Rational Oblivious Transfer

Xiong Fan
xfan@cs.umd.edu

Kartik Nayak
kartik1507@gmail.com

May 14, 2014

Abstract

Oblivious transfer is widely used in secure multiparty computation. In this paper, we propose a game theoretic analysis of this primitive and discuss how a protocol for efficient rational oblivious transfer should be in the real world.

1 Introduction

1-out-of-2 was first suggested by Even et al. [6] as a generalization of Rabin's oblivious transfer [15], who built the first OT based on RSA assumption. Even et al. built 1-out-of-2 OT based on arbitrary public key encryption instead of RSA assumption. The 1-out-of-N OT was then introduced by Brassard et al. [3] under the name ANDOS (all or nothing disclosure of secrets). This notion is widely used in multiparty computation. Here are two famous applications below.

Yao's Garbled Circuits Protocol Yao's garbled circuits [18] allows two parties to securely compute an arbitrary function if it can be expressed as a Boolean circuit. First the sender use symmetric keys to encrypt the Boolean gates and sends the encrypted function together with the keys corresponding to his inputs to the receiver. The receiver then uses a 1-out-of-2 OT to obliviously obtain the keys that correspond to his inputs and evaluates the encrypted function gate by gate. Then either the receiver sends the output to the sender or the sender reveals the mapping from keys to output bits will get the result.

The GMW Approach The protocol proposed by Goldreich, Micali and Wigderson (GMW) [8] also achieves almost the same result as Yao's approach but use different techniques. They also represent the function to be computed as Boolean circuit. Then both parties use the XOR operation to secretly share their inputs. An XOR gate is evaluated by just locally XORing the share while an AND gate is evaluated with the help of 1-out-of-2 OT on bits. To reconstruct the outputs, both parties exchange their output shares. The performance of GMW highly depends on the number of OTs and the depth of evaluated circuits since the computation of AND gates require OT evaluation.

There exist two kinds of OT extensions. One is to use the existent 1-out-of-2 to construct 1-out-of-N OT or even k-out-of-N OT. For instance, the 1-out-of-N OT constructions of [4] and [5] need N calls to the 1-out-of-2 OT protocol, but in [14], they just need $\log N$ calls to the 1-out-of-2 OT protocol plus $O(N)$ evaluations of a pseudo-random function. Moreover, in [13] they implement 1-out-of-N OT with an amortized overhead of a single 1-out-of-2 OT. The other kind is to use OT for short strings to get OT for long strings [11].

Rational Cryptography. We used to consider the worst-case scenario, i.e., we allow the adversary cannot deviate from the specified protocol arbitrarily, even without a reason. While this approach yields strong security guarantees, it has been criticized as being overly pessimistic since it neglects the *incentives* that lead parties to deviate from their prescribed behavior; this may result in protocols designed to defend against highly unlikely attacks. Motivated by this, a recent line of work on "rational cryptography" has focused on using ideas from game theory to analyze cryptographic protocols run by a set of rational parties [9, 2, 10, 1, 7]. There, parties are no longer viewed as being "good" (semi-honest) or "bad" (malicious); all parties are simply *rational*, motivated by some utility function. The goal of this line of work is to design protocols for which following the protocol is a game-theoretic equilibrium for the parties. It has been shown that by incorporating incentives one can circumvent impossibility results [1, 7], or design protocols with better efficiency.

RSA based 1-out-of-2 oblivious transfer protocol. This was proposed by Even et al. [6]. This scheme uses RSA encryption to obtain a 1-out-of-2 oblivious transfer between the sender and the receiver. The protocol is as follows:

1. The sender has two messages m_0 and m_1. and the receiver has a choice bit σ.
2. The sender generates an RSA key pair where d is the secret key, e is the public key and the corresponding value of N.
3. The sender sends N, e and two random keys r_0 and r_1 to the receiver.
4. The receiver computes $v = (m_\sigma + k^\epsilon) \mod N$ and sends it to the sender.
5. The sender now computes $k_0 = (v - m_0)^d \mod N$ and $k_1 = (v - m_1)^d \mod N$, adds m_0 and m_1 to these values respectively and sends the corresponding values m'_0 and m'_1 to the receiver.
6. The receiver can obtain $m_\sigma = m'_\sigma - k_\sigma$.

2 Preliminaries

In this section, we present the oblivious transfer primitive from the perspective of game theory and also some game theory basics. For completeness, we include the definition of oblivious transfer in the appendix.

Notations. We say a function $\mu : \mathbb{N} \leftarrow \mathbb{R}$ is called *negligible*, denoted as $\mathsf{negl}(\cdot)$, if for any polynomial p, there exists a value $N \in \mathbb{N}$ such that for any $n > N$, we have $\mu(n) < 1/p(n)$. We say that two distributions $\mathcal{X}$ and $\mathcal{Y}$ are *computationally indistinguishable*, denoted as $\mathcal{X} \approx \mathcal{Y}$, if for every PPT distinguishier D, it holds that

$$|\mathbf{Pr}[D(\mathcal{X}) = 1] - \mathbf{Pr}[D(\mathcal{Y}) = a]| \leq \mathsf{negl}(\lambda)$$

where λ is the security parameter.

2.1 Game Theoretic Definitions

We review some the relevant concepts from Game Theory, and the extensions needed to put these concepts in the oblivious transfer model. Traditionally, a two-player game $\Gamma = (\{A_0, A_1\}, \{u_0, u_1\})$ is determined by specifying, for each player P_i, a set A_i of possible actions and a utility function $u_i : A_0 \times A_1 \to \mathbb{R}$. Let $A = A_0 \times A_1$ denote the outcome of the game. The utility function u_i of player P_i expresses this player's preferences over outcome $a = (a_0, a_1) \in A$. We say that player P_i prefers outcome a to outcome a' if and only if $u_i(a) > u_i(a')$. A strategy s_i for player P_i is a distribution on the action set A_i. Given a strategy vector $s = (s_1, s_2)$, we let $u_i(s)$ be the expected utility of P_i with respect to the event that the other party plays according to s. We continue with a definition of Nash equilibrium:

Definition 2.1. *Let $\Gamma = (\{A_0, A_1\}, \{u_0, u_1\})$ be a game defined as above, and $s = (s_0, s_1)$ be the strategy vector as above. Then s is a Nash equilibrium if for all i and any strategy s', we have*

$$u_i(s''_0, s''_1) \leq u_i(s)$$

where $s''_i = s'_i$, and $s''_{1-i} = s_{1-i}$.

The above formalism is also naturally extended to the case of *extensive form* games, where the parties take turns when taking actions. Another natural extension is to games with *incomplete information*.

Extensions for cryptographic model. There exist three kinds of players in cryptographic world. One is *semi-honest* where the player follows the protocol but is curious about the privacy of other players. One is *malicious* where the places can deviate from the specified protocol arbitrarily. Another one is *rational* where the players may deviate from the protocol only for a high utility. Hence, we have to consider different types of players in the scenario, which directs us to *Bayesian game*. We will formalize oblivious transfer using a Bayesian game.

Definition 2.2. *The game is defined as $\Gamma = (N, \{A_i, u_i, T_i\}_{i \in N})$, where:*

- *N is the set of players, $N = \{P_0(sender), P_1(receiver)\}$.*
- *A_i is the action set for player P_i.*
- *T_i is the type of player P_i, $T_i = \{s, m, r\}$, which stands for semi-honest, malicious and rational correspondingly.*
- *$u_i : A_i \to \mathbb{R}$ is utility function for player P_i.*

Since we consider rationality in oblivious transfer protocol, so the type for each player P_i should be r. A *Bayesian Nash equilibrium* is defined as a strategy profile specified for each player about the type of the other player that maximizes the expected utility for each player given their beliefs about the other players' types, and the strategy played by the other player.

3 Cryptographic Perspective Oblivious Transfer

Execution in the ideal world. The ideal world includes a trusted third party. An ideal oblivious transfer proceeds as follows:

Input: The sender S obtains an input pair (m_0, m_1) with $|m_0| = |m_1|$, and the receiver obtains a selection bit $\sigma \in \{0,1\}$.

Send input to trusted party: An honest party always sends its input unchanged to the trusted party. A malicious party may either abort, in which case it sends $\perp$ to the trusted party, or send some other inputs to the trusted party.

Trusted party computes output: If the trusted party receives $\perp$ from one of the parties, then it sends $\perp$ to both parties and halts. Otherwise, upon receiving messages (m'_1, m'_0) from sender and a bit σ' from receiver, the trusted party sends $m'_{\sigma'}$ to receiver R and halts.

Outputs: An honest party always outputs the message it has obtained from the trusted party, while a malicious party may output a function of its initial inputs and the message obtained from the trusted third party.

Denote f the oblivious transfer functionality and let (S, R) be a pair of non-uniform PPT algorithms. Then the joint execution of f under algorithms (S, R) in the ideal model, denoted as $\text{IDEAL}_{f,(S,R)}((m_0, m_1), \sigma)$, is defined as the output pair of S and R in the above ideal execution.

Execution in the real world. We next consider the real model in which a real two-party protocol is executed and there exists no trusted third party. In this case, a malicious player may follow any arbitrary feasible strategy, which can be implemented by a PPT algorithm. Let π be a protocol between sender and receiver, and (S, R) be the corresponding non-uniform PPT algorithms adopted by the parties. Then the joint execution of π under (S, R) in the real model, denoted as $\text{REAL}_{\pi,(S,R)}((m_0, m_1), \sigma)$, is defined as the output pair of (S, R) resulting from the protocol execution.

Having defined the ideal and real models, we can now define security of protocols. We follow the ideal/real world paradigm in secure computation. Loosely speaking, the definition asserts that a secure two-party protocol (in the real model) emulates the ideal model (in which a trusted party exists).

Definition 3.1. *Let f denote an oblivious transfer protocol in the ideal world, and π denote an oblivious transfer protocol in the real world. Protocol π is said to be a secure oblivious transfer protocol if for every pair of non-uniform* PPT *algorithms (S, R) in the real world, there exist a pair of non-uniform* PPT *algorithms (S', R') in the ideal world, such that for every $m_0, m_1 \in \{0,1\}^*$ of the same length and every $\sigma \in \{0,1\}$, it holds that*

$$\text{IDEAL}_{f,(S',R')}((m_0, m_1), \sigma) \approx \text{REAL}_{\pi,(S,R)}((m_0, m_1), \sigma)$$

References

[1] Gilad Asharov, Ran Canetti, and Carmit Hazay. Towards a game-theoretic view of secure computation. In Kenneth G. Paterson, editor, Advances in Cryptology-EUROCRYPT 2011.

[2] Ittai Abraham, Danny Dolev, Rica Gonen, and Joseph Y. Halpern. Distributed com- puting meets game theory: robust mechanisms for rational secret sharing and multi- party computation. ACM PODC 2006.

[3] Brassard, Gilles, Claude Crpeau, and Jean-Marc Robert. "All-or-nothing disclosure of secrets." Advances in Cryptology-CRYPTO 1986. Springer Berlin Heidelberg, 1987.

[4] Brassard, Gilles, Claude Crepeau, and Jean-Marc Robert. "Information theoretic reductions among disclosure problems." Foundations of Computer Science, 1986., 27th Annual Symposium on. IEEE, 1986.

[5] Brassard, Gilles, Claude Crpeau, and Miklos Santha. "Oblivious transfers and intersecting codes." Information Theory, IEEE Transactions on 42.6 (1996): 1769-1780.

[6] Even, Shimon, Oded Goldreich, and Abraham Lempel. "A randomized protocol for signing contracts." Communications of the ACM 28.6 (1985): 637-647.

[7] Adam Groce and Jonathan Katz. Fair computation with rational players. In Advances in Cryptology-EUROCRYPT 2012, July 2012.

[8] Goldreich, Oded, Silvio Micali, and Avi Wigderson. "How to play any mental game." Proceedings of the nineteenth annual ACM symposium on Theory of computing. ACM, 1987.

[9] Joseph Y. Halpern and Vanessa Teague. Rational secret sharing and multiparty compu- tation: Extended abstract. ACM STOC 2004.

[10] Joseph Y. Halpern and Rafael Pass. Game theory with costly computation: Formula- tion and application to protocol security. ICS 2010.

[11] Ishai, Yuval, et al. "Extending oblivious transfers efficiently." Advances in Cryptology-CRYPTO 2003. Springer Berlin Heidelberg, 2003. 145-161.

[12] Kreuter, Benjamin, Abhi Shelat, and Chih-Hao Shen. "Billion-gate secure computation with malicious adversaries." Proceedings of the 21st USENIX conference on Security symposium. USENIX Association, 2012.

[13] Naor Moni, and Benny Pinkas. "Efficient oblivious transfer protocols." Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2001.

[14] Naor Moni, and Benny Pinkas. "Computationally secure oblivious transfer." Journal of Cryptology 18.1 (2005): 1-35.

[15] Rabin, Michael O. "How to Exchange Secrets with Oblivious Transfer." (1981).

[16] Rogaway, Phillip, and John Steinberger. "Constructing cryptographic hash functions from fixed-key blockciphers." Advances in Cryptology-CRYPTO 2008. Springer Berlin Heidelberg, 2008. 433-450.

[17] Shannon, Claude E. "Communication theory of secrecy systems." Bell system technical journal 28.4 (1949): 656-715.

[18] Yao, Andrew Chi-Chih. "How to generate and exchange secrets." Foundations of Computer Science, 1986. 27th Annual Symposium on. IEEE, 1986.