Unconditionally Secure Oblivious Transfer Based on Channel Delays

Kai-Yuen Cheong and Atsuko Miyaji

Japan Advanced Institute of Science and Technology,
1-1 Asahidai, Nomi, Ishikawa, 923-1292 Japan
{kaiyuen, miyaji}@jaist.ac.jp

Abstract. Without the use of computational assumptions, unconditionally secure oblivious transfer (OT) is impossible in the standard model where the parties are using a clear channel. Such impossibilities can be overcome by using a noisy channel. Recently, Palmieri and Pereira proposed a protocol based on random channel delays only. Their scheme is secure in the semi-honest model, but not in the general malicious model. In this paper we study oblivious transfer in the same setting but we improve the result to obtain a fully secure protocol in the malicious model.

Keywords: oblivious transfer, unconditional security, channel delay

1 Introduction

Oblivious Transfer (OT) is a two-party cryptographic protocol with a simple function. However, it is an important primitive because any secure computation can be based on OT [7, 11]. It is considered a universal primitive for cryptographic functionalities where the users do not fully trust each other.

In the first OT system introduced by Rabin [9], a message is received with probability 1/2 and the sender does not know whether the message reaches the other side. Even et al. defined the 1-out-of-2 OT [6], where the sender has two secrets and the receiver can choose one of them in an oblivious manner. That is, the sender cannot know the receiver's choice and the receiver cannot know more than one of the sender's secrets. The 1-out-of-2 OT is equivalent to the Rabin OT [1]. Also, we may assume that the sender's secrets are one-bit messages in 1-out-of-2 OT, because the case with string messages is reducible to it efficiently [5].

By simple arguments, it can be shown that OT cannot achieve information theoretic security for both parties over a standard, noiseless communication channel. If a noisy channel of certain form is available between the sender and the receiver, OT can be constructed with unconditional security [2, 4, 10]. While OT based on noisy channels with transmission errors are relatively well-studied, in 2010 Palmieri and Pereira [8] proposed a new scheme using a completely different source of channel randomness. In their paper, the channel does not have

any error on the content but a transmission delay. This seems to be a remarkably weaker assumption than the noisy channel, and random channel delays are abundant in media like the Internet.

In [8], a semi-honest OT protocol is proposed, where the parties follow the protocol strictly, but may try to compute more information afterwards from the communication transcript. In this paper, we improve it by removing the semi-honest assumption. The final protocol is unconditionally secure. Some techniques we use are from the standard ones [3] for general OT enhancement.

Our paper is organized as follows: Section 2 describes the preliminaries including the assumptions about the channel used and the definition of OT. In Section 3 we describe a basic protocol which only works in the semi-honest model, similar to the protocol in [8]. The main contribution of this paper is shown in Section 4 where we provided the fully secure protocol, before the conclusion in Section 5.

2 Preliminaries

2.1 Delay Channel

Following the tracks of [8], we first define the properties of the channel called the binary discrete-time delaying channel (BDDC). In this model, the channel accepts binary strings called packets and delivers them with some delay. It is a memoryless channel such that delays happen to each packet independently according to certain known probabilistic distribution. The channel operates at discrete times, such that there is a fixed set of allowed time for transmitting and receiving the packets. A packet always arrives as a whole at the same time, without being broken into parts. The channel has no other forms of errors.

A delay probability is denoted by p with $p < 1/2$. We assume that p is publicly known, and set $q = 1 - p$. Neither the sender nor the receiver gets any feedback information about the delay that occurred. The BDDC has the following properties:

1. There is a discrete (either finite or infinite) set of allowed input times $T = \{t_0, t_1, ...\}$ and output times $U = \{u_0, u_1, ...\}$.
2. A packet sent at t_i will arrive at u_i if there is no delay. Otherwise it will arrive at u_j with probability p_{ij}. A packet may delay once with probability p, and is subject to further delays with the same probability. Therefore, for $j \geq i$ we have $p_{ij} = p^{j-i} - p^{j-i+1}$. For $j < i$, $p_{ij} = 0$.

It is clear that $p_{ii} = q$ denotes the probability that the packet arrives on time. The assumption that $p < 1/2$ can generally be justified. First, p should be low for an efficient channel. Also, in reality, time is an analog quantity, and the mean and variance of the actual delay can be used to derive the interval in which the packet is expected to arrive with high probability. In [8], it is also assumed that $p < 1/2$.

2.2 Oblivious Transfer

In this paper, the 1-out-of-2 bit OT with perfect security is defined as:

1. The sender Alice inputs a pair of secret bits (s_0, s_1) and the receiver Bob inputs a choice bit c.
2. Correctness: If both Alice and Bob are honest, Bob outputs s_c and Alice outputs nothing.
3. Security for Alice: Regardless of Bob's actions, if Alice is honest, there exists $c' \in \{0, 1\}$ such that Bob receives zero information of $s_{c'}$.
4. Security for Bob: If Bob is honest, Alice receives zero information on c regardless of her actions.

In the case where the properties above are not perfectly satisfied, if the failure probability for each of them is negligible, we say that the OT protocol is unconditionally secure. A protocol is said to be in the semi-honest model if all parties are assumed to follow the protocol. Otherwise it is in the malicious model, where the cheating party does not need to follow the protocol. Therefore, security in the malicious model is strictly stronger than security in semi-honest model.

3 A building block protocol

3.1 Semi-honest OT

In this part we introduce a protocol similar to [8]. It is a semi-honest OT protocol. A small modification is made to reduce the communication cost. Impact to security is basically none but the analysis becomes easier in our version. The change is that we set the variables e_i to be one bit, rather than a general binary string. Our version of the protocol is:

1. For security parameter n, Alice prepares random bits $e_1, e_2, ...e_n$. For convenience n is an even number. For $1 \leq i \leq n$ she prepares $v_i = i||e_i$ which is the string created by the concatenation of index i and bit e_i. Next, she also creates $v_i' = i||(1 - e_i)$.
2. At time t_0, Alice sends all v_i to the BDDC. At time t_1, she sends all v_i'. Each of v_i and v_i' is treated as one packet in the BDDC channel.
3. At time u_0, Bob receives the packets coming from the BDDC. If fewer than $n/2$ packets are received, Bob aborts the protocol.
4. Otherwise Bob randomly selects a set of indices $I_c \subset \{1, 2, \ldots n\}$, where c is his OT choice bit, under the condition that $|I_c| = n/2$ and Bob has received a string in the form $i||*$ at u_0 for all $i \in I_c$. He sets I_{1-c} to be the set of all $i \in \{1, 2, \ldots n\}$ such that $i \notin I_c$.
5. Bob sends (I_0, I_1) over a clear channel to Alice. If there are no other channels the BDDC can also be used for this purpose. In either case we do not add extra assumptions about the properties of the channels used in the protocol.

6. Alice computes

$$\beta_0 = \bigoplus_{i \in I_0} e_i$$
$$\beta_1 = \bigoplus_{i \in I_1} e_i \quad (1)$$

   and then sets $\sigma_0 = s_0 \oplus \beta_0$ and $\sigma_1 = s_1 \oplus \beta_1$.
7. Alice sends (σ_0, σ_1) to Bob.
8. Bob knows e_i whenever $i \in I_c$. He computes $\beta_c = \bigoplus_{i \in I_c} e_i$ and finally $s_c = \sigma_c \oplus \beta_c$.

In essence, in the protocol v_i is a random message Alice sends to Bob and v_i' is for confusion such that Bob will not be able to get Alice's message if he gets v_i and v_i' at the same time. Thus the setting has the feature of Rabin OT [1]. It is then used to construct the 1-out-of-2 OT in the standard way.

3.2 Security in the semi-honest model

Following the definition of OT, the proof of security properties and functionality is divided into three parts, correctness, security for Alice, and security for Bob.

Correctness: If both parties are honest, the packets sent at t_0 follows a binomial distribution regarding to whether they are delayed or not. Failure happens when there are not enough packets received at u_0. Same as [8] we use the Hoeffding's inequality to see that the upper bound of failure probability is $e^{-2n(\frac{1}{2}-p)^2}$. As it decreases exponentially with the increase of n, we can say that it is negligible.

Alice's security: We can show that the protocol is secure even against a malicious Bob, which is stronger than the semi-honest Bob. To simplify the analysis further, let us assume that the malicious Bob is equipped with a special power: whenever a packet is received at or after u_2, Bob can tell the time that the packet in question is sent. Therefore, the only uncertainty is on the packets received at u_1, which may be sent at t_0 or t_1. Note that the real Bob has no such power and is thus strictly weaker.

For any i, if both v_i and v_i' are received at u_1, they are indistinguishable to Bob, and he would have zero information on e_i. For one i, the probability for this to happen is pq^2. Therefore, the probability that it never happens for $1 \leq i \leq n$ is $(1 - pq^2)^n$. Note that this probability falls exponentially with n.

Otherwise, there exists at least one i such that Bob has zero information on e_i. Since either $i \in I_0$ or $i \in I_1$, it is ensured that there exists c' such that Bob cannot get $s_{c'}$.

Bob's security: If Alice follows the protocol, it is clear that she cannot get any information about c because all possible sets of (I_0, I_1) are equally likely. Therefore the protocol is perfectly secure against the semi-honest Alice.

4 Constructing the full protocol

4.1 Insecurity of the basic protocol

The problem with the semi-honest protocol is that when Alice is malicious, it is insecure. Note that Alice will not send the same packet twice or send something with incorrect format, because they are detectable with absolute certainty. On the other hand, the honest Bob does not need to look at packets arriving at u_1 or later in the protocol, so Alice also does not need to care about what she sends to the BDDC at time t_1 and after. Moreover, Alice cannot gain any information about c after the last message from Bob. Therefore it is clear that the malicious Alice will only focus on adding or deleting messages to be sent at t_0 in the following manner. For each i, essentially there are only three possible deviations from the protocol:

1. Alice sends v_i' instead of v_i at t_0.
2. Alice sends both v_i and v_i' at t_0.
3. Alice sends neither v_i nor v_i' at t_0.

The first case cannot cause any harm to Bob as it is equivalent to flipping the randomly chosen e_i before the protocol begins. For the second case, Bob can detect it with probability q^2 by seeing both v_i and v_i' at u_0. This is a weak attack, but not to be ignored. The third case is a strong attack which leaves the scheme completely broken. Bob will not detect anything wrong, and it becomes certain that $i \in I_{1-c}$. Therefore Alice can get Bob's choice c with absolute certainty at zero risk. These problems are to be solved in the full protocol.

4.2 The enhancement scheme

Using a method in [3], any OT scheme can be used k times as sub-protocols, to build a stronger OT for Bob's security against the malicious Alice. At the end, only the XOR value of all the Bob's choice bits is his real choice. For completeness, we describe the general method here:

1. At the beginning, Alice has OT input $\{s_0, s_1\}$, while Bob has a choice c.
2. Alice generates a list of $k-1$ random bits $(\phi_{0,1}, \phi_{0,2} \ldots \phi_{0,k-1})$.
3. Alice chooses $\phi_{0,k}$ such that $\bigoplus_{i=1}^{k} \phi_{0,i} = s_0$.
4. Alice sets the second list of bits as $\phi_{1,i} = \phi_{0,i} \oplus s_0 \oplus s_1$ for all i.
5. The two parties run k copies of the sub-protocol OT. For each i, they use it to transfer the pair $(\phi_{0,i}, \phi_{1,i})$.
6. Bob makes the choices randomly, except that the XOR of all choices represents the real choice c. That is, denoting the choices in the OT sub-protocols by c_i, we have

$$\bigoplus_{i=1}^{k} c_i = c. \tag{2}$$

7. The final output of the receiver is s_c, as it can be computed from

$$s_c = \bigoplus_{i=1}^{k} \phi_{c_i,i}. \tag{3}$$

In this enhancement scheme, if Alice wants to guess c, she has to guess each of the c_i correctly. Therefore it can enhance Bob's security such that only one of the sub-protocols needs to be secure for Bob.

But in our case, using this enhancement only does not give a secure protocol from the semi-honest protocol, because the semi-honest protocol is completely insecure for Bob. Some extras measures are required to build a full OT scheme.

4.3 The complete protocol

In our full protocol, we set $k = n^3$. Alice and Bob run k times the semi-honest OT, using it as a sub-protocol. The idea is that, for each sub-protocol, Bob records the number of packets received at u_0. If Alice cheats by sending neither v_i nor v'_i for at least one i, the expected number of packets received at u_0 drops to $q(n-1)$ or below from the value of nq in the honest case. To distinguish the two distributions, we use the mid-point $q(n-\frac{1}{2})$ of the two mean values. After seeing all k sub-protocols, Bob aborts the main protocol if there are more than $k/2$ sub-protocols where the number of packets received at u_0 is below $q(n-\frac{1}{2})$. The full protocol is:

1. The k sub-protocols to be run in parallel are indexed by j. Alice prepares a matrix of random bits e_{ij} for $1 \leq i \leq n$ and $1 \leq j \leq k$ and sets $v_{ij} = j||i||e_{ij}$. She also sets $v'_{ij} = j||i||(1-e_{ij})$.
2. Alice sends all v_{ij} to the BDDC at t_0, and all v'_{ij} at t_1.
3. Bob waits to receive all packets and records their time of arrival. He checks for basic consistency, such that for every i and j he receives both $j||i||0$ and $j||i||1$ for exactly once each. He also checks that he does not receive both $j||i||0$ and $j||i||1$ at u_0. He aborts the protocol if any of these tests fail.
4. Otherwise Bob sets a counter $X = 0$ and enter the following procedure. For $1 \leq j \leq k$, he records the number of packets in format $j||*$ received in u_0. For any j, if this number is smaller than $n/2$, Bob aborts the protocol. If it is larger than $n/2$ but smaller than $q(n-\frac{1}{2})$, Bob adds one to the counter X.
5. Finishing the procedure above, Bob aborts the protocol if $X > \frac{k}{2}$.
6. If the protocol is not aborted, Bob selects c_j randomly for $1 \leq j \leq k$ except that

$$c = \bigoplus_{i=1}^{k} c_i. \tag{4}$$

7. For $1 \leq j \leq k$ Bob randomly selects a set of indices I_{j,c_j}, such that $|I_{j,c_j}| = n/2$ and for all $i \in I_{j,c_j}$ Bob has received some $j||i||*$ at time u_0. He sets $I_{j,1-c_j}$ to be the set of all $i \in \{1, 2, \ldots n\}$ such that $i \notin I_{j,c_j}$. Bob sends all $(I_{j,0}, I_{j,1})$ to Alice.

8. Alice generates a list of $k-1$ random bits $(\phi_{0,1}, \phi_{0,2} \ldots \phi_{0,k-1})$.
9. Alice chooses $\phi_{0,k}$ such that $\bigoplus_{j=1}^{k} \phi_{0,j} = s_0$.
10. Alice sets the second list of bits as $\phi_{1,j} = \phi_{0,j} \oplus s_0 \oplus s_1$ for all j.
11. For each j, Alice computes

$$\begin{aligned} \beta_{j,0} &= \bigoplus_{i \in I_{j,0}} e_{ij} \\ \beta_{j,1} &= \bigoplus_{i \in I_{j,1}} e_{ij} \end{aligned} \tag{5}$$

    and then sets $\sigma_{j,0} = \phi_{0,j} \oplus \beta_{j,0}$ and $\sigma_{j,1} = \phi_{1,j} \oplus \beta_{j,1}$.
12. Alice sends all $(\sigma_{j,0}, \sigma_{j,1})$ to Bob.
13. Bob knows e_{ij} whenever $i \in I_{j,c_j}$. He computes $\phi_{c_j,j}$ for all j.
14. The final output of Bob is s_c, as it can be computed from

$$s_c = \bigoplus_{j=1}^{k} \phi_{c_j,j}. \tag{6}$$

4.4 Security analysis

As usual, the proof of security is divided into correctness, Alice's security and Bob's security. Relying on the security of the sub-protocol, we show that the complete protocol has negligible failure probabilities in these three aspects.

Correctness: Observe that when both parties are honest, correctness is ensured if Bob does not abort the protocol. In this case, Bob may abort the protocol in two possible ways. The first possibility is at least one of the k sub-protocols has more than $n/2$ delayed packets. By the union bound, the probability for this is bounded by

$$n^3 e^{-2n(\frac{1}{2}-p)^2} \tag{7}$$

which is negligible in n. Next, the second possibility to abort is that $X > \frac{k}{2}$. For one sub-protocol, regarding to the probability of having the number of packets received at u_0 to be below $q(n-\frac{1}{2})$, the Hoeffding's inequality gives the upper bound as

$$\frac{1}{2} e^{\frac{-q^2}{2n}}. \tag{8}$$

Note that this quantity increases with n. This is because, when n is larger, the variance of the number of delayed packets is also larger. Next, we deal with the total number of such cases in the k runs of the sub-protocol. Setting

$$\delta = \frac{1}{2} - \frac{1}{2} e^{\frac{-q^2}{2n}}, \tag{9}$$

the probability of getting $X > \frac{k}{2}$ is bounded by

$$e^{-2k\delta^2} = e^{-2n^3\delta^2} \tag{10}$$

using the Chernoff bound. Observe that

$$n\delta = n(\frac{1}{2} - \frac{1}{2}e^{\frac{-q^2}{2n}}) \tag{11}$$

is a quantity that increases with n, but bounded such that

$$\lim_{n\to\infty} n(\frac{1}{2} - \frac{1}{2}e^{\frac{-q^2}{2n}}) = \frac{q^2}{4}. \tag{12}$$

Therefore the value of $e^{-2k\delta^2}$ falls exponentially in n. Thus the correctness of the final protocol is established.

Alice's security: By union bound, the probability of failure in Alice's security in the final protocol is no more than k times that of the sub-protocol. That is, it is upper bounded by $n^3(1-pq^2)^n$. This quantity drops exponentially in n too.

Bob's security: The malicious Alice must be dishonest in every sub-protocol in order to have any hope to get information on c. Security is perfect for Bob otherwise. Recall that Alice can only do the following for cheating:

1. For some i, j Alice sends both v_{ij} and v'_{ij} at t_0.
2. For some i, j Alice sends neither v_{ij} nor v'_{ij} at t_0.

We argue that, for the first type of cheating, if Alice sends both v_{ij} and v'_{ij} at t_0, this behavior will be detected with probability q^2. Therefore Alice can only do this a few times. To be more precise, let us assume that Alice does this m_1 times. It is clear that $m_1 < n$, or the probability of detection will be overwhelming with the increase of n. On the other hand, for the second type of cheating, if Alice sends neither v_{ij} nor v'_{ij} at t_0, it will not be detected immediately. Let us assume that Alice is doing this m_2 times. In order to gain any real advantage, she has to do at least one of either types of cheating in every sub-protocol. Therefore $m_2 > k - n$, and there are at least $k - n$ sub-protocols where only the second type of cheating occurs.

Focusing on such cases, the chance for that sub-protocol to have more than $q(n - \frac{1}{2})$ packets received at u_0 is upper bounded by

$$\mu = \frac{1}{2}e^{\frac{-q^2}{2(n-1)}}. \tag{13}$$

In the protocol, to deter Alice from cheating, Bob aborts if $X > \frac{k}{2}$. We show that X can reach $\frac{k}{2}$ even if we only consider these $k-n$ sub-protocols and ignore the rest. With the Hoeffding's inequality, the probability that X fails to reach $\frac{k}{2}$ is upper bounded by

$$e^{\frac{-2}{k-n}(\frac{k}{2}-n-(k-n)\mu)^2} \tag{14}$$

where we see that

$$\frac{-2}{k-n}(\frac{k}{2} - n - (k-n)\mu)^2 = \frac{-2}{n^3-n}(\frac{n^3}{2} - n - \frac{1}{2}(n^3-n)e^{\frac{-q^2}{2(n-1)}})^2$$

$$= \frac{-2}{n^3 - n}\left(\frac{n^3}{2}\left(1 - e^{\frac{-q^2}{2(n-1)}}\right) - \frac{ne^{\frac{-q^2}{2(n-1)}}}{2} - n\right)^2. \quad (15)$$

Using the fact that $n(1 - e^{\frac{-q^2}{2(n-1)}})$ converges to $\frac{q^2}{2}$ asymptotically, it is clear that the probability of $X > \frac{k}{2}$ is overwhelming with the increase of n.

5 Conclusion

With failure probabilities on correctness, Alice's security and Bob's security being negligible in n, we obtain the unconditionally secure OT protocol. Our protocol is the first to give unconditionally security in OT using channel delays. The practical value of our protocol is still limited, because accurate knowledge of p is required for Bob's security. Relaxing of this non-trivial assumption would be interesting for future study. Also, this protocol relies on the BDDC model. We believe a scheme based on the real, analog time channel delay may be possible. The final communication overhead of our protocol is $O(n^4 \log n)$ for security parameter n. This is rather high and it would be better if it can be reduced.

References

1. C. Crépeau: Equivalence between two flavours of oblivious transfer, In *Advances in Cryptology — CRYPTO '87*, LNCS 293, pp.350–354, 1988.
2. C. Crépeau: Efficient cryptographic protocols based on noisy channels, In *Advances in Cryptology — EUROCRYPT 1997*, LNCS 1233, pp.306–317, 1997.
3. C. Crépeau and J. Kilian: Achieving oblivious transfer using weakened security assumption, In *Proc. IEEE FOCS*, pp.42–52, 1988.
4. C. Crépeau, K. Morozov, and S. Wolf: Efficient unconditional oblivious transfer from almost any noisy channel, In *Fourth Conference on Security in Communication Networks*, LNCS 3352, pp.47–59, 2004.
5. C. Crépeau and G. Savvides: Optimal reductions between oblivious transfers using interactive hashing, In *Advances in Cryptology — EUROCRYPT 2006*, LNCS 4004, pp.201–221, 2006.
6. S. Even, O. Goldreich, and A. Lempel: A randomized protocol for signing contracts, *Communications of the ACM*, 28(6), pp.637–647, 1985.
7. J. Kilian: Founding cryptography on oblivious transfer, In *Proc. 20th ACM Symposium on Theory of Computing*, pp.20–31, 1988.
8. P. Palmieri and O. Pereira: Building oblivious transfer on channel delays, In *Proc. Inscrypt 2010*, LNCS 6584, pp.125–138, 2011.
9. M. Rabin: How to exchange secrets by oblivious transfer, *Technical Report TR-81*, Aiken Computation Laboratory, Harvard University, 1981.
10. J. Wullschleger: Oblivious transfer from weak noisy channels, In Theory of Cryptography Conference 2009, LNCS 5444, pp.332–349, 2009.
11. A. Yao: Protocols for secure computations, *Proc. 23rd IEEE Symposium on Foundations of Computer Science*, pp.160–164, 1982.