Hindawi Publishing Corporation
Advances in Multimedia
Volume 2012, Article ID 630610, 9 pages
doi:10.1155/2012/630610

Research Article

A Novel *k*-out-of-*n* Oblivious Transfer Protocol from Bilinear Pairing

Jue-Sam Chou

Department of Information Management, Nanhua University, No. 55, Section 1, Nanhua Road, Dalin Township, Chiayi County 62249, Taiwan

Correspondence should be addressed to Jue-Sam Chou, jschou@mail.nhu.edu.tw

Received 30 November 2011; Revised 13 March 2012; Accepted 27 March 2012

Academic Editor: Mohamed Hamdi

Oblivious transfer (OT) protocols mainly contain three categories: 1-out-of-2 OT, 1-out-of-*n* OT, and *k*-out-of-*n* OT. In most cases, they are treated as cryptographic primitives and are usually executed without consideration of possible attacks that might frequently occur in an open network, such as an impersonation, replaying, or man-in-the-middle attack. Therefore, when used in certain applications, such as mental poker games and fair contract signings, some extra mechanisms must be combined to ensure the security of the protocol. However, after a combination, we found that very few of the resulting schemes are efficient enough in terms of communicational cost, which is a significant concern for generic commercial transactions. Therefore, we propose a novel *k*-out-of-*n* oblivious transfer protocol based on bilinear pairing, which not only satisfies the requirements of a *k*-out-of-*n* OT protocol, but also provides mutual authentication to resist malicious attacks. Meanwhile, it is efficient in terms of communication cost.

1. Introduction

An oblivious transfer (OT) is an important primitive for designing security services. It can be used in various applications like the signing of fair contracts, oblivious database searches, mental poker games, privacy-preserving auctions, secure multiparty computations [1], and so on. In 1981, Rabin [2] first proposed an interactive OT scheme in which the probability of the receiver's capability to decrypt a message sent by the sender is 1/2. Rabin used the proposed OT to design a 3-pass secret exchange (EOS) protocol, hoping that two parties can exchange their secrets fairly. In 1985, Even et al. [3] presented a more generalized OT, called 1-out-of-2 OT (OT_1^2), in which a sender sends two encrypted messages to a chooser with only one of which the chooser can decrypt. They also presented a contract-signing protocol by evoking OT_1^2 multiple times to prevent one party from obtaining the other party's contract signature without first showing his own. In 1986, Brassard et al. [4] further extended OT_1^2 into a 1-out-of-*n* OT (OT_1^n, also known as "all-or-nothing"), in which only one out of *n* sent messages can actually be obtained by the chooser. The authors pointed out that their OT_1^n scheme can be used to implement a multiparty mental poker game [5] against a player coalition. In contrast to the interactive versions described above, Bellare and Micali [6] first proposed a noninteractive OT_1^2 scheme in 1989. In this scheme, a user obliviously transfers two messages to another party equipped with two public keys to decrypt one of the messages.

From 1999 to 2001, based on the above-mentioned interactive and noninteractive OT schemes, Naor and Pinkas proposed some related OT methods, such as an adaptive OT_k^n [7], proxy OT_1^2 [8], distributed OT_k^n [9], efficient OT_1^n [10], and efficient OT_k^n [11]. Here, OT_k^n is the final form of the OT schemes. In this form, from the *n* encrypted messages sent, the chooser can obtain *k* chosen messages in plaintext form without the sender's knowledge regarding which part of the messages are decrypted. In Naor and Pinkas's distributed OT_k^n schemes [9], the sender distributes two messages (M_0, M_1) among *n* servers, and the chooser contacts k $(k < n)$ servers to receive one $(M_\sigma, \sigma = 0 \text{ or } 1)$ of them. The authors claimed that their schemes can protect the privacy of both parties. However, in 2007, Ghodosi [12] showed two possible attacks on these schemes. In

the first attack, two collaborating servers can reveal the chooser's choice of σ, while, in the second attack, the chooser can learn both M_0 and M_1 by colluding with only a single server. In 2002, Mu et al. [13] proposed three OT_k^n schemes constructed using RSA encryption, a Nyberg-Rueppel signature, and an ElGamal encryption scheme, respectively. Two of these are interactive, while the other can be either interactive or noninteractive. The authors claimed that their schemes are complete, robust, and flexible and induce a significant improvement in communication cost. However, in 2006, Ghodosi and Zaare-Nahandi [14] showed that these schemes fail to satisfy the requirements of an oblivious transfer protocol. In 2004, Ogata and Kurosawa [15] proposed another OT_k^n scheme, based on an RSA blind signature, which can be employed in either an adaptive or a nonadaptive manner. The authors claimed that their scheme can be applied to oblivious key searching. In 2005, three OT_k^n schemes are proposed [16–18]. Among these, Chu and Tzeng's scheme [16] is the most efficient as it needs only 2 passes to send 1024 k bits from the chooser to the sender, and $1024^*(k+1) + n^*|\mathrm{Data}|$ bits from the sender to the chooser, where Data is a message or ciphertext, and $|\mathrm{Data}|$ represents the bit length of Data. In 2006, Parakh [19] proposed an elliptic-curve-based algorithm allowing A to obliviously transfer his secrecy, n_A, to B with a 50% probability of success. However, we found that A can decide whether B can obtain his secret n_A (which is one-to-one mapped to Pn_A) by first assuming that $P_A = P_B$. Under this assumption, upon receiving $\{n_BP_B; n_B(n_AP_A) + R; n_BR\}$ from B, A can obtain B's one-time random variable R by computing $(n_B(n_AP_A) + R) - n_A(n_BP_B)$. Then, by computing $n_A(n_BR) = n_B(n_AR)$, A can obtain n_BK. Subsequently, by computing $(n_A(n_B\ R) + Pn_A) - n_BK$, A obtains Z_B, just as B does in step 5(b). Therefore, if A finds $Z_B = Pn_A$, it confirms that B can obtain n_A after the protocol runs; otherwise, it knows B cannot obtain the value of n_A. This violates B's privacy. In the same year, for coping with all possible attacks encountered in an open network, Kim and Lee [20] proposed two OT_1^2 protocols, which are modified from Bellare-Micali noninteractive OT_1^2 scheme [6] by appending the sender's signature to make the sender undeniable about what he sent and be authentic to the chooser. However, we found, other than the weaknesses pointed by Chang and Shiao [21], Kohnfelder's protocol still has the reblocking problem [22]. Because when modulus $n_A > n_B$, message M_A cannot be recovered by Bob. This makes legal Alice unable to be authenticated by Bob.

In 2007, Halevi and Kalai [23] proposed another OT_1^2 scheme by using smooth projective hashing and showed that the used RSA composite in their scheme need not be a product of safe primes. Also in 2007, Camenish et al. and Green and Hohenberger proposed two related OT schemes [24, 25], respectively. Both focus on the security of full simulatability for the sender and receiver to resist against selective-failure attack [7]. In 2009, Qin et al. [26] proposed two noninteractive OT_1^n schemes. However, in their protocols, a receiver has to interact with a third party to obtain the choice-related secret key each time it wants to select one of the n sent message. This makes their scheme somewhat inconvenient and inconsistent with the meaning of noninteractive protocols as indicated in the title (this phenomenon can also be found in some proposed noninteractive OT schemes). In the same year, Chang and Lee [27] presented a robust OT_k^n scheme using both the RSA blind signature and Chinese Remainder Theorem. However, we found their scheme fails since the sender can decide which parts of the messages were chosen by the chooser. We will describe this weakness in Section 3.2. In addition, in 2011, Ma et al. [28] proposed an oblivious transfer using a privacy scheme for a timed-release receiver. Their scheme has a good timed-release property. However, it needs to call ZKP k times to learn k of the n sent messages. This makes their protocol less efficient. Moreover, it does not have mutual authentication. Therefore, when the sender and receiver want to communicate, they need a secure channel. Otherwise, without identity authentication, malicious attackers can simultaneously launch many ZKPs. This will degrade the system performance and may cause the system to suffer from a denial-of-service (DOS) attack (according to the definition in [29]).

After surveying all of the above-mentioned OT schemes, we found that almost all of them lack the consideration of adding security features. Only [2, 20] do consider the protection against all possible attacks. However, study [20] fails which we have described earlier. Hence, if we wish all of the proposed OT protocols, other than scheme [2], to be able to resist against various attacks, we should run them through secure channels. This would incur extra communicational overhead. For this reason, in this paper, we propose a novel interactive OT_k^n scheme that needs only two passes but can get rid of using a secure channel to avoid adding extra communicational overhead. It not only is simple in concept but also encompasses some essential security features such as mutual authentication, the prevention of man-in-the-middle (MIMA) attack, and replay attack. Thus, when compared with other interactive OT schemes, our scheme promotes not only in the communicational efficiency but also in the aspect of security.

The rest of this paper is organized as follows. The introduction has been presented in Section 1, and some preliminaries are shown in Section 2. In Section 3, we review Chang et al.'s scheme and show its weakness. After that, we show our protocol in Section 4. Then, the security analyses and communicational cost comparisons among related works and our scheme are made in Section 5. Finally, a conclusion is given in Section 6.

2. Preliminaries

In this section, we briefly introduce the security features of our OT_k^n scheme in Section 2.1, the principles of bilinear pairing in Section 2.2, and some intractable problems used in this paper in Section 2.3.

2.1. Security Features of Our OT_k^n Scheme. Just as traditional OT schemes, our OT_k^n also has two parties, the sender S and the chooser C. In the scheme, S obliviously transfers

n messages to C, and C can choose k messages among them without S's knowledge about which k messages were selected, where $n \geq 2$ and $k < n$. In addition, our scheme also possesses the following three security features which are needed in a traditional OT scheme.

(1) Correctness. After the protocol run, C should be able to obtain the valid data chosen by him before.

(2) Chooser's Privacy. In the protocol, each of the k chooser's choices should not be known to the sender or any third party. More precisely, each of the chooser's encrypted choice can be any valid choice with equal probability, that is, for an encrypted choice y and any valid choice x, $\Pr[x \mid y] = \Pr[x]$. This property is known as *Shannon perfect secrecy*.

(3) Sender's Privacy. At end of the protocol run, the chooser cannot get any knowledge about the other messages it did not choose. More formally, the ciphertexts sent by the sender are semantically secure [30]. The chooser can obtain a plaintext decrypted from its ciphertext only if it has the key offered by the sender.

Except for the above three properties, our interactive OT_k^n scheme also has the following three security features, (4) through (6), to guard against possible security threats.

(4) Impersonation Attack Resistance. Each party has to authenticate the counterpart. That is, it should be a mutual-authentication OT.

(5) Replaying Attack Resistance. An adversary could not obtain any messages by only replaying old messages sent by the sender.

(6) Man-in-the-Middle Attack (MIMA) Resistance. MIMA is an attack that an adversary eavesdropping on the communication line between two communicating parties uses as some means to make them believe that they each are talking to the intended party. But indeed, they are talking to the adversary.

2.2. Bilinear Pairing. Let G_1 be an additive group composed of points on an elliptic curve with order q, and let G_2 be a multiplicative group with the same order. A bilinear mapping is defined as $\hat{e} : G_1 \times G_1 \rightarrow G_2$ which must satisfy the following properties [31].

(1) Bilinear: a mapping $\hat{e} : G_1 \times G_1 \rightarrow G_2$ is bilinear if $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$ and all $a, b \in Z_q^*$.

(2) Nondegenerate: the mapping does not map all pairs in $G_1 \times G_1$ to the identity in G_2.

(3) Computable: there is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in G_1$.

(4) If P is a generator for G_1 then $\hat{e}(P, P)$ is a generator for G_2.

(5) Commutative: for all $P_1, P_2 \in G_1$, $\hat{e}(P_1, P_2) = \hat{e}(P_2, P_1)$.

(6) Distributive: for all $P_1, P_2, P_3 \in G_1$, $\hat{e}(P_1 + P_2, P_3) = \hat{e}(P_1, P_3)\hat{e}(P_2, P_3)$.

2.3. Some Diffie-Hellman Problems. Let $a, b, c, g \in_R Z_q^*$, let P be a base point of a group on an Elliptic curve, and let $G = \langle g \rangle$, $G_1 = \langle P \rangle$, and $G_2 = \langle g (= \hat{e}(P, P)) \rangle$ *be* three groups with each having a prime order q. Using these definitions, we describe some well-known intractable Diffie-Hellman problems [32] that will be used in this paper.

(1) The Computational Diffie-Hellman (CDH) Problem. In G, given (g, g^a, g^b), finding the element $C = g^{ab} \bmod q$.

(2) The Decisional Diffie-Hellman (DDH) Problem. In G, given (g, g^a, g^b, g^c), deciding whether $c = ab \bmod q$.

(3) The Bilinear Computational Diffie-Hellman (BCDH) Problem. Given (P, aP, bP, cP) in G_1, finding $\hat{e}(P, P)^{abc}$ in G_2.

According to Boneh and Franklin's study [31], the BCDH problem is no harder than the CDH problem in G (or equivalently G_2).

(4) Chosen-Target CDH (CTCDH) Problem. Let $H : \{0,1\}^* \rightarrow G$ be a hash function, let $T(\cdot)$ be a target oracle which returns a random element in G, and $(\cdot)^c$ a helper oracle which returns $T(j)^c$ when queried by $T(j)$, where c is an unknown random integer in Z_q^*. Also, let q_t be the number of queries to $T(\cdot)$ and q_h the number of queries to $(\cdot)^c$. The CTCDH problem is finding l pairs of $(j_1, \nu_1), \ldots,$ and (j_l, ν_l), with each satisfying $\nu_i = (T(j_i))^c$, for $1 \leq i \leq l$ and $q_h < l \leq q_t$. Without loss of generality, we can let q_h and q_t be $l - 1$ and l, respectively. The CTCDH problem can then be rephrased as that after obtaining $T(j_1), \ldots, T(j_l)$ and $(j_1, \nu_1), \ldots, (j_{l-1}, \nu_{l-1})$ via querying the $T(\cdot)$ oracle and the helper oracle $(\cdot)^c$ correspondingly, trying to find the lth pair (j_l, ν_l) without the knowledge of c. The CTCDH problem is proposed and considered as a hard problem by Boldyreva in 2002 [33]. Its former version in RSA is proved by Bellare et al. in [34].

3. Review of Chang et al.'s Protocol

In 2009, Chang et al. proposed a robust OT_k^n scheme based on CRT, hoping that their scheme can achieve the security requirements of a general OT_k^n scheme. However, we found their scheme cannot satisfy the chooser's privacy. In the following, we first review the scheme in Section 3.1 then show the weakness found in Section 3.2.

3.1. Review. We roughly describe the protocol by listing the relevant steps in the following (see [27] for more details).

Step 1. After receiving the request from Bob for all messages $a_1, a_2, \ldots, a_n$, Alice owning these n messages selects n

relatively prime integers, $d_1, d_2, \ldots, d_n$, and computes $D = d_1 * d_2 * \cdots * d_n$. She then constructs the congruence system

$$C \equiv a_1 \bmod d_1,\ C \equiv a_2 \bmod d_2, \ldots, \quad C \equiv a_n \bmod d_n. \tag{1}$$

Furthermore, Alice computes the following values: $T_1 = d_1^e \bmod N$, $T_2 = d_2^e \bmod N, \ldots$, and $T_n = d_n^e \bmod N$, where N be the product of two large primes and (e, d) be Alice public/private key pair satisfying $ed = 1 \bmod \varphi(N)$, by using her public key e. Finally, she publishes C and the n pairs of (ID_i, T_i), for $i = 1$ to n, in the public board.

Step 2. If Bob wants to learn k messages among them, he must select k pairs of (ID'_j, T'_j), for $j = 1$ to k, from the public board and first generate k corresponding random numbers $r_1, r_2, \ldots, r_k$, for each pair of (ID'_j, T'_j). Then, he subsequently computes the following:

$$\alpha_1 = r_1^e * T'_1 \bmod N,\ \alpha_2 = r_2^e * T'_2, \quad \bmod N, \ldots, \alpha_k = r_k^e * T'_k \bmod N, \tag{2}$$

by using Alice's public key e and sends $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ back to Alice.

Step 3. Upon receiving the messages sent by Bob, Alice employs her private key d to compute $\beta_1 = \alpha_1^d = r_1 T_1'^d = r_1 d'_1 \bmod N$, $\beta_2 = \alpha_2^d = r_2 T_2'^d = r_2 d'_2 \bmod N, \ldots, \beta_k = \alpha_k^d = r_k T_k'^d = r_k d'_k \bmod N$ and then sends the results $\{\beta_1, \beta_2, \ldots, \beta_k\}$ to Bob.

Step 4. After receiving the messages from Alice, Bob computes the following values: $d'_1 = r_1^{-1} * \beta_1 \bmod N, d'_2 = r_2^{-1} * \beta_2 \bmod N, d'_k = r_k^{-1} * \beta_k \bmod N$. Consequently, Bob learns the demanded messages successfully by computing

$$b_1 = C \bmod d'_1,\ b_2 = C \bmod d'_2, \ldots, \quad b_k = C \bmod d'_k. \tag{3}$$

3.2. Weaknesses. Although Chang et al. claimed that their scheme can satisfy the security requirements demanded by the OT_k^n scheme, we found that Bob's privacy has been violated, since according to their protocol, Alice first sets n values of d_i ($i = 1$ to n), and Bob commits his k choices to the k values of α_j ($j = 1$ to k). After computing the k values of β_j ($= 1$ to k), Alice can use each of the $d_i^{-1}s$ ($i = 1$ to n) to compute $r_{ji} = \beta_j * d_i^{-1}$, for $j = 1$ to k and $i = 1$ to n. In addition, using each r_{ji}, Alice can compute the n values of $\alpha_i^{(*)} = (r_{ji} * d_i)^e$, for $i = 1$ to n, to compare with the k committed values, α_j. For example, suppose Bob chooses the first message, $T_1 = {d_1}^e \bmod N$, and Alice wants to guess which T_i Bob chose, Alice starts to use ${d_1}^{-1}$ to compute $r_{11} = \beta_1 * {d_1}^{-1} \bmod N = {\alpha_1}^d (= r_1 * d_1) * {d_1}^{-1} \bmod N = r_1 \bmod N$. He will get $\alpha_1{}^{(*)} = (r_{11} * d_1)^e \bmod N = \alpha_1 = r_1^e * T_1$. That is, Alice will find a match, α_1, and knows that Bob chose the first message. Conversely, if Alice uses d_i^{-1}, $(i = 2, n)$ to compute $r_{1i} = \beta_1 * d_i^{-1}$, he will get $\alpha_i^{(*)} = (r_{1i} * d_i)^e \bmod N$, which is not equal to α_1. In other words, Alice cannot know the correct message T_1 that Bob chose. That is, once a pair, $(\alpha_i{}^{(*)}, \alpha_i)$, for example, has been matched, Alice knows that Bob chose the ith message. Hence, we can easily see that such explorations cost at most $n * k$ multiplications to obtain r_{ji}, and $n^2 * k$ multiplications and $n^2 * k$ exponentiations to yield all values of $\alpha_i^{(*)}$. Therefore, with at most $(n^2 * k + n * k)$ multiplications and $n^2 * k$ exponentiations, it is computationally feasible for Alice to decide which k values Bob selected, which violates Bob's privacy.

4. Proposed Protocol

In this section, we present our ID-based OT_k^n protocol based on bilinear pairings, which were proved and applied to cryptography by Boneh and Franklin in 2001 [31]. Our scheme consists of two phases: (1) an initialization phase and (2) an oblivious transfer phase. In the following, we first describe these two phases. Then, to demonstrate the chooser's privacy preservation, we use a misleading attack for an explanation. As the receiver's privacy preservation can be reasoned in a similar fashion, we omit its description here.

(1) Initialization Phase. In this phase, we adopt the same system parameters as the ones used in [31]. In addition, there also exists a trusted key generation center (KGC) which is assumed to be key-escrow-attack free. Initially, KGC chooses an additive group $G_1 = \langle P \rangle$ of order q, a multiplicative group $G_2 = \langle \hat{e}(P, P) \rangle$ of the same order, where $\hat{e}$ is a bilinear mapping, that is, $\hat{e} : G_1 \times G_1 \to G_2$, and three one-way hash functions: H: $\{0,1\}^* \to \{0,1\}^l$, $H_2 : G_1 \to \{0,1\}^l$, and H_1 which maps a string (a user's ID) to an element in G_1, that is, H_1:$\{0,1\}^* \to G_1$. Moreover, it selects $s \in Z_q^*$ as its private master key and computes the corresponding system public key as $P_{\mathrm{pub}} = sP$. Then, KGC publishes the system parameter set $\{G_1, G_2, q, \hat{e}, P, P_{\mathrm{pub}}, H, H_1, H_2\}$. After that, when a user U (sender/chooser) registers his identifier ID_U, KGC will compute a public/private key pair $U_{\mathrm{pub}}/U_{\mathrm{priv}}$ for him, where $U_{\mathrm{pub}} = H_1(ID_U)$ and $U_{\mathrm{priv}} = sU_{\mathrm{pub}}$.

(2) Oblivious Transfer Phase. In this phase, when a sender possessing n messages ($m_1, m_2, \ldots$, and m_n) wants to obliviously transfer k messages of them ($m_{\sigma_1}, m_{\sigma_2}, \ldots$, and m_{σ_k}) to a chooser, they together will execute the following steps, where the public/private key pairs of the sender and chooser are $S_{\mathrm{pub}}/S_{\mathrm{priv}}$ and $C_{\mathrm{pub}}/C_{\mathrm{priv}}$, respectively, and $\{\sigma_1, \sigma_2, \ldots, \sigma_k\} \subset \{1, 2, \ldots, n\}$ are the set of k choices selected by the chooser in advance. We also depict them in Table 1.

Step 1. The chooser randomly chooses two integers $a, b \in Z_q^*$ and computes $V = abC_{\mathrm{pub}}$, $V_j = bH(\sigma_j)C_{\mathrm{priv}}$, where $j = 1, 2, \ldots, k$ and V_j are the k random choices. After that, he generates a signature Sig on V by computing $h = H_2(V)$ and $\mathrm{Sig} = hC_{\mathrm{priv}}$. Then, he sends ID_c, V, $V_1, \ldots, V_k$ together with Sig to the sender.

Table 1: The proposed k-out-of-n authentic OT protocol.

Sender $(S_{\text{pub}}/S_{\text{priv}}(= sS_{\text{pub}}))$	Chooser $(C_{\text{pub}}/C_{\text{priv}}(= sC_{\text{pub}}))$
	(1) Selects , $b \in_R Z_q^*$, computes $V = abC_{\text{pub}}$, for $j = 1$ to k, computes $V_j = bH(\sigma_j)C_{\text{priv}}$, computes $h = H_2(V)$ and $\text{Sig} = hC_{\text{priv}}$.
$\xleftarrow{ID_C, V, V_1, \ldots, V_k, \text{Sig}}$	
(2) Computes $h = H_2(V)$ and verifies $\hat{e}(P, \text{Sig}) =? \hat{e}(P_{\text{pub}}, hC_{\text{pub}})$. If it does not hold, aborts. Selects $c \in_R Z_q^*$ and computes $U_j = cV_j$, for $j = 1, \ldots, k$, and $ct_i = m_i \oplus \hat{e}(H(i)V, S_{\text{priv}})^c$, for $i = 1, \ldots, n$.	
$\xrightarrow{U_1, \ldots, U_k, ct_1, \ldots, ct_n}$	
	(3) For $j = 1$ to k and $i = 1$ to n, computes $m_{\sigma_i} = ct_{\sigma_i} \oplus \hat{e}(U_j, S_{\text{pub}})^a$.

Step 2. After receiving IDc, V, $V_1, \ldots, V_k$ and Sig from the chooser, the sender computes $h = H_2(V)$ and verifies the chooser's signature by checking whether the equation $\hat{e}(P, \text{Sig}) = \hat{e}(P_{\text{pub}}, hC_{\text{pub}})$ holds. If it holds, he believes that the chooser is the intended party as claimed. Then, the sender randomly chooses an integer $c \in Z_q^*$ and computes $U_j = cV_j$ and $ct_i = m_i \oplus \hat{e}(H(i)V, S_{\text{priv}})^c$, where $j = 1, \ldots, k$, $i = 1, \ldots, n$, and m_i are the n messages. He/She then sends $U_1, \ldots, U_k, ct_1, \ldots,$ and ct_n to the chooser.

Step 3. After receiving the message $U_1, \ldots, U_k, ct_1, \ldots,$ and ct_n from the sender, the chooser can obtain the k intended messages by at most computing the equation, $m_{\sigma_j} = ct_{\sigma_j} \oplus \hat{e}(U_j, S_{\text{pub}})^a$, $nk - \cdot(k(k-1)/2)(= n + (n-1) + \cdots + (n - (k-1)))$ times.

(3) A Misleading Attack for Chooser's Privacy Preservation. To demonstrate the chooser's privacy more clearly, we take the following as a counterexample. According to step 1 in our protocol, the chooser computes $V_1, \ldots, V_k$, where $V_j = bH(\sigma_j)C_{\text{priv}}$ and $j = 1$ to k. Since b and C_{priv} are both the same for V_i and V_j, a misleading attack may be that $V_i/V_j = H(\sigma_i)/H(\sigma_j)$. A malicious sender can precompute $H(\sigma_i)/H(\sigma_j)$ for each i, j in the interval $[1, n]$. After receiving $V_1, \ldots, V_k$ from the chooser, he computes each V_i/V_j for all i, j in $[1, k]$ for a comparison with the precomputed values. Consequently, the sender may guess some or all of the chooser's choices. Therefore, the protocol cannot achieve chooser privacy. However, the mistake here is that both V_i and V_j are points in the additive group G_1. The division operation V_i/V_j is invalid because G_1 is an additive group.

5. Security Analysis

In this section, we use the following claims to show that our protocol not only is correct but also possesses the properties of mutual authentication, chooser's privacy, and sender's privacy and can resist against active attacks such as relay attack, man-in-the-middle attack, and denial of service attack.

Claim 1. The proposed protocol is correct.

Proof. After the protocol runs, the chooser can exactly obtain the k messages which he/she selected by computing

$$\begin{aligned} ct_{\sigma_j} \oplus \hat{e}\left(U_j, S_{\text{pub}}\right)^a &= ct_{\sigma_j} \oplus \hat{e}\left(cbH\left(\sigma_j\right)C_{\text{priv}}, S_{\text{pub}}\right)^a \\ &= ct_{\sigma_j} \oplus \hat{e}\left(H\left(\sigma_j\right)bcsC_{\text{pub}}, S_{\text{pub}}\right)^a \\ &= ct_{\sigma_j} \oplus \hat{e}\left(H\left(\sigma_j\right)abC_{\text{pub}}, sS_{\text{pub}}\right)^c \\ &= ct_{\sigma_j} \oplus \hat{e}\left(H\left(\sigma_j\right)V, S_{\text{priv}}\right)^c = m_{\sigma_j}. \end{aligned} \tag{4}$$

□

Claim 2. The proposed protocol can achieve mutual authentication.

Proof. We show the holdness of this claim by using the following two reasons.

(1) Apparently, it can be easily seen that the sender can authenticate the chooser by verifying the chooser's signature, Sig (as described in step 2 of the oblivious transfer phase).

(2) For that the ciphertext ct_i $(= m_i \oplus \hat{e}(H(i)V, S_{\text{priv}})^c)$ contains the sender's private key $S_{\text{priv}}(= sS_{\text{pub}})$, the chooser can compute the meaningful message m_{σ_j} only via using the sender's public key S_{pub} (also refer to the equation in claim 1). This means that only the true sender can produce the right ct_i and thus can be authenticated by the chooser using his public key.

□

Claim 3. The proposed protocol can achieve the chooser's privacy.

Proof. Due to the fact that each of the chooser's k choices $\sigma_j \in \{1, 2, \ldots, n\}$ are first hashed and randomized by H and b respectively, and then signed as $V_j = bH(\sigma_j)C_{\text{priv}}$ by chooser C in step 1, where b is a random number. We argue that nobody except for the chooser can know the choice σ_j. Because even an attacker might steal the chooser's private key C_{priv}, he/she cannot obtain $bH(\sigma_j)$ from V_j owing to the hardness of ECDLP. That is, he cannot figure out $bH(\sigma_j)$, and therefore not to mention σ_j. More formally, let $\mathcal{A} = \{(b, \sigma_j) \in Z_q * Z_n \mid bH(\sigma_j)C_{\text{priv}} = V_j\}$; that is, $\mathcal{A}$ consists of all the possible ordered pairs (b, σ_j) satisfying the equation $bH(\sigma_j)C_{\text{priv}} = V_j$. If we are given a value V_j, then under fixed C_{priv}, there only exists a unique value $bH(\sigma_j)$ satisfying the equation. And for a given $bH(\sigma_j)$, under the definition of a collision-free one-way hash function, once σ_j has been determined, the value of b is determined as well. That is, the relationship between b and σ_j is one-to-one. Having this observation in mind and the dimension of σ_j is n, we can see that there are n (b, σ_j) pairs in $\mathcal{A}$. In other words, $\Pr[\sigma_j \mid V_j] = \Pr[\sigma_j] = 1/n$ which means that, under seeing a specific V_j, the choice σ_j of the chooser cannot be revealed other than guessing. This achieves the *Shannon perfect secrecy*. Therefore, the proposed protocol possesses chooser's privacy. □

Claim 4. The proposed scheme can achieve the sender's privacy.

Proof. Assume that malicious chooser $\hat{C}$ wants to obtain more than k messages in the protocol. If he/she could succeed, then, the sender's privacy is violated (see Section 2.1). However, we will prove that, other than his k chosen messages, it is computationally infeasible for $\hat{C}$ to obtain the $(k+1)$th message by using the following two arguments, (I) and (II). In argument (I), we show why $\hat{C}$ must follow the protocol to form the values of V and $kV_j s$; otherwise, he/she cannot obtain the k chosen messages. In argument (II), we show that if $\hat{C}$ intends to obtain the $(k+1)$th message, he/she will face the intractable CTCDH problem under the assumption that $H(\cdot)$ is a random hash function. □

Argument (I). $\hat{C}$ must follow the protocol to form the values of $V(= ab\hat{C}_{\text{pub}})$ and $V_j(= bH(\sigma_j)\hat{C}_{\text{priv}})$, for $j = 1$ to k; otherwise, he cannot obtain the k chosen messages, $m_{\sigma_1}, \ldots, m_{\sigma_j}$.

In the following, we further divide this argument into three cases: (a) $\hat{C}$ fakes V but forms V_j honestly, (b) $\hat{C}$ fakes V_j but forms V honestly, and (c) $\hat{C}$ fakes both the values of V and V_j. (For each case's explanation, refer to Table 1.)

(a) $\hat{C}$ fakes V but forms V_j honestly. Assume that $\hat{C}$ is dishonest in forming V but forms V_j honestly as specified in the protocol. For example, without loss of generality, it replaces V with a specific $X \in G_1$ and computes $V_j = bH(\sigma_j)\hat{C}_{\text{priv}}$. Then, the sender will compute $U_j = cV_j$, $ct_i = m_i \oplus \hat{e}(H(i)X, S_{\text{priv}})^c$ and send them back to $\hat{C}$. As a result, $\hat{C}$ cannot decrypt $ct_{\sigma_j}(ct_{\sigma_j} = m_{\sigma_j} \oplus \hat{e}(U_j, S_{\text{pub}})^a)$ to obtain the k messages since $\hat{e}(U_j, S_{\text{pub}})^a$ is obviously not equal to $\hat{e}(H(\sigma_j)X, S_{\text{priv}})^c$ (refer to claim 1). Perhaps, for obtaining the k messages, $\hat{C}$ may try another way by computing $\hat{e}(H(i)X, S_{\text{priv}})^c$ expected to be equal to $\hat{e}(U_j, S_{\text{pub}})^a$. But this is computationally infeasible since $\hat{C}$ does not know both the sender's private key S_{priv} and the one-time secrecy c. To extract c from U_j is an ECDLP.

(b) $\hat{C}$ fakes $V_j s$ but forms V honestly. Assume that $\hat{C}$ is dishonest in forming $V_j s$ but forms V in the same manner as specified in the protocol. For example, without loss of generality, he replaces each V_j with a specified $X_j \in G_1$ and computes $V = ab\hat{C}_{\text{pub}}$. Then, the sender will compute $U_j = cV_j = cX_j$, $ct_i = m_i \oplus \hat{e}(H(i)V, S_{\text{priv}})^c = m_i \oplus \hat{e}(H(i)ab\hat{C}_{\text{pub}}, S_{\text{priv}})^c$, for $i = 1$ to n, and send them back to $\hat{C}$. As a result, $\hat{C}$ cannot decrypt ct_{σ_j} since $\hat{e}(U_j, S_{\text{pub}})^a = \hat{e}(cX_j, S_{\text{pub}})^a$ is obviously not equal to $\hat{e}(H(i)V, S_{\text{priv}})^c$. Perhaps, for obtaining the k messages, $\hat{C}$ may try another way by computing $\hat{e}(H(i)V, S_{\text{priv}})^c$ $(= \hat{e}(H(i)ab\hat{C}_{\text{pub}}, S_{\text{priv}})^c)$ expected to be equal to $(U_j, S_{\text{pub}})^a$. But again this is computationally infeasible since $\hat{C}$ does not know both the sender's private key S_{priv} and the one-time secrecy c. Even he knows S_{priv}, extracting c from $U_j(= cX_j)$ is an ECDLP. Hence, $\hat{C}$ cannot compute the value $\hat{e}(H(i)V, S_{\text{priv}})^c$ to decrypt ct_{σ_j} for obtaining the k messages, m_{σ_j}.

(c) $\hat{C}$ fakes both the values of V and V_j. Without loss of generality, we assume that $\hat{C}$ replaces V with X and also fakes V_j as $H(\sigma_j)X$. Under this construction, the value of U_j computed by the sender would be $U_j = cV_j = cH(\sigma_j)X$ and the ciphertexts ct_{σ_j} would be $m_{\sigma_j} \oplus \hat{e}(H(\sigma_j)X, S_{\text{priv}})^c$, for $j = 1$ to k, or equivalently, $ct_{\sigma_j} = m_{\sigma_j} \oplus \hat{e}(cH(\sigma_j)X, S_{\text{priv}})$. Although, $\hat{C}$ knows the value of $cH(\sigma_j)X$ (since it just equals to U_j received from the sender), it still cannot compute $\hat{e}(cH(\sigma_j)X, S_{\text{priv}})$ without the knowledge of S_{priv}. From above description, we know that when the setting of V is X and V_j is $H(\sigma_j)X$, $\hat{C}$ cannot obtain m_{σ_j}. Not to mention, $\hat{C}$ might set V_j as $H(\sigma_j)Y$, where $Y(\neq X)$ is a random chosen element in G_1. In summary, $\hat{C}$ cannot obtain the k selected messages under the violation of setting both the values, V and V_j.

Argument (II). If $\hat{C}$ follows the protocol honestly to obtain k messages, but intends to extract the $(k+1)$th message then it will face the intractable CTCDH problem under the assumption that $H(\cdot)$ is a random hash function.

That $\hat{C}$ wants to obtain message m_i implies $\hat{C}$ would have the knowledge of $\hat{e}(H(i)V,\ S_{\text{priv}})^c$ $(= \hat{e}(U_j,\ S_{\text{pub}})^a)$ (in fact, according to argument (I), an honest chooser C could know k of the n values, $\hat{e}(H(i)V,\ S_{\text{priv}})^c$, for $i = 1$ to n, since $\hat{e}(H(i)V,\ S_{\text{priv}})^c = \hat{e}(U_j,\ S_{\text{pub}})^a$, for $i = \sigma_j$ and $j = 1$ to k). Let $y^{(i)} \in G_2$ and $\hat{e}(H(i)V,\ S_{\text{priv}})^c = y^{(i)}$. According to argument (I), for obtaining the k chosen messages, $\hat{C}$ cannot change the structures of $V(= abC_{\text{pub}})$ and V_j $(= bH(\sigma_j)\hat{C}_{\text{priv}})$. Under this situation, $y^{(i)}$ only can be decomposed as $y^{(i)} = \hat{e}(H(i)ab\hat{C}_{\text{pub}},\ S_{\text{priv}})^c = \hat{e}(abH(i)\hat{C}_{\text{priv}},\ S_{\text{pub}})^c$ since $S_{\text{priv}} = sS_{\text{pub}}$ and $\hat{C}_{\text{priv}} = s\hat{C}_{\text{pub}}$. Moreover, under the assumption that $H(\cdot)$ is a random hash function and the fact that $\hat{C}$ has the knowledge of a, b, $\hat{C}_{\text{priv}}$, and S_{pub}, $y^{(i)}$ can be represented as $(g_i)^c$, where g_i equals to $\hat{e}(abH(i)\hat{C}_{\text{priv}},\ S_{\text{pub}})$ and is a random element in G_2 due to the assumption that $H(\cdot)$ is a random hash function. Consequently, the problem $\hat{C}$ really faces is finding the $(k+1)$th pair $(\sigma_{k+1},\ (g_{\sigma_{k+1}})^c)$ with the knowledge of k pairs of $(\sigma_1,\ (g_{\sigma_1})^c), (\sigma_2,\ (g_{\sigma_2})^c), \ldots,$ and $(\sigma_k,\ (g_{\sigma_k})^c)$, where $(g_{\sigma_j})^c = \hat{e}(U_j,\ S_{\text{pub}})^a$, but without the knowledge of sender's one-time secrecy c (since it is an ECDLP for extracting c from $U_j(= cV_j)$). This is known as the intractable CTCDH problem introduced in Section 2.3 by letting $k = (l-1)$. Therefore, the chooser cannot obtain the $(k+1)$th message.

According to arguments I and II, we have proven claim 4 that our scheme has the sender's privacy.

Claim 5. The proposed scheme can resist against replay attack.

Proof. Suppose that an adversary intercepts a chooser's OT request (containing ID_C, V, V_j, and Sig) and replays it later. After receiving the sender's new response $(U_1, \ldots, U_k, ct_1, \ldots, ct_n)$ computed from the replayed V and V_j, the adversary cannot obtain the k selected messages by computing $m_{\sigma_j} = ct_{\sigma_j} \oplus \hat{e}(U_j,\ S_{\text{pub}})^a$ since he/she does not know the value of a embedded in the replayed message V. It is computationally infeasible for the adversary to extract a from $V = abC_{\text{pub}}$, due to the hardness of ECDLP. □

Claim 6. The proposed scheme can resist against man-in-the-middle attack (MIMA).

Proof. MIMA is an attack that an adversary E intercepts the communication line between two communicating parties and uses some means to make them believe that they each are talking to the intended party as claimed. But indeed, they are talking to E. Figure 1 illustrates the scenario of such a MIMA. We first argue that the adversary E cannot succeed in this scenario since it cannot generate the valid message (2), $(\text{ID}_C,\ V', V'_1, \ldots,\ V'_k, \text{Sig}')$ as shown in the figure. More clearly, without the knowledge of chooser's private key C_{priv}, he/she cannot forge a valid signature Sig′ in message (2) to be successfully verified by the sender since Sig′ should be equal to $H_2(V)\ C_{\text{priv}}$. In addition, it is also hard for E to forge valid message (4), $(U'_1, \ldots,\ U'_k,\ ct'_1, \ldots,\ ct'_n)$, to be accepted by the chooser. Since that for embedding a meaningful m'_i into ct'_i, E must have the knowledge of $\hat{e}(H(i)V,\ S_{\text{priv}})^c$. Although E can choose another random nonce c' such that $U'_j = c'V_j$, it still has to know the sender's private key S_{priv} to form the valid $ct'_i (= m_i \oplus \hat{e}(H(i)V, S_{\text{priv}})^{c'})$. Therefore, without the knowledge of S_{priv}, E cannot launch such a MIMA attack. □

Claim 7. The proposed scheme can resist a denial of service attack (DOS).

Proof. Our protocol has a built-in mutual authentication property; thus, it can prevent this kind of attack, as the sender needs only one hash and two bilinear pairing computations to authenticate the chooser in step (2). Once the sender finds that the authenticating equation $\hat{e}(P, \text{Sig}) = \hat{e}(P_{\text{pub}},\ hC_{\text{pub}})$ does not hold, it aborts the procedure. □

5.1. Communicational Cost Comparisons. Generally, the communicational cost of a protocol run consists of three factors: (1) needed passes, (2) computational overhead, and (3) needed transmission data size (NTDS) or bandwidth consumption. It is well known that factor (1) is always dominant over factor (2). Hence, in this section, we focus only on factor (1) and (3) to demonstrate the communication cost comparisons among our nonadaptive OT_k^n protocol and the other same type OT_k^n protocols, such as Chu and Tzeng's [16] (which is to our best knowledge, the most efficient OT_k^n scheme up to date), Mu et al.'s [13], Naor and Pinkas's [7], and recent works [17, 18, 24, 27]. From factor (1), our scheme is the most efficient since it only requires two passes. As to factor (3), the data size transmitted in our scheme is also the minimal among such type of OT_k^n schemes. For demonstrating this in the following, we will first describe two underlying facts and used notations for making comparisons about factor (3).

Generally speaking, we have the following two facts for cryptosystems.

Fact 1. To the same security level, a RSA cryptosystem would require a key length of 1024 bits while an ECC-based cryptosystem only needs 160 bits.

Fact 2. The length of the ciphertexts for RSA, ElGamal, and ECC-based cryptosystems is 1024 bits, 1024 bits, and 160 bits, correspondingly.

Notations. We use |string/action| to represent the bit length of a *string*, or the required bit length that an *action* performs.

After the description of used facts and notations, we now use them to estimate the needed transmission data size (NTDS) of our scheme and the above-mentioned OT_k^n protocols. In our scheme, each of the variables V, $V_1, \ldots, V_k$, Sig, $U_1, \ldots, U_k$ transmitted between the chooser and sender is an ECC point. Thus, the NTDS from the chooser to the sender is estimated as $160 * (k+2)$ bits and from the sender to the chooser is $160\,k + n * |\text{ciphertext}|$ bits. Naor and Pinkas's scheme [7] constructs their OT_k^n scheme by evoking an OT_1^2 primitive $\log n$ times. Thus, the needed number of passes is $\log n$ times the number of passes required in one of their OT_1^2's protocol run and

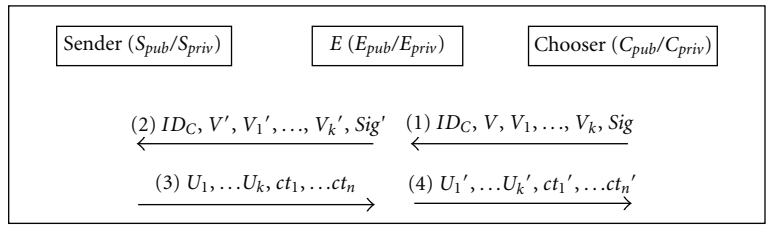

Figure 1: The scenario of MIMA attack.

Table 2: Needed rounds and data size comparisons among OT_k^n protocols.

Protocol	Passes	Size of message: $C \rightarrow S$ (bits)	Size of message: $S \rightarrow C$ (bits)	Mutual authentication
Ours	2	$160 * (k+2)$	$160k + n * \lvert\text{ciphertext}\rvert$	Yes
Naor and Pinkas [7]	$k* \log n$ OT_1^2	depends on OT_1^2	depends on OT_1^2	No
Mu et al.'s scheme (1) [13]	3	$1024k$	$1024n + nk * \lvert\text{ciphertext}\rvert$	No
Mu et al.'s scheme (2) [13]	2	$1024 * 2n$	$n * \lvert\text{ciphertext}\rvert$	No
Chu and Tzeng [16]	2	$1024k$	$1024 * (k + 1) + n * \lvert\text{ciphertext}\rvert$	No
Zhang and Wang [17]	2	$1024 * (k+3)$	$1024n + n * \lvert\text{ciphertext}\rvert$	No
Huang and Chang [18]	3	$1024k$	$(n + k) * \lvert\text{ciphertext}\rvert$	
Camenisch et al. [24]	$2 + k * \text{Pok}$	$\lvert\text{Pok}\rvert + k * \lvert\text{BlindExtract}\rvert$	$n * \lvert\text{ciphertext}\rvert + \lvert\text{Pok}\rvert + k * \lvert\text{BlindExtract}\rvert$	No
Chang and Lee [27]	4	$1024k$	$(n + 2k + 2)* 1024$	No
Ma et al. [28]	$k * (2 + \text{Pok})$	$k * 3\lvert\text{ciphertext}\rvert$	$k * (n * 2\lvert\text{ciphertext}\rvert)$	No

likewise the NTDS is about $\log n$ times of the NTDS that an OT_1^2's work demands. Therefore, their scheme has the most expensive communicational cost. As for Camenisch et al.'s protocol [24], the communicational cost is expensive as well due to the complexity of the protocol. In their protocol, the sender first sends n commitments to the chooser, and then the sender and the chooser together run a proof-of-knowledge (Pok) subprotocol for assuring the correctness of the commitments. If the proof is valid, the sender sends n ciphertexts to the chooser, and the chooser then runs the BlindExtract subprotocol k times with the help of the sender to extract the blind choices to decrypt the ciphertexts.

Consequently, the number of passes for executing protocol [24] is $2 + k * \text{Pok}$, where Pok represents the required passes for executing the proof-of-knowledge subprotocol. Besides, the NTDS from chooser to sender is estimated as $\lvert\text{Pok}\rvert + k * \lvert\text{BlindExtract}\rvert$ and from sender to chooser is $n * \lvert\text{ciphertext}\rvert + \lvert\text{Pok}\rvert + k * \lvert\text{BlindExtract}\rvert$. Similarly, the passes and NTDS of other studies can be estimated in the same manner. We show the comparison results in Table 2.

From Table 2, we can see that our scheme not only possesses the mutual authentication function but also is the most efficient in both needed passes and NTDS among these related. Therefore, our scheme can be gracefully used when applied in commercial applications (e.g., Kerschbaum et al.'s method [1] used OT scheme as a building block in constructing RFID benchmarking protocols).

6. Conclusion

An OT scheme which is secure and efficient in communicational cost is essential and eager for commercial applications. After reviewing most of the OT schemes, we found that, other than considering the protocol's correctness and privacy of both communication parties, almost all of them lack the security services, such as mutual authentication, and the prevention of replay, DOS, and main-in-the-middle attacks. Hence, they should run under a secure channel when applied in commercial applications. This will increase execution overhead. Therefore, to get rid of using the secure channel (for improving the communicational efficiency in some applications, such as mental poker playing, oblivious key searching), we propose a novel k-out-of-n oblivious transfer protocol by combining an OT scheme with a security mechanism based on bilinear pairing. We have proved that our scheme not only is correct but also possesses the properties of mutual authentication, the sender's privacy, and the chooser's privacy and can resist against replay and MIMA attacks. Further, we have compared our scheme with other nonadaptive k-out-of-n OT schemes in the aspects of needed passes, NTDS, and the function of mutual authentication and shown the result in Table 2. From Table 2, we can see that our scheme is the most efficient in communicational cost (including needed passes and NTDS). In addition, to our knowledge, it is the only OT_k^n scheme that has successfully integrated the function of mutual authentication nowadays.

References

[1] F. Kerschbaum, N. Oertel, and L. W. F. Chaves, "Privacy-preserving computation of benchmarks on item-level data using RFID," in *Proceedings of the 3rd ACM Conference on Wireless Network Security (WiSec '10)*, pp. 105–110, March 2010.

[2] M. O. Rabin, "How to exchange secrets with oblivious transfer," Tech. Rep. TR-81, Aiken Computation Lab, Harvard University, Cambridge, Mass, USA, 1981.

[3] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Communications of the ACM*, vol. 28, no. 6, pp. 637–647, 1985.

[4] G. Brassard, C. Crepeau, and J.-M. Robert, "All-or-nothing disclosure of secrets," in *Proceedings of the International Conference on Advances in Cryptology (CRYPTO '86)*, vol. 263 of *Lecture Notes in Computer Science*, pp. 234–238, 1986.

[5] J. S. Chou and Y. S. Yeh, "Mental poker game based on a bit commitment scheme through network," *Computer Networks*, vol. 38, no. 2, pp. 247–255, 2002.

[6] M. Bellare and S. Micali, "Non-interactive oblivious transfer and application," in *Proceedings of the International Conference on Advances in Cryptology (CRYPTO '89)*, vol. 435 of *Lecture Notes in Computer Science*, pp. 547–557, 1989.

[7] M. Naor and B. Pinkas, "Oblivious transfer with adaptive queries," in *Proceedings of the International Conference on Advances in Cryptology (CRYPTO '99)*, Lecture Notes in Computer Science, pp. 573–590, 1999.

[8] M. Naor, B. Pinkas, and R. Sumner, "Privacy preserving auctions and mechanism design," in *Proceedings of the 1st ACM Conference on Electronic Commerce*, 1999.

[9] M. Naor and B. Pinkas, "Distributed oblivious transfer," in *Proceedings of the International Conference on Advances in Cryptology (CRYPTO '00)*, vol. 1976 of *Lecture Notes in Computer Science*, 2000.

[10] M. Naor and B. Pinkast, "Oblivious transfer and polynomial evaluation," in *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (FCRC '99)*, pp. 245–254, May 1999.

[11] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," in *Proceedings of the 12th annual ACM-SIAM symposium on Discret Mathematics (SODA '01)*, pp. 448–457, 2001.

[12] H. Ghodosi, "On insecurity of Naor-Pinkas' distributed oblivious transfer," *Information Processing Letters*, vol. 104, no. 5, pp. 179–182, 2007.

[13] Y. Mu, J. Zhang, and V. Varadharajan, "m out of n oblivious transfer," in *Proceedings of the 7th Australasian Conference on Information Security and Privacy (ACISP '02)*, vol. 2384 of *Lecture Notes in Computer Science*, pp. 395–405, 2002.

[14] H. Ghodosi and R. Zaare-Nahandi, "Comments on the 'm out of n oblivious transfer'," *Information Processing Letters*, vol. 97, no. 4, pp. 153–155, 2006.

[15] W. Ogata and K. Kurosawa, "Oblivious keyword search," *Journal of Complexity*, vol. 20, no. 2-3, pp. 356–371, 2004.

[16] C. K. Chu and W. G. Tzeng, "Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries," in *Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC '05)*, pp. 172–183, January 2005.

[17] J. Zhang and Y. Wang, "Two provably secure k-out-of-n oblivious transfer schemes," *Applied Mathematics and Computation*, vol. 169, no. 2, pp. 1211–1220, 2005.

[18] H. F. Huang and C. C. Chang, "A new design for efficient t-out-n oblivious transfer scheme," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA '05)*, pp. 28–30, March 2005.

[19] A. Parakh, "Oblivious transfer using elliptic curves," in *Proceedings of the 15th International Conference on Computing (CIC '06)*, pp. 323–328, November 2006.

[20] S. Kim and G. Lee, "Secure verifiable non-interactive oblivious transfer protocol using RSA and Bit commitment on distributed environment," *Future Generation Computer Systems*, vol. 25, no. 3, pp. 352–357, 2009.

[21] Y. F. Chang and W. C. Shiao, "The essential design principles of verifiable non-interactive OT protocols," in *Proceedings of the 8th International Conference on Intelligent Systems Design and Applications (ISDA '08)*, pp. 241–245, November 2008.

[22] L. M. Kohnfelder, "On the signature reblocking problem in public-key cryptography," *Communications of the ACM*, vol. 21, no. 2, p. 179, 1978.

[23] S. Halevi and Y. T. Kalai, "Smooth projective hashing and two-message oblivious transfer," Cryptology ePrint Archive 2007/118, 2007.

[24] J. Camenisch, G. Neven, and A. Shelat, "Simulatable adaptive oblivious transfer," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 4515 of *Lecture Notes in Computer Science*, pp. 573–590, 2007.

[25] M. Green and S. Hohenberger, "Blind identity-based encryption and simulatable oblivious transfer," Cryptology ePrint Archive 2007/235, 2007.

[26] J. Qin, H. W. Zhao, and M. Q. Wang, "Non-interactive oblivious transfer protocols," in *Proceedings of the International Forum on Information Technology and Applications (IFITA '09)*, pp. 120–124, May 2009.

[27] C. C. Chang and J. S. Lee, "Robust t-out-of-n oblivious transfer mechanism based on CRT," *Journal of Network and Computer Applications*, vol. 32, no. 1, pp. 226–235, 2009.

[28] X. Ma, L. Xu, and F. Zhang, "Oblivious transfer with timed-release receiver's privacy," *Journal of Systems and Software*, vol. 84, no. 3, pp. 460–464, 2011.

[29] W. Stallings, *Cryptography and Network Security—Principals and Practices*, Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 2003.

[30] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proceedings of the 40th annual ACM symposium on Theory of Computing (STOC '82)*, pp. 365–377, 1982.

[31] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," in *Proceedings of the International Conference on Advances in Cryptology (CRYPTO '01)*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229, 2001.

[32] D. R. Stinson, *Cryptography—Theory & Practice*, Chapman & Hall/CRC Taylor & Francis Group, 3rd edition, 2006.

[33] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-group signature scheme," in *Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography*, vol. 2567 of *Lecture Notes in Computer Science*, pp. 31–46, 2003.

[34] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko, "The one-more-RSA-inversion problems and the security of chaum's blind signature scheme," in *Proceedings of Financial Cryptography (FC '01)*, vol. 2248 of *Lecture Notes in Computer Science*, pp. 319–338, 2003.