i. If $(g, u, v, w) = (g, g^a, g^b, g^c)$ is from R and $a \neq 0$,

$$(g, h, v, m_i v^r w^{\alpha-i}) = (g, h, g^b, m_i g^{br+c(\alpha-i)})$$

is uniformly distributed over $G_q \backslash \{1\} \times G_q \backslash \{1\} \times G_q \times G_q$, which is Z. Therefore, if $\mathcal{A}$ distinguishes C_i and Z with a non-negligible advantage ϵ, $\mathcal{A}'$ distinguishes R and D with an advantage $\epsilon \cdot (1 - 1/q) + 1/q$, where $1/q$ is the offset probability in Step 1. $\square$

2.1 Without System-Wide Parameters

We can remove the requirement of using system-wide parameters (g, h, G_q). Now, S first chooses g, h and G_q, and sends them to R, that is, the following step is added to the scheme.

0. S chooses (g, h, G_q) and sends them to R, where $g, h \in_R G_q \backslash \{1\}$.

Even if S knows $\log_g h$, R's choice α is still unconditionally secure.

2.2 Forcing R to obtain $\mathbf{m}_\alpha$

R may compute y of some special form such that it can compute combined information of the secrets. We don't know whether such y exists. To prevent this attack, we can require R to send a non-interactive zero-knowledge proof of knowledge of his knowing r and α that satisfy $y = g^r h^\alpha$, denoted by NI-ZKIP(g, h, y). The new step 1 of the OT_n^1 scheme becomes:

1′. R sends $y = g^r h^\alpha$ and $\beta =$ NI-ZKIP(g, h, y), where $r \in_R Z_q$.

In this case, S should check validity of NI-ZKIP(g, h, y) in Step 2. If the check fails, S aborts the protocol. In fact, this modification results in a very secure OT_n^1 scheme. We shall discuss this in Section 7.

The property of forcing R to get some m_α may be useful in some applications. For example, the sender can be assured that R cannot later deny that he has gotten some m_α.

3 k-out-n Oblivious Transfer

We can have k parallel runs of the OT_n^1 scheme to obtain an efficient OT_n^k scheme, which takes only two rounds.

$\mathbf{OT_n^k}$ scheme:

- S's input: $m_1, m_2, \ldots, m_n$; R's choice: $\alpha_1, \alpha_2, \ldots, \alpha_k$, where $1 \leq \alpha_i \leq n$, $1 \leq i \leq k$;

1. R sends $y_l = g^{r_l} h^{\alpha_l}$, $r_l \in_R Z_q$, $1 \leq l \leq k$.
2. S sends $c_{i,l} = (g^{k_{i,l}}, m_i(y_l/h^i)^{k_{i,l}})$, $k_{i,l} \in_R Z_q$, $1 \leq l \leq k$, $1 \leq i \leq n$;
3. By $c_{\alpha_l,l} = (a, b)$, R computes $m_{\alpha_l} = b/a^{r_l}$. $1 \leq l \leq k$.

We can show that the OT_n^k scheme has the same correctness and security properties as those of the OT_n^1 scheme.

4 Threshold Oblivious Transfer

For a threshold t-out-of-p OT_n^1 (or (t,p)-OT_n^1) scheme, there are three types of parties: one sender S, p servers $S_1, S_2, \ldots, S_p$, and one receiver R. S has n secrets $m_1, m_2, \ldots, m_n$. It computes shares $m_{i,j}$, $1 \leq j \leq p$, of m_i, $1 \leq i \leq n$, and distributed shares $m_{i,j}$, $1 \leq i \leq n$, to server S_j, $1 \leq j \leq p$. Then, R chooses α, $1 \leq \alpha \leq n$, and contacts with any t or more servers to get information about the shares. By the received information, R should be able to compute m_α and no others.

By [28], a (t,p)-OT_n^1 scheme should meet the following requirements:

1. Correctness: if R and servers follow the protocol and R receives information from t or more servers, R can compute one m_α, where α is its choice.
2. Sender's privacy: even if R receives information from t or more servers, it gains no information about any other m_i, $1 \leq i \neq \alpha \leq n$. Furthermore, if R receives information from less than t servers, it gains no information about any m_i, $1 \leq i \leq n$.
3. Receiver's privacy: there is a threshold t', $t' \geq 1$, such that no coalition of less than t' servers can gain any information about the choice α of R. The threshold t' should be as large as possible.
4. Security against receiver-server collusion: after R gets m_α, there is a threshold t'', $1 \leq t'' \leq t$, such that no coalition of less than t'' servers and R can gain any information about any other m_i, $1 \leq i \neq \alpha \leq n$. The threshold t'' should be as close to t as possible.

By the OT_n^1 scheme in Section 2, we can easily construct a threshold (t,p)-OT_n^1 scheme. Our scheme can make use of any threshold secret sharing scheme. Our (t,p)-OT_n^1 scheme achieves $t' = \infty$ and $t'' = t$. Both are optimal.

We construct our (t,p)-OT_n^1 scheme using the standard (t,p)-secret-sharing scheme. Let m_i be shared by the servers via polynomial $f_i(x)$ of degree t-1 such that $f_i(0) = m_i$, $1 \leq i \leq n$. Each server S_j, $1 \leq j \leq p$, holds the shares $m_{i,j} = f_i(j)$, $1 \leq i \leq n$. By contacting t servers, R can compute t shares of $m_{\alpha,j}$'s and construct m_α, where α is R's choice. Our (t,p)-OT_n^1 scheme is as follows.

$\mathbf{(t,p)-OT_n^1}$ **scheme:**

– S_j's input: $m_{1,j}, m_{2,j}, \ldots, m_{n,j}$; R's choice: α, $1 \le \alpha \le n$;

1. R sends $y = g^r h^\alpha$ to t different servers $S_{j_1}, S_{j_2}, \ldots, S_{j_t}$, $r \in Z_q$;
2. Each S_{j_l}, $1 \le l \le t$, sends $c_{i,j_l} = (g^{k_{i,j_l}}, m_{i,j_l}(y/h^i)^{k_{i,j_l}})$, $1 \le i \le n$, to R;
3. By $c_{\alpha,j_l} = (a_{j_l}, b_{j_l})$, R computes shares $m_{\alpha,j_l} = b_{j_l}/a_{j_l}^r$, $1 \le l \le t$. Then, R interpolates these t shares to get

$$m_\alpha = \sum_{l=1}^{t} m_{\alpha,j_l}(\prod_{1 \le d \ne l \le t} \frac{j_d}{j_d - j_l})$$

by Lagrange's interpolation method.

Correctness. If R contacts with t or more servers, it can compute t shares m_{α,j_l} of m_α, $1 \le l \le t$. Therefore, it can compute m_α as shown in the scheme.

Efficiency. The scheme takes only two rounds. This is optimal, again. R sends one message y to t servers and each contacted server S_j responds with n messages $c_{i,j}$, $1 \le i \le n$. This is very efficient. For computation, R need do $t+1$ modular exponentiations for y and t shares m_{α,j_l}, $1 \le l \le t$, and one Lagrange interpolation for m_α. Each contacted server S_j need do $2n$ modular exponentiations for $c_{i,j}$, $1 \le i \le n$.

Security. Our (t,p)-OT_n^1 scheme has the following security properties:

1. Sender's privacy: if R contacts with t or more servers, the privacy of m_i, $1 \le i \ne \alpha \le n$, is at least as strong as the hardness of the the DDH problem. (The proof is similar to that of Theorem 2.2.) Furthermore, if R gets information from less than t servers, R cannot compute information about any m_i, $1 \le i \le n$. This is guaranteed by the polynomial secret sharing scheme we use.

2. Receiver's privacy is unconditionally secure. Since for any α', there is r' that satisfies $y = g^{r'} h^{\alpha'}$. Even if the servers have unlimited computing power, they cannot compute R's choice α.

3. It is secure against collusion of R and t-1 servers $S_{r_1}, S_{r_2}, \ldots, S_{r_{t-1}}$, assuming the hardness of the DDH problem. Since for R and $S_{r_l}, 1 \le l \le t-1$, the privacy of shares $m_{i,j}$, $i \ne \alpha$, $j \ne r_1, r_2, \ldots, r_{t-1}$, is at least as strong as the hardness of the DDH problem, R and these t-1 servers cannot compute any information about other secrets m_i, $1 \le i \ne \alpha \le n$.

4.1 $\mathbf{(t,p)}$-$\mathbf{OT_n^k}$ Scheme

We can extend the (t,p)-OT_n^1 scheme to a (t,p)-OT_n^k scheme easily. This is done by executing k parallel runs of the (t,p)-OT_n^1 scheme, similar to the OT_n^k scheme in Section 4.

4.2 $(\mathbf{p},\mathbf{p})$-$\mathbf{OT_n^1}$ Scheme

For (p,p)-OT_n^1, we can use $m_i = m_{i,1}m_{i,2}\cdots m_{i,p}$ to share m_i. Then, R can compute $m_\alpha = (b_1b_2\cdots b_p)/(a_1a_2\cdots a_p)^r$. It need do $2p-1$ modular multiplications and one modular exponentiations, which is very efficient.

4.3 Verifiable $(\mathbf{t},\mathbf{p})$-$\mathbf{OT_n^k}$ Scheme

We can combine Feldman's or Peterson's verifiable secret sharing scheme and our OT_n^k scheme to form a verifiable (t,p)-OT_n^k scheme. In this case, the sender S, who has all m_i's, publishes the verification values for m_i's. Typically, the verification values for the shares of m_i are $g^{a_0}, g^{a_1}, \ldots, g^{a_{t-1}}$, where m_i is shared via a degree-(t-1) polynomial $f_i(x) = m_i + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}$. After computing m_{α,j_l}, $1 \le l \le t$, R can verify these shares using the verification values published by S.

5 Access-Structure Oblivious Transfer

Let $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_z\}$ be a monotonic access structure over p servers $S_1, S_2, \ldots, S_p$. Each $\gamma_i = \{S_{i_1}, S_{i_2}, \ldots, S_{i_l}\}$ is an authorized set of servers such that all servers in γ_i together can construct the shared secret. Assume that n messages $m_1, m_2, \ldots, m_n$ are shared according to Γ by some secret sharing scheme $\mathcal{S}$ such that $S(\gamma) = (m_1, m_2, \ldots, m_n)$ if and only if $\gamma \in \Gamma$, where $\mathcal{S}(\gamma)$ means that $\mathcal{S}$ computes shared secrets from shares of the servers in γ.

We define Γ-OT_n^1 such that R can get the secret m_α from the servers in an authorized set $\gamma \in \Gamma$, where α is R's choice. The requirements for a satisfactory Γ-OT_n^1 are the same as those for the threshold OT_n^1 schemes in Section 4.

We can combine our OT_n^1 scheme and a general secret sharing scheme $\mathcal{S}$ to form a Γ-OT_n^1 scheme as follows.

1. Let S_j obtain a share $m_{i,j}$ of m_i by the secret sharing scheme S, $1 \le i \le n$.
2. Let γ be an authorized set that R contacts its servers to obtain m_α. When R contacts $S_j \in \gamma$ with $y = g^r h^\alpha$, S_j responds with $c_{i,j} = (g^{k_{i,j}}, m_{i,j}(y/h^i)^{k_{i,j}})$, $1 \le i \le n$.
3. R computes $m_{\alpha,j}$ for each $S_j \in \gamma$ and applies $S(\gamma)$ to compute m_α.

The above Γ-OT_n^1 scheme meets the requirements. This can be proved by the same arguments for the threshold oblivious transfer schemes in Section 4. We omit them here.

6 Applications

Efficient string OT_n^1 schemes can improve practical efficiency of the schemes in which oblivious transfer is used. One primary application is for private information retrieval (PIR), in which the user (U) wants to query one data block from a database, but U does not want the database manager (DBM) to know which data block he is interested in [14]. The regular PIR does not restrict U to obtain only one data block of the database. We consider the symmetric PIR (SPIR), in which DBM does not want to release more than one data block [22].

Assume that the database has n data blocks m_i's, each is in G_q. The following steps achieve SPIR. U wants to obtain m_α.

1. U sends $y = g^r h^\alpha$ to DBM;
2. DBM computes $c_i = (g^{k_i}, m_i(y/h^i)^{k_i})$, $1 \leq i \leq n$;
3. Now, DBM treats c_i's as its data blocks. DBM and U perform a regular PIR protocol so that U obtains c_α.
4. U computes $m_\alpha = b/a^r$, where $c_\alpha = (a, b)$.

This method transforms any single-database PIR scheme into a single-database SPIR scheme with only an extra unit of communication cost in step 1. If U's choice α of the base PIR scheme in Step 3 is computationally private, the transformed SPIR scheme's user privacy is computationally secure. On the other hand, if the base PIR scheme is unconditionally secure, the user's choice of the transformed SPIR is unconditionally private.

The transformed SPIR scheme uses at most one more round than that of the base PIR scheme. The reason is that the fist step may be combined with the first step of the base PIR in step 3.

Theorem 6.1 *If there exists a single-database PIR scheme with communication complexity $c(n)$ and round complexity $r(n)$, there exists an OT_n^1 scheme with communication complexity $c(n)+1$ and round complexity $r(n)+1$, but with the additional assumption of hardness of the DDH problem.*

7 Further Security

Naor and Pinkas [26] give a very formal definition for security of OT_n^1 oblivious transfer:

1. Receiver's privacy – indistinguishability: S's views of R's different choices α and α', $\alpha \neq \alpha'$, are computationally indistinguishable.
2. Sender's privacy – compared with Ideal Model: The Ideal Model is that there is a trusted third party (TTP) that gets S's secrets $m_1, m_2, \ldots, m_n$ and R's choice α and gives m_α to R. Sender's secrecy is that for every probabilistic poly-time substitute R' for R, there is a corresponding R'' in the Ideal Model such that the outputs of R' and R'' are computationally indistinguishable.

The modified OT_n^1 scheme, consisting of Steps $1'$, 2 and 3, in Section 2.2 meets both requirements.

Theorem 7.1 *The modified OT_n^1 scheme, consisting of Steps $1'$, 2 and 3, in Section 2.2 meets both the requirements of Receiver's privacy and Sender's privacy above.*

Proof. (Sketch) Since R's choice is unconditionally secure, the scheme meets the requirement of Receiver's privacy.

For each probabilistic polynomial-time adversary R', substituting for R, in the real run, we can construct a corresponding R'' (in the Ideal Model) whose output is computationally indistinguishable from that of R' as follows. R'' uses R' as a *re-settable* subroutine. When R' sends y and $\beta = \text{NI-ZKIP}(g, h, y)$ to S, R'' simulates R' to get α in a re-settable way with an overwhelming probability. If β is not legal or the simulation fails to produce α, TTP outputs $\perp$ (*abort*). The probability of TTP outputting $\perp$ is almost equal to that of S outputting $\perp$. After obtaining α, R'' sends α to TTP and gets m_α. R'' sets $c_\alpha = (g^k, m_\alpha(y/h^\alpha)^k)$ and $c_i = (a_i, b_i)$ for $1 \leq i \neq \alpha \leq n$, $a_i, b_i \in_R G_q$, and outputs the simulation result of R' on $c_1, c_2, \ldots, c_n$. The output of R'' is computationally indistinguishable from that of R'. If there is a claim that R' gets information about $m_{\alpha'}$, $\alpha' \neq \alpha$. We can use R' to solve the DDH problem by manipulating its input c_i's, which is similar to the proof of Theorem 2.2. Therefore, the scheme meets the requirement of Sender's privacy. □

8 Conclusion

The current trend of research on the design of cryptographic protocols is to find provably-secure practical protocols. Our results are along this direction. We have presented a very efficient (string) OT_n^1 scheme and extended it to construct threshold, access-structure and verifiable OT_n^k schemes for any $n \geq 2$ and $1 \leq k \leq n$. We also present its application on private information retrieval. It is interesting to find more applications of this construction.

References

[1] D. Beaver, "How to break a 'secure' oblivious transfer protocols," *In Proceedings of Advances in Cryptology - Eurocrypt 92*, Lecture Notes in Computer Science 658, pp.285-196, Springer-Verlag, 1993.

[2] D. Beaver, "Equivocable oblivious transfer," *In Proceedings of Advances in Cryptology - Eurocrypt 96*, Lecture Notes in Computer Science 1070, pp.119-130, Springer-Verlag, 1996.

[3] D. Beaver, J. Feigenbaum, J. Kilian,P. Rogaway, "Locally random reductions: improvements and applications," *Journal of Cryptology* 10(1), pp.17-36, 1997.

[4] M. Bellare, S. Micali, "Non-interactive oblivious transfer," *In Proceedings of Advances in Cryptology - Crypto 89*, Lecture Notes in Computer Science 435, pp.547-557, Springer-Verlag, 1990.

[5] M. Ben-Or, S. Goldwasser, A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation", *In Proceedings of the 20th ACM Symposium on the Theory of Computing*, pp.1-10, 1988.

[6] R. Berger, R. Peralta, T. Tedrick, "A provably secure oblivious transfer protocol," *In Proceedings of Advances in Cryptology - Eurocrypt 84*, Lecture Notes in Computer Science 209, pp.379-386, Springer-Verlag, 1985.

[7] B. den Boer, "Oblivious transfer protecting secrecy," *In Proceedings of Advances in Cryptology - Eurocrypt 90*, Lecture Notes in Computer Science 473, pp.31-45, Springer-Verlag, 1991.

[8] G. Brassard, C. Crépeau, "Oblivious transfers and privacy amplification," *In Proceedings of Advances in Cryptology - Eurocrypt 97*, Lecture Notes in Computer Science 1233, pp.334-346, Springer-Verlag, 1997.

[9] G. Brassard, C. Crépeau, J.-M. Robert, "Information theoretic reduction among disclosure problems," *In Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pp.168-173, 1986.

[10] G. Brassard, C. Crépeau, J.-M. Robert, "All-or-nothing disclosure of secrets," *In Proceedings of Advances in Cryptology - Crypto 86*, Lecture Notes in Computer Science 263, pp.234-238, Springer-Verlag, 1987.

[11] G. Brassard, C. Crépeau, M. Santha, "Oblivious transfer and intersecting codes," *IEEE Transactions on Information Theory* 42(6), pp.1769-1780, 1996.

[12] C. Cachin, "On the foundations of oblivious transfer," *In Proceedings of Advances in Cryptology - Eurocrypt 98*, Lecture Notes in Computer Science 1403, pp.361-374, Springer-Verlag, 1998.

[13] C. Cachin, S.Micali, M. Stadler, "Computationally private informational retrieval with polylogarithmic communication," *In Proceedings of Advances in Cryptology - Eurocrypt 99*, Lecture Notes in Computer Science 1592, pp.402-414, Springer-Verlag, 1999.

[14] B. Chor, O. Goldreich, E. Kushilevitz, M. Susdan, "Private information retrieval," *Journal of the ACM* 45(6), pp.965-982, 1998.

[15] C. Crépeau, "Equivalence between two flavors of oblivious transfers," *In Proceedings of Advances in Cryptology - Crypto 87*, Lecture Notes in Computer Science 293, pp.350-354, Springer-Verlag, 1988.

[16] C. Crépeau, "Verifiable disclosure of secrets and application", *In Proceedings of Advances in Cryptology - Eurocrypt 89*, Lecture Notes in Computer Science 434, pp.150-154, Springer-Verlag, 1990.

[17] C. Crépeau, J. van de Graff, A. Tapp, "Committed oblivious transfer and private multi-party computations," *In Proceedings of Advances in Cryptology - Crypto 95*, Lecture Notes in Computer Science 963, pp.110-123, Springer-Verlag, 1995.

[18] C. Crépeau, J. Kilian, "Achieving oblivious transfer using weakened security assumptions," *In Proceedings of the 29th IEEE Symposium on Foundations of Computer Science*, pp.42-52, 1988.

[19] G.Di Crescenzo, T. Malkin, R. Ostrovsky, "Single database private information retrieval implies oblivious transfer," *In Proceedings of Advances in Cryptology - Eurocrypt 00*, Lecture Notes in Computer Science , pp.122-138, Springer-Verlag, 2000.

[20] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory* 31(4), pp.469-472, 1985.

[21] S. Even, O. Goldreich, A. Lempel, "A randomized protocol for signing contracts," *Communications of the ACM* 28, pp.637-647, 1985.

[22] Y. Gertner, Y. Ishai, E. Kushilevitz, T. Malkin, "Protecting data privacy in private data retrieval schemes," *In Proceedings of the 30th ACM Symposium on Theory of Computing*, pp.151-160, 1998.

[23] O. Goldreich, R. Vainish, "How to solve any protocol problem: an efficient improvement," *In Proceedings of Advances in Cryptology - Crypto 87*, Lecture Notes in Computer Science 293, pp.73-86, Springer-Verlag, 1988.

[24] J. Kilian, "Founding cryptography on oblivious transfer," *In Proceedings of the 20th ACM Symposium on Theory of Computing*, pp.20-31, 1988.

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

[26] M. Naor, B. Pinkas, "Oblivious transfer and polynomial evaluation," *In Proceedings of the 31st ACM Symposium on Theory of Computing*, pp.145-254, 1999.

[27] M. Naor, B. Pinkas, "Oblivious transfer with adaptive queries," *In Proceedings of Advances in Cryptology - Crypto 99*, Lecture Notes in Computer Science 1666, pp.573-590, Springer-Verlag, 1999.

[28] M. Naor, B. Pinkas, "Distributed oblivious transfer," *In Proceedings of Advances in Cryptology - Asiacrypt 00*, Lecture Notes in Computer Science 1976, pp.205-219, Springer-Verlag, 2000.

[29] M. Naor, B. Pinkas, "Efficient oblivious transfer protocols," *In Proceedings of SODA 01*, 2001.

[30] V. Niemi, A.Renvall, "Cryptographic protocols and voting," *In Result and Trends in Theoretical Computer Science*, Lecture Notes in Computer Science 812, pp.307-316, 1994.

[31] M. Rabin, "How to exchange secrets by oblivious transfer," Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[32] R. Rivest, "Unconditionally secure commitment and oblivious transfer schemes using private channels and a trusted initializer," manuscript. Available at http://theory.lcs.mit.edu/~rivest/publications.html.

[33] A. Salomaa, L. Santean, "Secret selling of secrets with several buyers," *In the 42nd EATCS Bulletin*, pp.178-186, 1990.

[34] A. De Santis, G. Persiano, "Public-randomness in public-key cryptography," *In Proceedings of Advances in Cryptology - Eurocrypt 90*, Lecture Notes in Computer Science 473, pp.46-62, Springer-Verlag, 1991.

[35] J.P. Stern, "A new and efficient all-or-nothing disclosure of secrets protocol," *In Proceedings of Advances in Cryptology - Asiacrypt 98*, Lecture Notes in Computer Science 1514, pp.357-371, Springer-Verlag, 1998.