Oblivious Transfer from Noisy Channel

Project Report for CS6501 Cryptography 2014 Fall

Longze Chen

Dec. 2014

Abstract

Oblivious Transfer (OT) is one of the most fundamental primitives in cryptographic protocol design. It was first implemented in the cryptographic scenario by Rabin back in 1981, based on the assumption that factoring is hard. However, cryptographers were not satisfied with intractability assumptions and attempted to find weaker alternatives. In 1988, Crepeau and Killian removed this assumtpion by using the noisy channel. In this report, I present a brief review on such OT. Specifically, I show how to reduce OT to noisy channel.

1 Introduction

Oblivious Transfer is the most fundamental primitive in cryptographic protocol design, especially for Secure Two-party Communication introducecd by Yao [8]. Rabin [6] first implemented OT for the cryptography world in 1981. This implementation is based on one intractability assumption that factoring is hard, which can be further reduced to the existence of trap door permutation [5].

Yao [9] first reduced Oblivious Circuits Evaluation to 1-2-OT, but with the assumption of intractability of factoring. Kilian [10] and Goldreich-Vainish [11] further improved Yao's work by removing this assumption. Crepeau [1] also provided an alternative protocol which is proved to be equivalent to 1-2-OT.

In the same year, motivated by the fact that protocols previous had relied on intractability assumptions were then solved without them, Crepeau and Kilian [2] showed that we can achieve OT from weaker forms of OT, and further reduced OT to noisy channel. In 1999, Damgard et al. [3] introduced a even weaker form of noisy channel called unfair noisy channel, which was further improved in [4] and [7].

This report mainly focuses on explaining the possibility of implementing OT using a noisy communication channel, which requires several steps of reduction. The high level idea is to simulate a "bugged" noiseless channel using a noisy one,

where the "bugged" channel can be viewed as a variant of some weaker form of OT which can be reduced from 1-2-OT. I start by define proper weaker OTs, and provide reduction from 1-2-OT to them in Section 2. Then we apply these reduction to the simulated channel with care in Section 3. The proof primarily comes from Crepean and Kilian's 1988 paper [2].

2 OT from Weaker OT

2.1 1-2 Oblivious Transfer (1-2-OT)

In 1-2-OT: Alice (Sender) has two secret bits x_0 and x_1. Bob (Receiver) has the selection bit b. After running the protocol, Bob will learn x_b, Bob can correctly guess x_{1-b} with probability $\frac{1}{2}$, and Alice can correctly guess b with probability $\frac{1}{2}$.

2.2 Alternative Oblivious Transfer (A-OT)

There is an alternative definition of OT which is almost equivalent to 1-2-OT [1], denoted by A-OT: Alice has a secret bit x, Bob has no inputs and knows nothing of x. After running the protocol, either of the two events happens with probability $\frac{1}{2}$. 1. Bob learns the value of x. 2 Bob learns nothing. Furthermore, he knows which is the case, and Alice learns nothing.

2.3 α-1-2 Oblivious Transfer (α-1-2-OT)

In α-1-2-OT ($\frac{1}{2} \leq \alpha < 1$): Alice has two secret bits x_0 and x_1. Bob has the selection bit b. After running the protocol, Bob will learn x_b, Bob can correctly guess x_{1-b} with probability $\frac{1}{2}$, and Alice can correctly guess b with probability at most α. This is weaker than 1-2 OT because the sender has more power.

2.3.1 Reduction from 1-2-OT to α-1-2-OT

Alice has x_0 and x_1 and Bob has b. They run α-1-2-OT n times to simulate 1-2-OT.

1. Alice chooses $2n$ bits, where:

$$r_i^0 = \begin{cases} \text{randomly sampled from } \{0,1\} & \text{if } 1 \leq i \leq n-1 \\ x_0 \oplus (\oplus_{i=1}^{n-1} r_i^0) & \text{if } i = n \end{cases} \quad (1)$$

$$r_i^1 = r_i^0 \oplus x_0 \oplus x_1, \text{ for all } i \in [1, n] \quad (2)$$

2. Bob chooses n bits c_i, where $\oplus_{i=1}^{n} c_i = b$.
3. Alice and Bob run the α-1-2-OT n times, each time Alice inputs (r_i^0, r_i^1) and Bod inputs c_i.

4. Bob computes $x_b = \oplus_{i=1}^{n} r_i^{c_i}$.

It is easy to know that Alice can correctly guess b with at most $\frac{1}{2} + \alpha^n$, which is exponentially close to 1-2-OT.

2.4 α Alternative Oblivious Transfer (α-A-OT)

In α-A-OT ($\frac{1}{2} \leq \alpha < 1$): Alice has a secret bit x, Bob has no inputs and he knows nothing about x. After running the protocol, either of the two events happens with probability p and $1-p$ respectively. 1. Bob learns the value of x. 2. Bob learns nothing. Furthermore, he knows which is the case, and Alice learns nothing. If Alice is honest, $p = \frac{1}{2}$; otherwise, she can choose p where $1 - \alpha \leq p \leq \alpha$.

2.4.1 Reduction from α-1-2-OT to α-A-OT

Lets assume both Alice and Bob follows the protocol. Alice has x_0 and x_1 and Bob has b. They run α-A-OT k times to simulate α-1-2-OT.

1. Alice randomly chooses k bits $C_1, C_2, \cdots, C_k$. She send them to Bob through the α-A-OT channel by run the protocol k times.
2. Bob randomly choose two indexes (i_0, i_1), where $i_0 \in \{i|\text{Bob receives } C_i\}$ and $i_1 \in \{i|\text{Bob does not receive } C_i\}$. He sends (i_b, i_{1-b}) to Alice through a clean channel.
3. Alice returns $x_0 \oplus C_{i_b}$ and $x_1 \oplus C_{i_{1-b}}$. Bob will reconstruct x_b but has no information about x_{1-b}.

Since Alice never knows which bit Bob receives according to the protocol, as long as k is large Bob will be able to find proper (i_b, i_{1-b}), this is a perfect simulation of α-1-2-OT.

3 OT from Standard Noisy Channel

3.1 Noisy Channel Transfer (NCT)

In communication using a noisy channel: when Alice sends a bit x to Bob, x is flipped with probability ρ. Noisy Channel Transfer is a simulation of such noisy communication: Alice has a secret bit x, and Bob knows nothing about x. After running the protocol, Bob receives $x^{'}$, with probability ρ ($x^{'} = x$) he learns x ; with $1-\rho$, he learns nothing (wrong value). We expect $\frac{1}{2} < \rho < 1$ for the channel to be useful.

3.2 Simulate α-A-OT-dirty with NCT

In order for Bob to be able to tell whether he received the correct bit, Alice send each bit twice through the NCT. This can be viewed as an variant of α-A-OT. Alice tries to send bit x. If Bob receives 00 or 11, we say that he receives the correct x; otherwise, we say he fails to receive the bit. Obviously, Bob knows which event happens at the end of the protocol. The only difference compared with α-A-OT is that in this protocol, Bob is only ρ^2 certain of the value he received. We denote this transfer as α-A-OT-dirty.

3.3 Reduction from α-A-OT-dirty to α-1-2-OT

When applying the same reduction used in Section 2.4.1, we need to change slightly to cater for the difference.

Suppose both Alice and Bob follow the protocol. Alice has x_0, x_1 and Bob has b. They run α-A-OT-dirty k^c times to simulate α-1-2-OT, where c is a constant.

1. Alice randomly chooses k^c bits $C_1, C_2, \cdots, C_{k^c}$. She send them to Bob through the α-A-OT-dirty channel by running the protocol k^c times.
2. Bob receives $C_1^{'}, C_2^{'}, \cdots, C_{k^c}^{'}$. (If B receives 00, let $C_i^{'} = 0$; if B receives 11, let $C_i^{'} = 1$; and if B receives 01 or 10, let $C_i^{'} = \perp$.) Instead of picking two indexes as before, Bob randomly chooses two sets of indexes:

$$I_s : |I_s| = k, \text{and for all } i \in I_s, C_i^{'} \in \{0, 1\} \tag{3}$$

$$I_{1-s} : |I_{1-s}| = k, \text{and for all } i \in I_s, C_i^{'} = \perp \tag{4}$$

   , and sends I_s, I_{1-s} to Alice.
3. Alice returns two sets: $W_0 : \{w_0 | w_0 = x_0 \oplus C_{i_0^j}, j \in [1, k]\}$ and $W_1 : \{w_1 | w_1 = x_1 \oplus C_{i_1^j}, j \in [1, k]\}$. Bob computes $X_s = \{x_s | x_s = C_{i_s^j}^{'} \oplus w_s^j, j \in [1, k]\}$ and guesses x_s to be the majority of X_s.

It is easy to see that Bob has very high probability to have the right guess given k is large. Same as Section 2.4.1, Bob has no information about x_{1-s} and Alice has no idea of s.

4 Further Reading

The first issue is what if Alice and Bob are malicious. I am not quite comfortable with the proof sketch provide by [2] and need more time to understand the full proof.

The second issue is this reduction requires that both party know the noise level and it is also fixed. To solve this problem, Damgard et al. [3] introduced an unfair noisy channel (UNC) and weak oblivious transfer (WOT). I plan to read this paper in detail and its improvement version [4] as well.

5 Conclusion

Early OT implementations, including Rabin-OT and Yao's 1-2-OT, are based on the assumption of intractability of factoring. Crepeau and Kilian removed this assumption by using a noisy channel which had been well studied in the area information theory. I go over the reduction from 1-2-OT to standard noisy channel in semi-honest mode in this report and plan to further study the protocol in malicious mode as well as using weak noisy channel.

References

[1] Crepeau, C. "Equivalence between two flavors of oblivious transfer", Crypto 87

[2] Crepeau, C. and Killian, J. "Achieving oblivious transfer using weakened security assumptions", FOCS 88

[3] Damard, I. Kilian, J. and Salvail, L. "On the (im)possibility of basing oblivious transfers and bit commitment on weakened security assumptions" Eurocrypt 99

[4] Damard, I. Fehr, S, Morozov, K and Salvail, L. "Unfair noisy channels and oblivious transfer", TCC 04

[5] Even, S. Goldreich, O. and Lampel, A. "A randomized protocol for signing contracts", CACM 85

[6] Rabin, M. "How to exchange secrets by oblivious transfer", TR-81, Harvard

[7] Wul, J. "Oblivious transfer from weak noisy channels", TCC 09

[8] Yao, A. "Protocols for secure computations", FOCS 82

[9] Yao, A. "How to generate and exchange secrets", FOCS 86

[10] Killian, J. "Founding cryptography on oblivious transfer", STOC 88

[11] Goldreich, O. and Vainish, R. "How to solve any protocol problem - an efficiency improvement", Crypto 87