Conditional Oblivious Cast [⋆]

Cheng-Kang Chu and Wen-Guey Tzeng
Department of Computer Science,
National Chiao Tung University,
Hsinchu, Taiwan 30050
Email:{ckchu,tzeng}@cis.nctu.edu.tw

Abstract. We introduce a new notion of *conditional oblivious cast* (COC), which involves three parties: a sender S and two receivers A and B. Receivers A and B own their secrets x and y, respectively, and the sender S holds the message m. In a COC scheme for the predicate Q (Q-COC), A and B send x and y in a masked form to S, and then S sends m to A and B such that they get m if and only if $Q(x, y) = 1$. Besides, the secrets x and y can not be revealed to another receiver nor the sender. We also extend COC to 1-out-of-2 COC (COC_2^1) in which S holds two messages m_0 and m_1, and A and B get m_1 if $Q(x, y) = 1$ and m_0 otherwise. We give the definitions for COC and COC_2^1, and propose several COC and COC_2^1 schemes for "equality", "inequality", and "greater than" predicates. These are fundamental schemes that are useful in constructing more complex secure interactive protocols. Our schemes are efficiently constructed via homomorphic encryption schemes and proved secure under the security of these encryption schemes.

Keywords: oblivious cast, conditional oblivious transfer, secure computation.

1 Introduction

Oblivious transfer (OT) is an important cryptographic primitive proposed by Rabin [18]. It involves two parties: the sender S and the receiver R, where S sends a bit of which R gets it with probability $\frac{1}{2}$. After Rabin's work, OT was developed in several types, such as 1-out-of-2 OT [11], 1-out-of-n OT [5, 16, 21], k-out-of-n OT [8, 14, 15], conditional OT (COT) [3, 10], etc. In Q-COT, S owns a secret x and a message m, and R owns a secret y such that R gets m from S if and only if the condition $Q(x, y)$ is evaluated as true.

Oblivious cast (OC) [12] is a generalization of OT to the three-party case: one sender S and two receivers A and B. The bit is received by exactly one of A and B, each with probability $\frac{1}{2}$. We generalize OC and introduce a new notion of *conditional oblivious cast* (COC), where A and B own their secrets x and y, respectively, and the sender S holds the message m. In a COC scheme for the predicate Q (Q-COC), A and B send x and y in a masked form to S, and

[⋆] Research supported in part by National Science Council grants NSC-94-2213-E-009-116, Taiwan, ROC.

then S sends m to A and B such that they get m if and only if $Q(x, y) = 1$. Furthermore, the secrets x and y can not be revealed to another receiver nor the sender. We also extend COC to 1-out-of-2 COC (COC_2^1) in which S holds two messages m_0 and m_1, and A and B get m_1 if $Q(x, y) = 1$ and m_0 otherwise.

There are two cases for the message receiving: A and B both get m, or only one of them gets m. The schemes we propose in this paper are all designed for the first case. However, in some applications only one receiver, determined by the condition, is allowed to get the message, and S can not know who gets the message. We have a general transformation of our COC_2^1 schemes to suit this kind of model (Section 4.3).

In this paper, we give the definitions for COC and COC_2^1, and propose several COC and COC_2^1 schemes for "equality", "inequality", and "greater than" predicates. These are fundamental schemes that are useful in constructing more complex secure interactive protocols. Our schemes are efficiently constructed via homomorphic encryption schemes and proved secure.

COC not only covers all functionalities of COT, but also broadens the range of its applications. We provide three examples:

- *Priced oblivious transfer*: Aiello et al. [1] introduced the notion of "priced oblivious transfer", which protects the privacy of a customer's purchase from a vendor. In their setting, the buyer needs to deposit an amount in each vendor. This is not very practical if a buyer wants to purchase various goods from many vendors. By using our COC schemes, we can construct a generalized priced OT such that the buyer can deposit the money in one bank only. When the buyer wants to buy an item from a vendor, he sends the corresponding price and the bank sends the buyer's current balance in the encryption form to the vendor. The vendor then sends the item such that the buyer can get it if the price does not exceed his balance.
- *Oblivious two-bidder system*: A party S has a secret for selling, and A and B are two bidders. The winner can obtain the secret from S directly. At the end, S has no idea who the winner is. This system can be constructed from COC for the "greater than" predicate (in the second message-receiving case) immediately.
- *Oblivious authenticated information retrieval*: A can get some information from S if he passes the authentication procedure provided by B. For instance, consider a mobile news subscription service provided by an independent agent. We assume that a mobile phone has no extra memory to store the subscription information but only an IMSI (International Mobile Subscriber Identity) in the SIM card. Users can pay the subscription fee to their mobile phone company, and the company provides an encrypted subscription list of IMSIs to the news provider. When a user wants to read news on the bus, his mobile phone sends the encrypted IMSI to the news provider. The news provider then sends news to the user if the IMSI is in the subscription list. In this case, the user's identity (IMSI) is anonymous to the news provider. The scheme can be constructed by COC for the "membership" predicate discussed in Section 5.2.

Related works. COT was first proposed by Di Crescenzo et al. [10]. In their definition of COT, the focus is to provide "all-or-nothing" transfer of the message from S to R by the condition. Blake et al. [3] strengthened COT to strong COT (SCOT), which provides "1-out-of-2" message transfer from S to R by the condition and adds more security requirements for S.

The notion of our COC is to separate the role of the secret holder from S. The main difference in design techniques is that, in COT and SCOT, the secure computation is done by S with a masked input and a plain input, whereas the secure computation in our COC and COC_2^1 is done by S with two masked inputs. A COC scheme that meets the requirements of our definitions can be easily transferred to a COT or SCOT scheme.

2 Definitions and Preliminaries

In this section we give formal definitions for COC and COC_2^1 and introduce useful tools and notations.

2.1 Conditional Oblivious Cast

Informally speaking, a COC scheme for predicate Q (Q-COC) has the following three properties:

- Correctness: both of A and B get m from S if $Q(x, y) = 1$.
- Sender's security: A and B cannot get any information about m if $Q(x, y) = 0$.
- Receiver's security: after running the protocol, x is kept secret from B and S, and y is kept secret from A and S.

The definition for Q-COC is as follows:

Definition 1 (Q-COC). *Let k be the security parameter, and A, B and S be all polynomial-time probabilistic Turing machines (PPTMs). Let $\langle A, B, S\rangle(\cdot)$ denote the communication transcript. We say that a three-party interactive system $\Pi = (A, B, S)$ is a secure Q-COC scheme if it satisfies the following requirements for some constant c:*

1. *Correctness: For any $x, y, m \in \{0,1\}^{k^c}$ with $Q(x, y) = 1$,*
   $\Pr[\mu \leftarrow \{0,1\}^{k^c}; tr \leftarrow \langle A(x), B(y), S(m)\rangle(\mu) :$
   $\quad$ "$A(x, \mu, tr) = m$" $\wedge$ "$B(y, \mu, tr) = m$"$] = 1.$
2. *Sender's security: For any PPTM A', B' and any $x, y, m, m' \in \{0,1\}^{k^c}$ with $Q(x, y) = 0$, A' and B' cannot distinguish the following probability ensembles with non-negligible advantage, respectively:*
   - $V^{\Pi}_{A'B'} = (x, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B'(y), S(m)\rangle(\mu)),$
   - $R^{\Pi}_{A'B'} = (x, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B'(y), S(m')\rangle(\mu)),$

   and
   - $V^{\Pi}_{B'A'} = (y, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B'(y), S(m)\rangle(\mu)),$
   - $R^{\Pi}_{B'A'} = (y, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B'(y), S(m')\rangle(\mu)).$

3. *Receiver's security:*
   (a) *For any PPTM A', B', S' and any $x, x', y, y', m \in \{0,1\}^{k^c}$ with $Q(x,y) = Q(x,y') = Q(x',y)$, S' cannot distinguish the following probability ensembles with non-negligible advantage:*
      - $V^{\Pi}_{S'A'} = (m, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B(y), S'(m)\rangle(\mu))$,
      - $S^{\Pi}_{S'A'} = (m, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B(y'), S'(m)\rangle(\mu))$,

      and
      - $V^{\Pi}_{S'B'} = (m, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A(x), B'(y), S'(m)\rangle(\mu))$,
      - $S^{\Pi}_{S'B'} = (m, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A(x'), B'(y), S'(m)\rangle(\mu))$.

   (b) *For any PPTM A', B', S' and any $x, x', y, y', m \in \{0,1\}^{k^c}$ with $Q(x,y) = Q(x,y') = Q(x',y)$, A' and B' cannot distinguish the following probability ensembles with non-negligible advantage, respectively:*
      - $V^{\Pi}_{A'S'} = (x, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B(y), S'(m)\rangle(\mu))$,
      - $S^{\Pi}_{A'S'} = (x, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B(y'), S'(m)\rangle(\mu))$,

      and
      - $V^{\Pi}_{B'S'} = (y, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A(x), B'(y), S'(m)\rangle(\mu))$,
      - $S^{\Pi}_{B'S'} = (y, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A(x'), B'(y), S'(m)\rangle(\mu))$.

2.2 1-out-of-2 Conditional Oblivious Cast

In COC$^1_2$, the message sender S holds two messages m_0 and m_1. A Q-COC$^1_2$ scheme must satisfy the following three properties:

- Correctness: both of A and B get m_1 from S if $Q(x,y) = 1$, and m_0 if $Q(x,y) = 0$.
- Sender's security: A and B get exactly one message from S.
- Receiver's security: after running the protocol, x is kept secret from B and S, and y is kept secret from A and S.

The definition for Q-COC$^1_2$ is as follows.

Definition 2 (Q-COC$^1_2$). *Let k be the security parameter, and A, B and S be all PPTMs. Let $\langle A, B, S\rangle(\cdot)$ denote the communication transcript. We say that a three-party interactive system $\Pi = (A, B, S)$ is a secure Q-COC$^1_2$ scheme if it satisfies the following requirements for some constant c:*

1. *Correctness:*
   (a) *For any $x, y, m_0, m_1 \in \{0,1\}^{k^c}$ with $Q(x,y) = 0$,*
   $\Pr[\mu \leftarrow \{0,1\}^{k^c}; tr \leftarrow \langle A(x), B(y), S(m_0, m_1)\rangle(\mu) :$
   $\text{“}A(x,\mu,tr) = m_0\text{”} \wedge \text{“}B(y,\mu,tr) = m_0\text{”}] = 1.$
   (b) *For any $x, y, m_0, m_1 \in \{0,1\}^{k^c}$ with $Q(x,y) = 1$,*
   $\Pr[\mu \leftarrow \{0,1\}^{k^c}; tr \leftarrow \langle A(x), B(y), S(m_0, m_1)\rangle(\mu) :$
   $\text{“}A(x,\mu,tr) = m_1\text{”} \wedge \text{“}B(y,\mu,tr) = m_1\text{”}] = 1.$
2. *Sender's security: For any PPTM A', B' and any $x, y, m_0, m_1, m'_1 \in \{0,1\}^{k^c}$ with $Q(x,y) = 0$, A' and B' cannot distinguish the following probability ensembles with non-negligible advantage, respectively:*
   - $V^{\Pi}_{A'B'} = (x, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B'(y), S(m_0, m_1)\rangle(\mu))$,

- $R^{\Pi}_{A'B'} = (x, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B'(y), S(m_0, m_1')\rangle(\mu))$,

and

- $V^{\Pi}_{B'A'} = (y, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B'(y), S(m_0, m_1)\rangle(\mu))$,
- $R^{\Pi}_{B'A'} = (y, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B'(y), S(m_0, m_1')\rangle(\mu))$.

The similar requirements is met $Q(x,y) = 1$.

3. *Receiver's security:*
   (a) *For any PPTM A', B', S' and any $x, x', y, y', m_0, m_1 \in \{0,1\}^{k^c}$ with $Q(x,y) = Q(x,y') = Q(x',y)$, S' cannot distinguish the following probability ensembles with non-negligible advantage:*
   - $V^{\Pi}_{S'A'} = (m_0, m_1, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B(y), S'(m_0, m_1)\rangle(\mu))$,
   - $S^{\Pi}_{S'A'} = (m_0, m_1, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B(y'), S'(m_0, m_1)\rangle(\mu))$,

   and

   - $V^{\Pi}_{S'B'} = (m_0, m_1, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A(x), B'(y), S'(m_0, m_1)\rangle(\mu))$,
   - $S^{\Pi}_{S'B'} = (m_0, m_1, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A(x'), B'(y), S'(m_0, m_1)\rangle(\mu))$.

   (b) *For any PPTM A', B', S' and any $x, x', y, y', m_0, m_1 \in \{0,1\}^{k^c}$ with $Q(x,y) = Q(x,y') = Q(x',y)$, A' and B' cannot distinguish the following probability ensembles with non-negligible advantage, respectively:*
   - $V^{\Pi}_{A'S'} = (x, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B(y), S'(m_0, m_1)\rangle(\mu))$,
   - $S^{\Pi}_{A'S'} = (x, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A'(x), B(y'), S'(m_0, m_1)\rangle(\mu))$,

   and

   - $V^{\Pi}_{B'S'} = (y, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A(x), B'(y), S'(m_0, m_1)\rangle(\mu))$,
   - $S^{\Pi}_{B'S'} = (y, \mu \leftarrow \{0,1\}^{k^c}, tr \leftarrow \langle A(x'), B'(y), S'(m_0, m_1)\rangle(\mu))$.

Remark. For clarity and simplicity, we will first assume that all parties in our COC and COC$^1_2$ schemes are semi-honest (honest-but-curious), that is, they follow the procedure step by step, but try to get extra information about the secrets or messages by extra computation. We also assume that A, B and S operates independently. No two parties will collude against the third one. Then we provide some techniques to transform the schemes into ones that are secure against malicious parties and their collusion in Section 5.1.

2.3 Homomorphic Encryption Schemes

Multiplicatively homomorphic encryption scheme. An encryption scheme (G, E, D) is multiplicatively homomorphic if for any m_0 and m_1, $D(E(m_0) \otimes E(m_1)) = D(E(m_0 \cdot m_1))$, where $\otimes$ is an operation defined on the image of E.

The ElGamal encryption scheme as follows is multiplicatively homomorphic.

- $G(1^k) = (p, q, g, \alpha, \beta)$, where p is a k-bit prime, and $q = \frac{p-1}{2}$ is also a prime, $\mathbb{G}_q$ is the subgroup of $\mathbb{Z}_p^*$ with order q, g is a generator of $\mathbb{G}_q$, and $\beta = g^\alpha \bmod p$ for $\alpha \in \mathbb{G}_q$. Let $PK = (p, q, g, \beta), SK = (p, q, g, \alpha)$. All relevant computations are under group $\mathbb{G}_q$.
- $E(m) = (g^r, m\beta^r)$, where $m \in \mathbb{G}_q, r \in_R \mathbb{Z}_q$.
- $D(c) = c_2/c_1^\alpha$, where $c = (c_1, c_2)$.

For $E(m_0) = (g^{r_0}, m_0\beta^{r_0})$ and $E(m_1) = (g^{r_1}, m_1\beta^{r_1})$, the operation $E(m_0) \times E(m_1) = (g^{r_0} \cdot g^{r_1}, m_0\beta^{r_0} \cdot m_1\beta^{r_1})$ is multiplicatively homomorphic since

$$\begin{aligned} D(E(m_0) \times E(m_1)) &= D(g^{r_0} \cdot g^{r_1}, m_0\beta^{r_0} \cdot m_1\beta^{r_1}) \\ &= D(g^{r_0+r_1}, m_0m_1\beta^{r_0+r_1}) \\ &= D(E(m_0 \cdot m_1)). \end{aligned}$$

We can compute $E(m^c)$ from $E(m)$ via repeated multiplication for a constant c.

Additively homomorphic encryption scheme. An encryption scheme (G, E, D) is additively homomorphic if for any m_0 and m_1, $D(E(m_0) \oplus E(m_1)) = D(E(m_0 + m_1))$, where $\oplus$ is an operation defined on the image of E.

The Paillier encryption scheme [17] as follows is additively homomorphic.

- $G(1^k) = (p, q, N, \alpha, g)$, where $N = pq$ is a k-bit number, p and q are two large primes, g is an integer of order $\alpha N \bmod N^2$ for some integer α. Let $PK = (g, N), SK = \lambda(N) = \mathrm{lcm}(p-1, q-1)$.
- $E(m) = g^m r^N \bmod N^2$, where $m \in \mathbb{Z}_N, r \in_R \mathbb{Z}_N$.
- $D(c) = \frac{L(c^{\lambda(N)} \bmod N^2, N)}{L(g^{\lambda(N)} \bmod N^2, N)} \bmod N$, where $L(u, N) = \frac{u-1}{N}$.

For $E(m_0) = g^{m_0} r_0^N \bmod N^2, E(m_1) = g^{m_1} r_1^N \bmod N^2$, the operation $E(m_0) \cdot E(m_1) = (g^{m_0} r_0^N) \cdot (g^{m_1} r_1^N)$ is additively homomorphic since

$$\begin{aligned} D(E(m_0) \cdot E(m_1)) &= D((g^{m_0} r_0^N) \cdot (g^{m_1} r_1^N)) \\ &= D((g^{m_0+m_1} (r_0r_1)^N)) \\ &= D(E(m_0 + m_1)). \end{aligned}$$

We can compute $E(cm)$ from $E(m)$ via repeated addition for a constant c.

Note that ElGamal and Paillier encryption schemes are proved semantically secure if and only if the Decisional Diffie-Hellman and the Computational Composite Residuosity assumptions hold, respectively [20, 17].

2.4 0-encoding and 1-encoding

In our COC scheme for "greater than" predicate, we use two types of encoding to reduce the "greater than" problem to the set intersection problem [13]. Let $s = s_n s_{n-1} \ldots s_1 \in \{0,1\}^n$ be a binary string of length n. The 0-encoding of s is

$$\hat{S}_s^0 = \{s_n s_{n-1} \ldots s_{i+1} 1 | s_i = 0, 1 \leq i \leq n\}.$$

and 1-coding of s is

$$\hat{S}_s^1 = \{s_n s_{n-1} \ldots s_i | s_i = 1, 1 \leq i \leq n\}.$$

For two binary strings x, y of the same length, we have that $x > y$ if and only if there is exact one common element in $\hat{S}_x^1$ and $\hat{S}_y^0$.

If we compare strings in $\hat{S}_x^1$ and $\hat{S}_y^0$ one against one, it would be quite inefficient since we need $O(n^2)$ comparisons. Because each element in $\hat{S}_s^0$ (or $\hat{S}_s^1$) has a different length, we compare the elements of the same length in the two sets only. We define the *ordered* sets for $b \in \{0,1\}, 1 \le i \le n$:

$$S_s^b[i] = \begin{cases} z_i & \text{if } \exists z_i \in \hat{S}_s^b \text{ and } |z_i| = i; \\ r_i^b & \text{otherwise,} \end{cases}$$

where $S_s^b[i]$ denotes the i-th element in S_s^b, and r_i^b is an arbitrary binary string with length $i{+}1{+}b$. Therefore, because of different lengths, r_i^b must not be equal to the string $S_s^{1-b}[i]$. Thus we just need to test if $S_x^1[i] = S_y^0[i]$ for each $i \in \{1,2,\ldots,n\}$.

2.5 Setup and Notations

In the setup phase of our schemes for semi-honest adversary, A and B need to agree on a public/secret key pair (PK, SK) of the homomorphic encryption scheme privately. There are several ways to accomplish this work. For example, if A and B have their own public/secret key pairs, one party generates (PK, SK) first, and securely sends it to the other party. This common key pair allows S to compute the predicate on their secrets by the homomorphic encryption scheme. Also, S need choose a key pair (PK_S, SK_S) (for any semantically secure public key encryption scheme) such that A and B can send their secrets to S privately (against the other party).

Let $\mathbb{G}_q$ be the group of the multiplicatively homomorphic encryption scheme and $\mathbb{Z}_N$ be the group of the additively homomorphic encryption scheme. For key pair (PK, SK), E_{PK} and D_{SK} represent encryption and decryption for the underlying encryption scheme.

We use x_i to denote the i-th bit of the value $x = x_n x_{n-1} \cdots x_1$. Let $X[i]$ denote the i-th element of the ordered set X. Let $x \in_R X$ mean that x is chosen from X uniformly and independently. Let $|x|$ be the length (in bits) of x. To encrypt a vector $v = \langle v_1, v_2, \ldots, v_n \rangle$, we write $E(v) = \langle E(v_1), E(v_2), \ldots, E(v_n) \rangle$.

In some schemes, A and B need to "identify" the correct message from a set of decrypted ciphertexts. This can be achieved by some padding technique (e.g. OAEP [2]) such that receivers can check the integrity of a message. If a decryption contains the valid padding, it is the correct message with overwhelming probability.

3 Conditional Oblivious Cast

We provide COC schemes for three basic predicates: "equality", "inequality", and "greater than".

3.1 COC for "Equality" Predicate

To determine if $x = y$, we compute x/y via the multiplicatively homomorphic encryption scheme. If $x/y = 1$, A and B get the message m; otherwise, they get nothing. The scheme EQ-COC is described in Figure 1.

- System parameters: (p, q, g).
- Message sender S has a message m and a key pair (PK_S, SK_S).
- Receiver A has a secret x, and receiver B has a secret y, where $x, y \in \mathbb{G}_q$.
- Receiver A and B have a common key pair (PK, SK)

1. A and B send $E_{PK_S}(E_{PK}(x))$ and $E_{PK_S}(E_{PK}(y))$ to S respectively.
2. S decrypts the received messages to get $E_{PK}(x)$ and $E_{PK}(y)$. S computes
$$e = E_{PK}(m) \otimes (E_{PK}(x) \otimes E_{PK}(y)^{-1})^r$$
and sends it to A and B, where $r \in_R \mathbb{Z}_q$.
3. A and B compute $\hat{m} = D_{SK}(e)$ and identify whether $\hat{m}$ is valid.

Fig. 1. COC scheme for "Equality" predicate: EQ-COC

Theorem 1. *The EQ-COC scheme has the correctness property, unconditional sender's security, and computational receiver's security if the underlying homomorphic encryption scheme has semantic security.*

Proof. For correctness, if $x = y$, A and B compute m by

$$\begin{aligned} D_{SK}(e) &= D_{SK}(E_{PK}(m) \otimes (E_{PK}(x) \otimes E_{PK}(y)^{-1})^r) \\ &= D_{SK}(E_{PK}(m) \otimes (E_{PK}(1)^r)) \\ &= D_{SK}(E_{PK}(m)) \\ &= m. \end{aligned}$$

For sender's security, we show that if $x \neq y$, m is unconditionally secure to A and B. Since $e = E_{PK}(m) \otimes (E_{PK}(x) \otimes E_{PK}(y)^{-1})^r) = E_{PK}(m \cdot (x/y)^r), r \in_R \mathbb{Z}_q$, for any possible m', there is another $r' \in \mathbb{Z}_q$ such that $e = E_{PK}(m' \cdot (x/y)^{r'})$. As long as $x \neq y$, e can be decrypted to any possible message in $\mathbb{G}_q$. This ensures unconditional security of S's message m.

For receiver's security, it is easy to see that S gets no information about x and y due to semantic security of the encryption scheme. Since A and B are symmetric, we only prove the security of B against A. We construct a simulator S_A for A's real view

$$V_A(PK, SK, PK_S, x) = (PK, SK, PK_S, x, E_{PK_S}(E_{PK}(x)), E_{PK_S}(E_{PK}(y)), e).$$

The simulator S_A on input $(PK, SK, PK_S, x, \hat{m})$ is as follows, where $\hat{m}$ (may be a valid message or a random value) is the output of a real execution:

1. Choose a random value $y^* \in \mathbb{G}_q$.
2. Compute $e^* = E_{PK}(\hat{m})$.
3. Output $(PK, SK, PK_S, x, E_{PK_S}(E_{PK}(x)), E_{PK_S}(E_{PK}(y^*)), e^*)$.

By semantic security of the encryption scheme, A cannot distinguish the ciphertexts $E_{PK_S}(E_{PK}(y^*))$ and $E_{PK_S}(E_{PK}(y))$. Furthermore, since e^* is identically distributed as e, the output of S_A is indistinguishable from V_A. Therefore, A gets no information about y except those computed from x and $\hat{m}$. □

- System parameters: n.
- Message sender S has a message m and a key pair (PK_S, SK_S).
- Receiver A has a secret x, and receiver B has a secret y, where $|x| = |y| = n$.
- Receiver A and B have a common key pair (PK, SK), where $PK = (g, N)$.

1. A and B send $E_{PK_S}(E_{PK}(x_i))$ and $E_{PK_S}(E_{PK}(y_i))$ to S respectively, $1 \leq i \leq n$.
2. For each $i \in \{1, 2, \dots, n\}$, S decrypts the received messages to get $E_{PK}(x_i)$ and $E_{PK}(y_i)$, and computes the following values via homomorphic encryption:
   (a) $d_i = x_i - y_i$, $d'_i = x_i + y_i - 1$.
   (b) $e_i = 2e_{i+1} + d_i$, where $e_{n+1} = 0$.
   (c) $c_i = m + r_i(e_i - d_i + d'_i)$, where $r_i \in_R \mathbb{Z}_N$
3. S sends $E_{PK}(c)$ in a random order to A and B, where $c = \langle c_1, c_2, \dots, c_n \rangle$.
4. A and B decrypt the received messages and identify the correct message if existent.

Fig. 2. COC scheme for "Inequality" predicate: INE-COC

In the scheme, we assume $x, y \in \mathbb{G}_q$. If the length of x (or y) is longer than $|p|$, A and B compare $h(x)$ and $h(y)$, where h is a collision-resistant hash function. This technique is applied to later schemes whenever necessary.

3.2 COC for "Inequality" Predicate

COC for the "inequality" predicate is more complicated than that for the "equality" predicate. A and B need to send the ciphertexts of their secrets bit by bit. We use additively homomorphic encryption schemes in this scheme, which is depicted in Figure 2.

In the scheme, $d_i = x_i - y_i$ and $d'_i = x_i - \bar{y}_i$ are 0, 1 or -1. If $x_i = y_i$, $d_i = 0$; otherwise, $d'_i = 0$. Let l be the leftmost different bit between x and y, i.e. the largest i such that $d_i \neq 0$. We have $e_i = 0$ if $i > l$, $e_i \neq 0$ if $i < l$, and $e_i = d_i$ if $i = l$.

If $x \neq y$, the message m is embedded into the index i at which x_i and y_i are distinct. However, we have to avoid leaking information of the number of distinct bits. So S masks m with random values on all indices except the index l. It leaves only one copy of m in c_i's:

- For $i = l$, since $e_l = d_l$ and $d'_l = x_l - \bar{y}_l = 0$, $(e_l - d_l + d'_l) = 0$. Therefore, $c_l = m$.
- For $1 \leq i < l$, c_i would be a random value because $e_i - d_i + d'_i = 2e_{i+1} + d'_i \neq 0$ and $r_i \in_R \mathbb{Z}_N$.
- For $l < i \leq n$, c_i is also a random value because $e_i = d_i = 0$, $d'_i \neq 0$ and $r_i \in_R \mathbb{Z}_N$.

Theorem 2. *The INE-COC scheme has the correctness property, unconditional sender's security, and computational receiver's security if the underlying homomorphic encryption scheme has semantic security.*

- System parameters: (p, q, g).
- Message sender S has a message m and a key pair (PK_S, SK_S).
- Receiver A has a secret x, and receiver B has a secret y, where $x, y \in \mathbb{G}_q$, $|x| = |y| = n$.
- Receiver A and B have a common key pair (PK, SK)

1. A encodes x as S_x^1, and sends $E_{PK_S}(E_{PK}(S_x^1[i]))$ to S, $1 \le i \le n$.
2. B encodes y as S_y^0, and sends $E_{PK_S}(E_{PK}(S_y^0[i]))$ to S, $1 \le i \le n$.
3. S decrypts the received messages and computes
$$e_i = E_{PK}(m) \otimes (E_{PK}(S_x^1[i]) \otimes E_{PK}(S_y^0[i])^{-1})^{r_i},$$
where $r_i \in_R \mathbb{G}_q$, $1 \le i \le n$. S sends e_i's to A and B in a random order.
4. A and B search $\hat{m}_i = D_{SK}(e_i)$, $1 \le i \le n$, to identify the correct m if existent.

Fig. 3. COC scheme for "Greater Than" predicate: GT-COC

Proof. (sketch) Let l be the index of the first different bit of x and y (from the most significant bit). We see that $d_l = e_l = x_l - y_l = 1$ or -1, and $d_l' = x_j - \bar{y}_j = 0$. Therefore, $c_l = m + r_l(e_l - d_l + d_l') = m + r_l \cdot 0 = m$. Thus, A and B get m from the permutation of the encryptions.

For sender's security, we see that if $x = y$, all d_i's and e_i's are 0, and all d_i''s are not 0 (in fact, $+1$ or -1). Thus, for each index i, $c_i = m + r_i(0 \pm 1) = m \pm r_i$. Since for any possible $\tilde{m}$, there exists an $\tilde{r}_i$ such that $c_i = \tilde{m} + \tilde{r}_i$, m is unconditionally secure to A and B.

For receiver's security, S gets no information about x and y by the semantic security of the encryption scheme. As in the proof of EQ-COC, for each of A and B, we can construct a simulator such that the adversary cannot distinguish the real view and the simulated view. Therefore the receiver's security holds. □

3.3 COC for "Greater Than" Predicate

For the "greater than" predicate, we use the encoding methods mentioned in Section 2.4. A encodes x via 1-encoding and B encodes y via 0-encoding. The problem is then reduced to the "equality" problem immediately. When S receives encrypted S_x^1 and S_y^0, he checks equality for corresponding strings. The scheme is presented in Figure 3. The security argument is the same as the proof of the EQ-COC scheme. This method is more efficient than the GT-COC$_2^1$ scheme (in the next section, by setting m_0 as a random number).

4 1-out-of-2 Conditional Oblivious Cast

In this section, we present COC$_2^1$ schemes for the "equality" ("inequality") and "greater than" predicates.

- System parameters: n.
- Message sender S has messages: (m_0, m_1) and a key pair (PK_S, SK_S).
- Receiver A has a secret x, and receiver B has a secret y, where $|x| = |y| = n$.
- Receiver A and B have a common key pair (PK, SK), where $PK = (g, N)$.

1. A and B send $E_{PK_S}(E_{PK}(x_i))$ and $E_{PK_S}(E_{PK}(y_i))$ to S respectively, $1 \leq i \leq n$.
2. For each $i \in \{1, 2, \ldots, n\}$, S decrypts the received messages to get $E_{PK}(x_i)$ and $E_{PK}(y_i)$, and computes the following values via homomorphic encryption:
   (a) $d_i = x_i - y_i, d'_i = x_i + y_i - 1$.
   (b) $e_i = 2e_{i+1} + d_i$, where $e_{n+1} = 0$.
   (c) $c_{eq} = m_1 + \sum_{i=1}^{n} r_i d_i$, $c'_i = m_0 + r'_i(e_i - d_i + d'_i)$, where $r_i, r'_i \in_R \mathbb{Z}_N$
3. S sends $E_{PK}(c_{eq}), E_{PK}(c')$ to A and B in a random order, where $c' = \langle c'_1, c'_2, \ldots, c'_n \rangle$.
4. A and B decrypt the received messages and identify the correct message

Fig. 4. 1-out-of-2 COC scheme for "Equality" predicate: EQ-COC$_2^1$

4.1 COC$_2^1$ for "Equality" Predicate

Our COC$_2^1$ scheme for the equality predicate is naturally extended from the EQ-COC and INE-COC schemes. Intuitively, if $x = y$, A and B get m_1 by the EQ-COC scheme and, otherwise, they get m_0 by the INE-COC scheme. For better integration, we modify the EQ-COC scheme to use additively homomorphic encryption schemes. The scheme is shown in Figure 4. It is almost the same as the INE-COC scheme except that S sends an extra ciphertext c_{eq} to A and B.

Theorem 3. *The EQ-COC$_2^1$ scheme has the correctness property, unconditional sender's security, and computational receiver's security if the underlying homomorphic encryption scheme has semantic security.*

Proof. (sketch) We see that if $x = y$, all d_i's are equal to 0, and c_{eq} is equal to m_1. The opposite case holds by the same arguments in the proof of Theorem 2. This ensures the correctness property.

For sender's security, let $r = \sum_{i=1}^{n} r_i d_i$. Since $r_i \in_R \mathbb{Z}_N$, if $x \neq y$, there is a $d_i \neq 0$ such that r is uniformly distributed, and thus m_1 is unconditionally secure to A and B. If $x = y$, by the proof of Theorem 2, m_0 is unconditionally secure to A and B.

For receiver's security, S gets no information about x and y by the semantic security of the encryption scheme. For each of A and B, we can construct a simulator such that the adversary cannot distinguish the real view and the simulated view. The receiver's security holds. □

4.2 COC$_2^1$ for "Greater Than" Predicate

It is obvious that we can apply the GT-COC scheme twice to achieve a GT-COC$_2^1$ scheme. One invocation is for testing $x > y$ and the other one is for testing $x \leq y$. But, this approach costs twice as much as the GT-COC scheme. Our scheme for

- System parameters: n.
- Message sender S has messages: (m_0, m_1) and a key pair (PK_S, SK_S).
- Receiver A has a secret x, and receiver B has a secret y, where $|x| = |y| = n$.
- Receiver A and B have a common key pair (PK, SK), where $PK = (g, N)$.

1. A and B send $E_{PK_S}(E_{PK}(x_i))$ and $E_{PK_S}(E_{PK}(y_i))$ to S respectively, $1 \leq i \leq n$.
2. For each $i \in \{1, 2, \ldots, n\}$, S decrypts the received messages to get $E_{PK}(x_i)$ and $E_{PK}(y_i)$, and computes the following values via homomorphic encryption:
   (a) $d_i = x_i - y_i$, $d'_i = x_i + y_i - 1$
   (b) $e_i = r_i e_{i+1} + d_i$, $e'_i = r'_i d'_i$, where $e_{n+1} = 0$, $r_i, r'_i \in_R \mathbb{Z}_N$
   (c) $f_i = e_i + e'_i$
   (d) $c_i = \frac{m_1 - m_0}{2} f_i + \frac{m_1 + m_0}{2}$, $c_{eq} = m_0 + \sum_{i=1}^{n} r''_i d_i$, where $r''_i \in_R \mathbb{Z}_N$.
3. S sends $E_{PK}(c), E_{PK}(c_{eq})$ in a random order to A and B, where $c = \langle c_1, c_2, \ldots, c_n \rangle$.
4. A and B decrypt the received messages and identify the correct message.

Fig. 5. 1-out-of-2 COC scheme for "Greater Than" predicate: GT-COC$^1_2$

GT-COC$^1_2$ in Figure 5 is more efficient. It costs an extra ciphertext (for the case $x = y$) from S to A and B only.

Let l be the leftmost different bit between x and y. For $i < l$ and $i > l$, e_i and e'_i would be random values in $\mathbb{Z}_N$, respectively. When $i = l$, we have $e_i = d_i$ and $e'_i = 0$. Therefore, f_i is a random value when $i \neq l$ and $f_l = d_l$. If $x > y$, $f_l = 1$ and thus $c_l = m_1$; if $x < y$, $f_l = -1$ and thus $c_l = m_0$. For the case $x = y$, we use an extra value c_{eq} to embed m_0 like scheme EQ-COC$^1_2$.

Theorem 4. *The GT-COC$^1_2$ scheme has the correctness property, unconditional sender's security, and computational receiver's security if the underlying homomorphic encryption scheme has semantic security.*

Proof. (sketch) For correctness, consider the following three cases:

- $x > y$: let l be the index of the first different bit of x and y (from the most significant bit), we have $e_l = d_l = 1, e'_l = d'_l = 0$, and thus $f_l = e_l + e'_l = 1$. Therefore $c_l = \frac{m_1 - m_0}{2} \cdot 1 + \frac{m_1 + m_0}{2} = m_1$.
- $x < y$: similarly, since $f_l = e_l = d_l = -1$ in this case, we have $c_l = \frac{m_1 - m_0}{2} \cdot (-1) + \frac{m_1 + m_0}{2} = m_0$.
- $x = y$: by the same argument in the proof of Theorem 3, A and B get m_0 from c_{eq}.

For sender's security, we see that if $x \neq y$, then for all $i \neq l$, f_i is uniformly distributed in $\mathbb{Z}_N$. That is, all c_i's except c_l are uniformly distributed in $\mathbb{Z}_N$. For index l, according to the above argument, $c_l = m_0$ if $x < y$ and $c_l = m_1$ if $x > y$. Moreover, by the proof of Theorem 3, $c_{eq} = m_0$ if $x = y$, and c_{eq} is uniformly distributed if $x \neq y$. Therefore, m_0 is unconditionally secure to A and B if $x > y$, and m_1 is unconditionally secure to A and B if $x \leq y$.

For receiver's security, S gets no information about x and y by the semantic security of the encryption scheme. As in the previous proofs, for each of A and

B, we can construct a simulator such that the adversary cannot distinguish the real view and the simulated view. Therefore, the receiver's security holds. □

4.3 A General Transformation

We provide a general transformation from COC_2^1 to the second case mentioned in Section 1 for COC. We use the GT-COC_2^1 scheme as an example. The alternative model for COC is that when $x > y$, only A gets the message m and when $x \leq y$, only B gets the message. We modify our GT-COC_2^1 scheme to meet this requirement. In the beginning, A and B choose their own public/secret key pairs, namely, (PK_A, SK_A) and (PK_B, SK_B). Then S lets $m_1 = E_{PK_A}(m)$ and $m_0 = E_{PK_B}(m)$, and performs the scheme as usual. We see that if $x > y$, both A and B get $m_1 = E_{PK_A}(m)$. But, only A can decrypt it to get the message m. Similarly, if $x \leq y$, only B gets the message.

5 Extensions

In this section we introduce how to modify our COC schemes against malicious parties and collusion. We also discuss the construction of other predicates. The details of these modifications and extensions are left to the full version of this paper.

5.1 Schemes Secure Against Malicious Parties and Collusion

We can make our COC schemes secure against malicious parties and their collusion by using the threshold version of homomorphic cryptosystems. At the initial stage, each party gets a secret key share (from a dealer or a distributed key generation protocol). If the number of collusive parties does not exceed the threshold, they get nothing about the message. Since all parties (including the sender) exchange messages in encrypted form, all computation can be publicly verified. After the final result in encrypted form is obtained, all parties perform the threshold decryption for the result.

We need some non-interactive zero-knowledge proof systems for verification in the corresponding schemes (assuming PK is the common public key):

- Proof of plaintext knowledge. The prover proves that he knows the plaintext x for the encryption $E_{PK}(x)$ he created.
- Proof of one-bit plaintext. The prover proves that x is 0 or 1 for the encryption $E_{PK}(x)$ he created.
- Proof of correct exponentiation. Given (multiplicatively homomorphic) $E_{PK}(x)$, the prover outputs $E_{PK}(a)$ and $E_{PK}(x^a)$, and proves that $E_{PK}(x^a)$ is indeed the encryption of x^a.
- Proof of correct multiplication. Given (additively homomorphic) $E_{PK}(x)$, the prover outputs $E_{PK}(a)$ and $E_{PK}(ax)$, and proves that $E_{PK}(ax)$ is indeed the encryption of ax.

We can find such proof systems for the ElGamal and Paillier homomorphic encryption schemes [7, 19, 6, 9]. For the schemes INE-COC, EQ-COC$_2^1$ and GT-COC$_2^1$, the receivers need to prove that the encrypted messages they send are indeed the encryptions of 0 or 1. Boneh et al. [4] provide a verification gadget for this type of checking. Thus we can avoid using the proof system of one-bit plaintext.

5.2 Other Predicates

In addition to the basic predicates, we can design COC (COC$_2^1$) schemes for many other interesting predicates. For these predicates, the sender may need perform multiplication on two messages encrypted by an additively homomorphic encryption scheme. However, there is no known encryption scheme with both additive and multiplicative homomorphism properties. Fortunately, Boneh et al. [4] introduced an additively homomorphic encryption scheme which can perform multiplication on two ciphertexts one time. In the setting of using threshold cryptosystem, the sender can even perform multiplication on two ciphertexts arbitrary times via some interactions [9].

In fact, our COC can be designed for any predicate based on the evaluation of bivariable polynomial $f(x, y)$. For example, to compute a public polynomial $f(x, y) = a_2x^2y^2 + a_1x^2y + a_0y$, the receivers send the encryptions of x, x^2 and y, y^2 to the sender respectively. The sender then computes the polynomial by the following steps.

1. Perform the multiplication on the encrypted messages [4] such that $z_2 = x^2y^2$ and $z_1 = x^2y$.
2. Perform the constant multiplication: a_2z_2, a_1z_1 and a_0y.
3. Perform $f(x, y) = a_2z_2 + a_1z_1 + a_0y$.

After computing $f(x, y)$, the sender can embed messages into the result.

Alternatively, we can assume that one receiver holds the polynomial f and the other holds the secret x, and the sender embeds messages into the result of $f(x)$. For example, for the "membership" predicate, one receiver first encodes his set of secrets as a k-degree polynomial such that $f(x) = 0$ iff x belongs to the set, and the other receiver computes $x, x^2, \ldots, x^k$ for his secret x. The sender then sends the message to the receivers such that they get it iff $f(x) = 0$. This "membership" predicate can be used in our oblivious authenticated information retrieval application described in Section 1.

6 Conclusion

We introduce a new notion of *conditional oblivious cast*, which extends conditional oblivious transfer to the three-party case. The definitions of this notion are given. We also provide some implementations for some basic predicates such as "equality", "inequality", and "greater than" predicates. We believe this new notion will be an useful primitive of cryptographic protocols.

References

1. William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *Proceedings of Advances in Cryptology - EUROCRYPT '01*, volume 2045 of *LNCS*, pages 119–135. Springer-Verlag, 2001.
2. Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In *Proceedings of Advances in Cryptology - EUROCRYPT '94*, volume 950 of *LNCS*, pages 92–111. Springer-Verlag, 1994.
3. Ian F. Blake and Vladimir Kolesnikov. Strong conditional oblivious transfer and computing on intervals. In *Proceedings of Advances in Cryptology - ASIACRYPT '04*, volume 3329 of *LNCS*, pages 515–529. Springer-Verlag, 2004.
4. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In *Proceedings of the 2nd Theory of Cryptography Conference (TCC 2005)*, volume 3378 of *LNCS*, pages 325–341. Springer-Verlag, 2005.
5. Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In *Proceedings of Advances in Cryptology - CRYPTO '86*, volume 263 of *LNCS*, pages 234–238. Springer-Verlag, 1986.
6. Jan Camenisch and Markus Stadler. Proof systems for general statements about discrete logarithms. Technical Report 260, Institute for Theoretical Computer Science, ETH Zurich, Mar 1997.
7. David Chaum, Jan-Hendrik Evertse, Jeroen van de Graaf, and Rene Peralta. Demonstrating possession of a discrete logarithm without revealing it. In *Proceedings of Advances in Cryptology - CRYPTO '86*, volume 263 of *LNCS*, pages 200–212. Springer-Verlag, 1986.
8. Cheng-Kang Chu and Wen-Guey Tzeng. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In *Proceedings of the Public Key Cryptography (PKC '05)*, volume 3386 of *LNCS*, pages 172–183. Springer-Verlag, 2005.
9. Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. Multiparty computation from threshold homomorphic encryption. In *Proceedings of Advances in Cryptology - EUROCRYPT '01*, volume 2045 of *LNCS*, pages 280–299. Springer-Verlag, 2001.
10. Giovanni Di Crescenzo, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Conditional oblivious transfer and timed-release encryption. In *Proceedings of Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *LNCS*, pages 74–89. Springer-Verlag, 1999.
11. Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.
12. Matthias Fitzi, Juan A. Garay, Ueli Maurer, and Rafail Ostrovsky. Minimal complete primitives for secure multi-party computation. In *Proceedings of Advances in Cryptology - CRYPTO '01*, volume 2139 of *LNCS*, pages 80–100. Springer-Verlag, 2001.
13. Hsiao-Ying Lin and Wen-Guey Tzeng. An efficient solution to the millionaires' problem based on homomorphic encryption. In *Proceedings of Applied Cryptography and Network Security 2005 (ACNS '05)*, volume 3531 of *LNCS*, pages 456–466. Springer-Verlag, 2005.
14. Yi Mu, Junqi Zhang, and Vijay Varadharajan. m out of n oblivious transfer. In *Proceedings of the 7th Australasian Conference on Information Security and Privacy (ACISP '02)*, volume 2384 of *LNCS*, pages 395–405. Springer-Verlag, 2002.
15. Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing (STOC '99)*, pages 245–254. ACM, 1999.

16. Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the 12th Annual Symposium on Discrete Algorithms (SODA '01)*, pages 448–457. ACM/SIAM, 2001.
17. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *LNCS*, pages 223–238. Springer-Verlag, 1999.
18. Michael O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.
19. Claus Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
20. Yiannis Tsiounis and Moti Yung. On the security of ElGamal based encryption. In *Proceedings of the Public-Key Cryptography (PKC '98)*, volume 1431 of *LNCS*, pages 117–134. Springer-Verlag, 1998.
21. Wen-Guey Tzeng. Efficient 1-out-n oblivious transfer schemes. In *Proceedings of the Public-Key Cryptography (PKC '02)*, pages 159–171. Springer-Verlag, 2002.