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

Jue-Sam Chou*[1], Cheng-Lun Wu [2], Yalin Chen [3]

[1] Department of Information Management, Nanhua University, Taiwan R.O.C
*: corresponding author: jschou@mail.nhu.edu.tw
Tel: 886+ (0)5+272-1001 ext.56536

[2] Department of Information Management, Nanhua University, Taiwan R.O.C
wfdawu@gmail.com

[3] Institute of information systems and applications, National Tsing Hua University
d949702@oz.nthu.edu.tw

Abstract

As traditional oblivious transfer protocols are treated as cryptographic primitives in most cases, they are usually executed without the consideration of possible attacks, e.g., impersonation, replaying, and man-in-the-middle attacks. Therefore, when these protocols are applied in certain applications, such as mental poker game playing and fairly contracts signing, some extra mechanisms must be combined to ensure its security. However, after the combination, we found that almost all of the resulting schemes are not efficient enough in communicational cost, which is a significant concern for all commercial transactions. Inspired by this observation, we propose a novel secure oblivious transfer protocol based on bilinear pairing which not only can provide mutual authentication to resist malicious attacks but also is efficient in communicational cost.

Keywords: *oblivious transfer, mutual authentication, ID-based cryptosystem, impersonation, bilinear pairing*

1. Introduction

Oblivious transfer (OT) is an important tool for designing cryptographic primitives and has been widely used in various applications like fairly contracts signing, obliviously database searching, mental poker games playing, privacy-preserving auctions, secure multiparty computations, and so on. In 1981, Rabin [1] first proposed an interactive OT scheme in which the probability for the receiver to be able 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, Goldreich, and Lempel [2] presented a more generalized form of OT, naming 1-out-of-2 OT (OT_1^2) which can let

a sender send two encrypted messages to a chooser, whereas the chooser can decrypt only one of them that he had chosen in advance. In addition, they also presented a contract-signing protocol by evoking OT_1^2 multiple times to achieve the goal that one party cannot obtain the other party's contract signature without first showing his own. In 1986, Brassard and Crépeau [3] further extended OT_1^2 to 1-out-of-*n* OT (OT_1^n, also known as "all-or-nothing"), the case of sending *n* messages to a chooser with only one of them can be obtained by the chooser. They pointed out that their OT_1^n scheme can be used to implement a multi-party mental poker game [31] against players' coalition. Except for the above interactive versions, Bellare and Micali [4] first proposed a non-interactive OT_1^2 scheme in 1989. In the scheme, a user can obliviously transfer messages to another party whom is equipped with two public keys.

During 1999 to 2001, based on the above-mentioned interactive and non-interactive OT schemes, Naor et al. proposed some related OT works such as, adaptive OT_k^n [5], proxy OT_1^2 [6], distributed OT_k^n [7], efficient OT_1^n [8], and efficient OT_k^n [9]. Here, OT_k^n scheme is the final form of OT schemes. In it, from *n* encrypted messages sent by the sender, the chooser can obtain *k* of them which he had chosen without the sender's knowledge about which part of the messages can be decrypted by him. In Naor et al.'s distributed schemes [7], the sender distributes her two messages (M_0, M_1) among *n* servers and the chooser contacts with *k* ($k<n$) servers to get one and only one (M_σ, σ = 0 or 1) of these messages. They claimed that their schemes can protect the privacy for both parties. However in 2007, Ghodosi [27] showed two attacks on their schemes. One is that two collaborating servers could reveal the chooser's choice σ, and the other is that by only collating with one server, the chooser could learn both M_0 and M_1. In 2002, Mu et al. [10] proposed three *k*-out-of-*n* OT schemes constructed from RSA encryption, Nyberg-Rueppel signature, and ElGamal encryption scheme, respectively. Two of them are interactive and the other can be either interactive or non-interactive. They claimed that their schemes are complete, robust, and flexible, and can induce a significant improvement in communicational cost. However in 2006, Ghodosi et al. [28] showed that their schemes fail to satisfy the requirement of the oblivious transfer. In 2004, Ogata and Kurosawa [11] based on RSA blind signature proposed another OT_k^n scheme which can be employed in either an adaptive or non-adaptive manner,. They claimed that their scheme can be applied in oblivious key searching. After that, in 2005, Chu et al.

also proposed three OT_k^n schemes [12-14]. Among them, [12] is the most efficient because it needs only 2 passes to send $1024k$ bits from the chooser to the sender and $1024*(k+1)+n*|\text{Data}|$ bits from the sender to the chooser, where Data is a message or a ciphertext and |Data| represents the bits length of Data. In 2006, Parakh [15] proposed an elliptic-curve based algorithm allowing *A* to obliviously transfer his secrecy n_A to *B* with one-half success probability. However, we found that *A* can decide whether *B* can obtain his secret n_A (one-to-one mapping to P_{n_A}) by first assuming that $P_A = P_B$. Under this assumption, on receiving $\{\boldsymbol{n_B P_B}; \boldsymbol{n_B(n_A P_A)+R}; \boldsymbol{n_B R}\}$ from *B*, *A* can obtain *B*'s one-time random variable *R* by computing $(\boldsymbol{n_B(n_A P_A)+R}) - n_A(\boldsymbol{n_B P_B})$. Then, *A* can obtain $\boldsymbol{n_B K}$ by computing $n_A(\boldsymbol{n_B R}) = n_B(n_A R)$. Finally, he can obtain Z_B by computing $(n_A(\boldsymbol{n_B R}) + P_{n_A}) - n_B K$, as *B* does in step 5(b). Therefore, if *A* can confirm that $Z_B = P_{n_A}$, *A* knows that *B* can obtain n_A after the protocol run; otherwise *B* cann't obtain the value. This violates *B*'s privacy. In the same year, for coping with all possible attacks encountered in an open network, Kim et al. [16] proposed two OT_1^2 protocols, which are modified from Bellare-Micali non-interactive OT_1^2 scheme [4] by appending the sender's signature to make the sender undeniable about what he had sent and be authentic to the chooser. However, we found that other than the weaknesses pointed by Chang et al. [25], Kim et al.'s protocol still has a the reblocking problem [23] 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 [17] proposed another OT_1^2 scheme by using smooth projective hashing and showed that the used RSA-composite in their scheme needn't be a product of safe primes. Also in 2007, Camenish et al. and Green et al. proposed two related OT schemes [18, 19] respectively. Both focus on the security of full simulatability for the sender and receiver to resist against the selective-failure attack [5]. In 2009, Qin et al. [29] proposed two non-interactive 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 when he wants to select one of the *n* messages sent by the sender,. This makes their scheme somewhat inconvenient and inconsistent with the meaning of non-interactive protocols as indicated in the title. (This phenomenon can also be found in other proposed non-interactive OT schemes as well.) Also in 2009, Chang et al. [20] presented a robust OT_k^n scheme using both the RSA blind signature and Chinese Remainder Theorem. However, we found that their scheme fails since the sender Alice can decide which parts of the sent messages were chosen by the chooser Bob. We will describe this weakness in Section 3.2.

After surveying all of the above-mentioned OT schemes, we found that almost all

of them lack the consideration of adding security features. Only [1] and [16] do contemplate the protection against all possible attacks. However, study [16] fails. Hence, if we wish all of the proposed OT protocols, other than scheme [1], 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, and 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 of our scheme and the communicational cost comparisons among related works 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 paring in Section 2.2, and some intractable problems used in this article in Section 2.3.

2.1 Security features of our OT_k^n scheme

As in a traditional OT scheme, our OT_k^n also has two parties, the sender *S* and the chooser *C*. In it, *S* obliviously transfers *n* messages to *C*, and *C* can choose *k* messages among them without *S*'s knowledge about which k messages are selected, where $n \geq 2$ and $k < n$. In addition, our scheme possesses the following three security features as does in a traditional OT scheme:

(1) ***Correctness***: Eventually after the protocol run, *C* should obtain the valid data which he had chosen.

(2) ***Chooser's privacy***: In the protocol, each of the k choices (chosen by the chooser) should not be known to the sender or any other third party. More precisely, each of the chooser's encrypted choice can be any valid choice with equal probability, i.e. for an encrypted choice *y* and any

valid choice x, $\Pr[x|y] = \Pr[x]$. 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 that he did not choose. More formally, the ciphertexts sent by the sender are semantically secure [30]. The chooser can obtain a plaintext from its ciphertext only if he has the key offered by the sender to decrypt the ciphertext.

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

(4) ***Impersonation attack resistance***: Each party has to authenticate the counterpart as the intended party. 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 that a sender sent to a chooser before.

(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 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 paring

Let G_1 and G_2 be two groups of order q, where q is a large prime, G_1 be a subgroup of an additive group composed of points on an elliptic curve $\boldsymbol{E}/\boldsymbol{F}_p$, and G_2 be a subgroup of a multiplicative group with elements in a finite field $\boldsymbol{F}_{p^2}^*$. A bilinear mapping is defined as $\hat{e} : G_1 \times G_1 \to G_2$. The mapping must satisfy the following properties:

(1) Bilinear: A mapping $\hat{e} : G_1 \times G_1 \to G_2$ 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) Non-degenerate: 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 Diffie-Hellman problems

Let $a, b, c \in_R Z_q^*$, and the three groups, $G =< g >$, $G_1 =< P >$, and $G_2 =< g (=\hat{e}(P, P)) >$, each be a group of prime order q. In the following, we describe some well known intractable Diffie-Hellman problems that will be used in this paper.

(1) **The Computational Diffie-Hellman (CDH) problem**: CDH problem is that in G, given (g, g^a, g^b), finding the element $C = g^{ab}$.

(2) **The Decisional Diffie-Hellman (DDH) problem**: DDH problem is that in G, given (g, g^a, g^b, g^c), deciding whether $c=ab$.

(3) **The Bilinear Computational Diffie-Hellman (BCDH) problem**: BCDH problem is that given (P, aP, bP, cP) in G_1, finding $\hat{e}(P, P)^{abc}$ in G_2.

According to Boneh and Frank's study [23], 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, $T(\cdot)$ be a target oracle which returns a random element in G, and $(\cdot)^c$ a helper oracle, where c is a random integer from 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 that finding l pairs of $(j_1, v_1), \ldots,$ and (j_l, v_l), with each satisfying $v_i = (T(j_i))^c$, for $1 \le i \le l$ and $q_h < l \le q_t$. 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 $(j_1, v_1), \ldots, (j_{l-1}, v_{l-1})$ via querying the $T(\cdot)$ oracle and the helper oracle $(\cdot)^c$ correspondingly, trying to find the l^{th} pair (j_l, v_l). The CTCDH problem is proposed and considered as a hard problem by Boldyreva in 2002 [21]. Its former version in RSA is proved by Bellare et al. in [22].

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 requirements of general OT_k^n schemes. However, we

found that their scheme can not satisfy the chooser's privacy. In the following, we first review the scheme in Section 3.1 then show the weakness in Section 3.2.

3.1 Review

We roughly describe the protocol by listing the relevant steps in the following (see [20] for more details).

Step 1: After receiving the request sent by Bob for all messages a_1, a_2, …, a_n, Alice owning these n messages selects n relatively prime integers, d_1, d_2, …, d_n, and computes $D = d_1 * d_2 * \ldots * d_n$. He then constructs the congruence system

$$C \equiv a_1 \bmod d_1,\ C \equiv a_2 \bmod d_2, \ldots,\ C \equiv a_n \bmod d_n$$

Furthermore, Alice computes the following values:

$$T_1 = d_1^e \bmod N,\quad T_2 = d_2^e \bmod N,\ \ldots,\ T_n = d_n^e \bmod N,$$

by using her public key e. Finally, Alice 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 the k messages among the information possessed by Alice, then Bob must select k pairs of (ID_j', T_j'), for j = 1 to k, from the public board and generate k corresponding random numbers r_1, r_2, …, r_k, for each pair of (ID_j', T_j') first. Bob subsequently computes the following:

$$\alpha_1 = r_1^e * T_1' \bmod N,\quad \alpha_2 = r_2^e * T_2' \bmod N,\ \ldots,\quad \alpha_k = r_k^e * T_k' \bmod N,$$

by using Alice's public key e and then 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 $\beta_1 = \alpha_1^d = r_1 T_1'^d = r_1 d_1' \bmod N$, $\beta_2 = \alpha_2^d = r_2 T_2'^d = r_2 d_2' \bmod N$, …, $\beta_k = \alpha_k^d = r_k T_k'^d = r_k d_k' \bmod N$, and then sends the results $\{\beta_1, \beta_2, \ldots, \beta_k\}$ to Bob.

Step 4: After receiving the messages sent by 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 $b_1 = C \bmod d_1'$, $b_2 = C \bmod d_2'$, …, $b_k = C \bmod d_k'$.

3.2 Weaknesses

Although Chang et al. claimed that their scheme can achieve the requirements an OT_k^n scheme needs, 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 (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. And by 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 want to guess out which T_i Bob chosen, Alice start 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 Bob chooses the first message. Conversely, if Alice uses d_i^{-1}, (i=2, n), to computer $r_{1i} = \beta_1 * d_i^{-1}$, he will get $\alpha_i(*) = (r_{1i} * d_i)$ e mod N which is not equal to $\alpha 1$. In other words, Alice cannot know the right message T1 Bob chosen. That is, once a pair, says ($\alpha_i^{(*)}$, α_j), has matched, Alice knows that Bob has chosen the ith message. Hence, we can easily see that such exploration at most needs $n*k$ multiplications to obtain r_{ji}, n^2*k multiplications and n^2*k exponentiations to yield all $\alpha_i^{(*)}$. Therefore, totally with at most ($n^2*k + n*k$) multiplications and n^2*k exponentiations which is computationally feasible, Alice can decide which k values Bob selected. This violates Bob's privacy.

4. Proposed protocol

For the simplicity in key distribution and management, an ID-based public key cryptosystem is often suggested for the authentication of user's identity. In this section, we present our ID-based OT_k^n protocol based on bilinear parings which were proved and applied to cryptography by Boneh and Franklin in 2001 [24]. Our scheme consists of two phases: (1) initialization phase, and (2) oblivious transfer phase. In the following, we describe these two phases. Then for demonstrate its chooser's privacy preservation, we take a counter example. The other case for receiver's privacy can be reasoned in a similar fashion, we omit it here.

(1) Initialization phase

In this phase, we adopt the same system parameters as the ones used in [24]. In addition, there also exists a key generation center (KGC) who initially chooses an additive group $G_1 = < P >$ of order q, a multiplicative group $G_2 = < \hat{e}(P, P) >$ of the same order, where $\hat{e}$ is a bilinear mapping, i.e. $\hat{e}: G_1 \times G_1 \to G_2$, 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, i.e. $H_1: \{0,1\}^* \to G_1$. Moreover, KGC selects $s \in Z_q^*$ as its private master key and computes the corresponding system public key as $P_{pub} = sP$. Then, KGC publishes the system parameter set $\{G_1, G_2, q, \hat{e}, P, P_{pub}, H, H_1, H_2\}$.

After that, when a user U (sender/chooser) registers his identifier ID_U to KGC, KGC will compute a public/private key pair U_{pub}/U_{priv} for him, where $U_{pub}=H_1(ID_U)$ and $U_{priv} = sU_{pub}$.

(2) Oblivious transfer phase

In this phase, when a sender possessing n messages (m_1, m_2,..., and m_n) wants to obliviously transfer k messages of them (m_{σ_1}, m_{σ_2},..., and m_{σ_k}) to a chooser, they will execute the following steps, where the public/private key pairs of the sender and chooser are S_{pub}/S_{priv} and C_{pub}/C_{priv} respectively, and $\{\sigma_1, \sigma_2, \ldots, \sigma_k\} \subset \{1, 2, ..., n\}$ are the set of k choices selected by the chooser in advance. We also depict them in Fig.1.

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

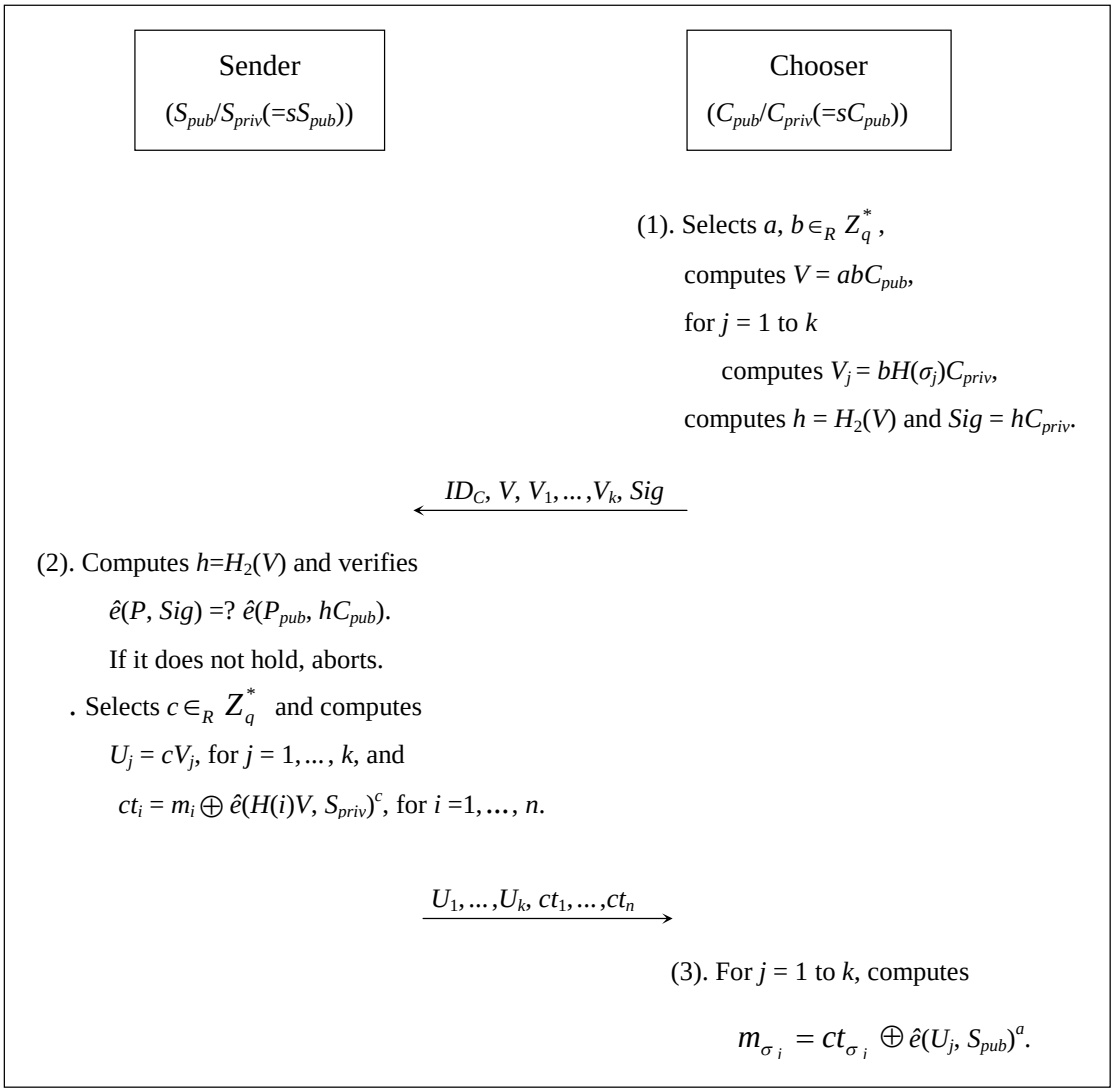

Fig. 1: The proposed *k*-out-of-*n* authentic OT protocol

Step (2): After receiving ID_R, 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 $\hat{e}(P, Sig) = \hat{e}(P_{pub}, hC_{pub})$ 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 $ct_i = m_i \oplus \hat{e}(H(i)V, S_{priv})^c$, where $j = 1, \ldots, k$ and $i = 1, \ldots, n$. He 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 intended messages by computing $m_{\sigma_j} = ct_{\sigma_j} \oplus \hat{e}(U_j, S_{pub})^a$ for j = 1, …, k.

(3) A counter example for chooser's privacy preservation

For demonstrating the chooser's privacy more clearly, we take a counter example. Suppose in step (1) of the protocol, the chooser computes $v_1,\ldots,v_k$, where $V_j = bH(\sigma_j)C_{priv}$, j=1 to k. A misleading may be that $v_i/v_j = H(\sigma_i)/H(\sigma_j)$, since b and C_{priv} are both the same in v_i and v_j. Then a cheating sender can precompute $H(\sigma_i)/H(\sigma_j)$ for each i, j in [1, n]. After the sender receives $v_1,\ldots,v_k$, he can first compute each v_i / v_j for [1, k], and then compare them with the precomputed values. Finally, the sender may guess some or all of the chooser's chooses. So the protocol cannot achieve chooser security. However, the mistake here is that both v_i and v_j are two points in the additive group G_1. The operation v_i / v_j is invalid. In other words, in the group there exists no division operation.

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 attacks such as relay attack and man-in-the-middle attack.

Claim 1: *The proposed protocol is correct.*

Proof: After the protocol run, the chooser can exactly obtain the *k* messages which he selected by computing

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

Claim 2: *The proposed protocol can achieve mutual authentication.*

Proof: We show the hold 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)).

(2).For that the ciphertext ct_i (= $m_i \oplus \hat{e}(H(i)V, S_{priv})^c$) contains the sender's private key S_{priv} (= sS_{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_is and thus can be authenticated by the chooser using his public key.

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

Proof: For each of the chooser's k choices $\sigma_j \in \{1,2,\ldots,n\}$ are first hashed and randomized by H and b respectively then signed as $V_j = bH(\sigma_j)C_{priv}$ by 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 cannot obtain $bH(\sigma_j)$ from V_j due to the hardness of ECDLP. That is, he can not figure out $bH(\sigma_j)$, not to mention σ_j. More formally, let $\mathcal{A}$ = {(b, σ_j) $\in Z_q^* Z_n \mid bH(\sigma_j)C_{priv} = V_j$}; that is, $\mathcal{A}$ consists of all the possible ordered pairs (b, σ_j) satisfying the equation $bH(\sigma_j)C_{priv} = V_j$. If we are given a value V_j, then under fixed C_{priv}, there only exists an 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 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, $\Pr[\sigma_j|V_j] = \Pr[\sigma_j] = 1/n$ which meas that under seeing a specific V_j, the choice σ_j of the chooser can't be revealed other than quessing. 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 chooser $\hat{C}$ wants to obtain more than k messages in the protocol. If he could succeed, then the sender's privacy is violated (see Section 2.1). However, we will prove that 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 that why $\hat{C}$ must follow the protocol to form the values of V and V_js; otherwise, he can not obtain the k chosen messages. In argument (II), we show that if $\hat{C}$ intends to obtain the $(k+1)^{th}$ message, he will face the intractable CTCDH problem under the assumption that $H(.)$ is a random hash function.

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

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

(a) $\hat{C}$ fakes V but forms V_js honestly

Assume that $\hat{C}$ is dishonest in forming V but forms V_js in the same manner as specified in the original protocol. For example, without loss of generality, he replaces V with a specified $X \in G_1$ and computes $V_j = bH(\sigma_j)\hat{C}_{priv}$. Then, the sender will compute $U_j = cV_j$, $ct_i = m_i \oplus \hat{e}(H(i)X, S_{priv})^c$, and send them back to $\hat{C}$. As a result, $\hat{C}$ can not decrypt ct_{σ_j} ($ct_{\sigma_j} = m_{\sigma_j} \oplus \hat{e}(U_j, S_{pub})^a$) to obtain the k messages since $\hat{e}(U_j, S_{pub})^a$ is obviously not equal to $\hat{e}(H(\sigma_j)X, S_{priv})^c$. Perhaps, for obtaining the k messages, $\hat{C}$ may try another way by computing $\hat{e}(H(i)X, S_{priv})^c$ expected to be equal to $\hat{e}(U_j, S_{pub})^a$. But this is computationally infeasible since $\hat{C}$ doesn't 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_js but forms V honestly

Assume that $\hat{C}$ is dishonest in forming V_js but forms V in the same manner as specifed in the original protocol. For example, without loss of generality, he replaces V_j with a specified $X_j \in G_1$ and computes $V = ab\hat{C}_{pub}$. Then, the sender will compute $U_j = cV_j = cX_j$, $ct_i = m_i \oplus \hat{e}(H(i)V, S_{priv})^c = m_i \oplus \hat{e}(H(i)ab\hat{C}_{pub}, S_{priv})^c$, for i=1 to n, and send them back to $\hat{C}$. As a result, $\hat{C}$ can not decrypt ct_{σ_j}s since $\hat{e}(U_j, S_{pub})^a$ is obviously not equal to $\hat{e}(H(i)V, S_{priv})^c$. Perhaps, for obtaining the k messages, $\hat{C}$ may try another way by computing $\hat{e}(H(i)V, S_{priv})^c$ (=$\hat{e}(H(i)ab\hat{C}_{pub}, S_{priv})^c$) expected to be equal to $(U_j, S_{pub})^a$. But again this is computationally infeasible since $\hat{C}$ doesn't know both the sender's private key S_{priv} and the

one-time secrecy c. Even he knows S_{priv}, however, it is an ECDLP to extract c from U_j (=cX_j). Hence, $\hat{C}$ can not compute the value $\hat{e}(H(i)V, S_{priv})^c$ to decrypt ct_{σ_j}s to obtain the k messages, m_{σ_j}s.

(c) $\hat{C}$ fakes both the values of V and V_js

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 $m_{\sigma_j} \oplus \hat{e}(H(\sigma_j)X, S_{priv})^c$, for j=1 to k, or equivalently, $ct_{\sigma_j} = m_{\sigma_j} \oplus \hat{e}(cH(\sigma_j)X, S_{priv})$. Although, $\hat{C}$ knows the value of $cH(\sigma_j)X$ (since it just equals to U_j received from the sender), he still can not compute $\hat{e}(cH(\sigma_j)X, S_{priv})$ without the knowledge of S_{priv}. From above description, we know that when the setting of V is X and V_j is $H(\sigma_j)X$, $\hat{C}$ can not 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, $\hat{C}$ can not obtain the k messages under the violation of setting both the values, V and V_js .

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

That $\hat{C}$ wants to obtain message m_i implies $\hat{C}$ would have the knowledge of $\hat{e}(H(i)V, S_{priv})^c$ (=$\hat{e}(U_j, S_{pub})^a$) (In fact, according to argument (I), an honest chooser $\hat{C}$ could know k of the n values, $\hat{e}(H(i)V, S_{priv})^c$, for i=1 to n, since $\hat{e}(H(i)V, S_{priv})^c = \hat{e}(U_j, S_{pub})^a$, for $i = \sigma_j$ and j =1 to k.). Let $y^{(i)} \in G_2$ and $\hat{e}(H(i)V, S_{priv})^c = y^{(i)}$. According to argument (I), for obtaining the k chosen messages, $\hat{C}$ can not change the structures of V (=$ab\hat{C}_{pub}$) and V_j (=$bH(\sigma_j)\hat{C}_{priv}$). Under this situation, $y^{(i)}$ only can be decomposed as $y^{(i)} = \hat{e}(H(i)ab\hat{C}_{pub}, S_{priv})^c = \hat{e}(abH(i)\hat{C}_{priv}, S_{pub})^c$. Moreover, under the assumption that $H(.)$ is a random hash function and the fact that $\hat{C}$ has the knowledge of a, b, $\hat{C}_{priv}$, and S_{pub}, $y^{(i)}$ can be represented as $(g_i)^c$, where g_i equals to $\hat{e}(abH(i)\hat{C}_{priv}, S_{pub})$ and is a random element in G_2 due to the assumption that $H(.)$ is a random

hash function. Consequently, the problem $\hat{C}$ really faces is that 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 $(g_{\sigma_j})^c = \hat{e}(U_j, S_{pub})^a$, but without the knowledge of sender's one-time secrecy c. This is known as the intractable CTCDH problem introduced in Section 2.3. Therefore, the chooser can not 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_js, 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_js, the adversary can not obtain the k selected messages by computing $m_{\sigma_j} = ct_{\sigma_j} \oplus \hat{e}(U_j, S_{pub})^a$ since he does not know the value of a embedded in the replayed message V. It is computationally infeasible for the adversary to extract a from $V = abC_{pub}$, 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 slinkingly 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. Fig. 2 illustrates the scenario of such a MIMA. We first argue that the adversary E cannot succeed in this scenario since he can not generate the valid message (2), $(ID_C, V', V_1', \ldots, V_k', Sig')$ as shown in the figure. More clearly, without the knowledge of chooser's private key C_{priv}, he can not forge a valid signature Sig' in message (2) to be verified successfully by the sender since Sig' should be equal to $H_2(V)C_{priv}$. 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 $\hat{e}(H(i)V, S_{priv})^c$. Although, E can choose another random nonce c' such that $U_j' = c'V_j$, he still has to know the sender's private key S_{priv} to form the valid $ct_i' (= m_i \oplus \hat{e}(H(i)V, S_{priv})^{c'})$. Therefore, without the knowledge of S_{priv}, E can not launch such a MIMA attack.

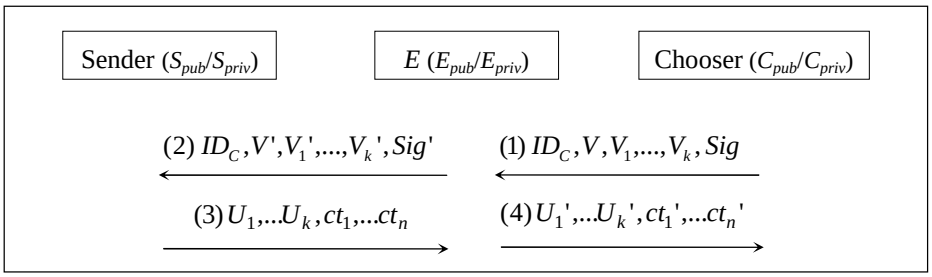

Fig. 2: The scenario of MIMA attack

5.2 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) among our non-adaptive OT_k^n protocol and the other OT_k^n protocols of the same type, such as Chu et al.'s [12] (which is up to now, to our best knowledge, the most efficient OT_k^n scheme), Mu et al.'s [10], Naor et al.'s [5], and recent works [13, 14, 18, 20]. From the needed passes viewpoint, our scheme is the most efficient since it just requires two passes. Moreover, except for the least requirement in needed passes, the data size transmitted in our scheme is also the minimal among such type of OT_k^n schemes. In the following, we will illustrate this by first describing the underlying facts and used notations before making the comparison.

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

Fact(1): For the same security level, a RSA cryptosystem would require a key length of 1024 bits while an ElGamal or ECC-based cryptosystem only needs 160 bits.

Fact(2): The length of the ciphertexts for RSA, ElGamal, and ECC-based cryptosystems are 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 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, $V_1, ...,$ V_k, Sig, $U_1, ...,$ U_k transmitted between the chooser and sender is an ECC point. Thus, the NTDS from a

chooser to a sender is estimated as 160∗(*k*+2) bits and from the sender to the chooser is 160*k*+*n*∗|*ciphertext*| bits. Naor et al.'s scheme [5] 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 of the OT_1^2's work and likewise the NTDS is about log*n* times of the OT_1^2's work. Therefore, their scheme has the most expensive communicational cost. As for Green et al.'s protocol [18], 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) sub-protocol 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 sub-protocol *k* times with the help of the sender to extract the blind choices to decrypt the ciphertexts.

Table 1: Needed rounds and data size comparisons among OT_k^n protocols

Protocol	**passes**	**Size of message: *C*→*S* (bits)**	**Size of message: *S*→*C* (bits)**	**Mutual Authentication**
Ours	2	160∗(*k*+2)	160*k*+*n*∗|*ciphertext*|	yes
Naor et al. [5]	*k*∗log *n* OT_1^2	depends on OT_1^2	depends on OT_1^2	no
Mu et al.'s scheme(1) [10]	3	1024*k*	1024*n*+*nk*∗|*ciphertext*|	no
Mu et al.'s scheme(2) [10]	2	1024∗2*n*	*n*∗|*ciphertext*|	no
Chu et al. [12]	2	1024*k*	1024∗ (*k*+1)+ *n*∗|*ciphertext*		no
Zhang et al. [13]	2	1024∗ (*k*+3)	1024*n*+ *n*∗|*ciphertext*		no
Huang et al. [14]	3	1024*k*	(*n*+*k*)∗|*ciphertext*		
Green et al. [18]	2+*k*∗Pok	|Pok|+*k*∗|BlindExtract|	*n*∗| *ciphertext* |+|Pok|+ *k*∗|BlindExtract|	no
Chang et al. [20]	4	1024*k*	(*n*+2*k*+2)*1024	no

Consequently, the number of passes for executing protocol [18] is 2+*k*∗Pok, where Pok represents the required passes for executing the proof-of-knowledge sub-protocol. Besides, the NTDS from chooser to sender is estimated as |Pok| + *k*∗|BlindExtract|, and from sender to chooser is *n*∗|*ciphertext*| + |Pok| + *k*∗|BlindExtract|. Similarly, the passes and NTDS of other studies can be estimated in the same manner. We show the comparison results in Table 1. From Table 1, 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 while compared with other works.

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 almost of them lack the security services, such as mutual authentication, and the prevention of replay and main-in-the-middle attacks. Hence, they should run under a secure channel when applied in commercial applications. This will increase execution overhead. Hence, 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 non-adaptive *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 1. From Table 1, 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 integrated the function of mutual authentication nowadays.

Reference

[1] M. O. Rabin, "How to exchange secrets with oblivious transfer, "*Technical Report TR-81*, Aiken Computation Lab, Harvard University, 1981.

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

[3] G. Brassard, C. Crepeau, and J.-M. Robert, "All-or-nothing disclosure of secrets," *Proc. Advances in Cryptology: CRYPTO'86, LNCS 263*, pp. 234-238, 1986.

[4] M. Bellare and S. Micali, "Non-interactive oblivious transfer and applications," *Proc. of Advances in Cryptology: CRYPTO'89, LNCS 435*, pp.547-557, 1989.

[5] M. Naor and B. Pinkas, "Oblivious transfer with adaptive queries," *Proc. Advances in Cryptology: CRYPTO'99, LNCS 1666*, pp. 573-590, 1999.

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

[7] M. Naor and B. Pinkas, "Distributed oblivious transfer," *Proc. Advances in Cryptology: ASIACRYPT'00, LNCS 1976*, 2000.

[8] M. Naor and B. Pinkas, "Oblivious transfer and polynomial evaluation," *Pro. of*

the 31th Annual ACM Symposium on the Theory of Computing (STOC'99), pp.245-254, ACM, 1999.

[9] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," *SODA'01*, pp.448-457, 2001.

[10] Y. Mu, J. Zhang, and V. Varadharajan, "*m* out of *n* oblivious transfer," *Proc. of the 7th Australasian Conference on Information Security and Privacy (ACISP'02)*, *LNCS 2384*, pp. 395-405, 2002.

[11] W. Ogata and K. Kurosawa, "Oblivious keyword search," *Journal of Complexity*, 20(2-3), pp.356-371, 2004.

[12] C. K. Chu, W. G. Tzeng, "Efficient *k*-out-of-*n* oblivious transfer Schemes with adaptive and non-adaptive queries," *PKC 2005*, *LNCS 3386*, pp. 172-183, 2005.

[13] J. Zhang, Y. Wang, "Two provably secure *k*-out-of-*n* oblivious transfer schemes," *Applied Mathematics and Computation*, vol. 169, pp. 1211-1220, 2005.

[14] H. F. Huang, C. C. Chang, "A new design for efficient *t*-out-*n* oblivious transfer scheme," *Advanced Information Networking and Applications 2, ANIA 2005*, pp. 28-30, 2005**.**

[15] A. Parakh, "Oblivious transfer using elliptic curves," *Proc. of the 15th International Conference on Computing*, IEEE, pp. 323-328, 2006.

[16] 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, issue 3, March, 2009.

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

[18] J. Camenish, G. Neven, and A. Shelat, "Simulatable adaptive oblivious transfer," *EUROCRYPT 2007, LNCS 4515*, pp. 573-590, 2007.

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

[20] C. C. Chang and J. S. Lee, "Robust *t*-out-of-*n* oblivious transfer mechanism based on CRT," *Journal of Network and Computer Applications, 32(2009)*, pp.226-235, 2009.

[21] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme," *Proc. of the Public-Key Cryptography (PKC'03)*, pp.31-46, 2003.

[22] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko, "Power of RSA inversion oracles and the security of Chaum's RSA-based blind signature scheme," *Proceedings of Financial Cryptography (FC'01), LNCS 2248*, pp.319-338, 2001.

[23] L.M. Kohnfelder, "On the signature reblocking problem in public-key

cryptography," *Communications of the ACM, vol. 21(2)179*, 1978.

[24] D. Boneh and M.K. Franklin, "Identity-based encryption from the Weil Pairing," *Proc. of Advances in Cryptology: CRYPTO'01, LNCS 2139*, pp. 213-229, 2001.

[25] Ya-Fen Chang and Wei-Cheng Shiao, "The essential design principles of verifiable non-interactive OT protocols," IEEE ISDA, Eighth International Conference, Volume: 3, On page(s): 241-245, Nov. 2008

[26] Qin Jing, Zhao Hua-wei, and Wang Ming-Qiang, "Non-interactive oblivious transfer protocols," IEEE IFITA '09, Volume 2, Page(s):120 – 124, May 2009

[27] Hossein Ghodosi, "On insecurity of Naor-Pinkas' distributed oblivious transfer, "Information Processing Letters, Volume 104,2007

[28] Hossein Ghodosi, Rahim Zaare-Nahandi, "Comments on the 'm out of n oblivious transfer, "Information Processing Letters, Volume 97, Issue 4, 28 February 2006, Pages 153-155

[29] J. Qin, H. W. Zhao and M. Q. Wang, "Non-interactive Oblivious Transfer Protocols", *Proc. of Information Technology and Applications, 2009*, IFITA '09, pp.120-124.

[30] Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information", Annual ACM Symposium on Theory of Computing, 1982, pp.365-377.

[31] J. S. Chou and Y.S. Yeh, "Mental poker game based on a bit commitment scheme through network", Computer Networks, 38(2), 247-255, 2002.