Reducing Complexity Assumptions for Oblivious Transfer

K.Y. Cheong and Takeshi Koshiba

Division of Mathematics, Electronics and Informatics,
Graduate School of Science and Engineering, Saitama University
255 Shimo-Okubo, Sakura, Saitama 338-8570, Japan
{kaiyuen,koshiba}@tcs.ics.saitama-u.ac.jp

Abstract. Reducing the minimum assumptions needed to construct various cryptographic primitives is an important and interesting task in theoretical cryptography. *Oblivious transfer*, one of the most basic cryptographic building blocks, could be also studied under this scenario. Reducing the minimum assumptions for oblivious transfer seems not an easy task, as there are a few impossibility results under black-box reductions.

Until recently, it is widely believed that oblivious transfer can be constructed with trapdoor permutations. Goldreich pointed out some flaw in the folklore and introduced some enhancement to cope with the flaw. Haitner then revised the enhancement more properly. As a consequence they showed that some additional properties for trapdoor permutations are necessary to construct oblivious transfers. In this paper, we discuss possibilities of basing not on trapdoor permutations but on trapdoor functions in general. We generalize previous results and give an oblivious transfer protocol based on a collection of trapdoor functions with some extra properties with respect to the length-expansion and the preimage size. We discuss that our reduced assumption is almost minimal and show the necessity for the extra properties.

Keywords: oblivious transfer, trapdoor one-way functions.

1 Introduction

1.1 Oblivious Transfer

Oblivious Transfer (OT) is an important two-party cryptographic protocol. The first known OT system was introduced by Rabin [27] in 1981 where a message is received with probability 1/2 and the sender cannot know whether his message reaches the receiver. Prior to this, Wiesner [31] introduced a primitive called multiplexing, which is equivalent to the 1-out-of-2 OT [10] known today, but it was then not seen as a tool in cryptography. In 1985, Even et al. defined the 1-out-of-2 OT [10], where the sender has two secrets σ_0 and σ_1 and the receiver can choose one of them in an oblivious manner. That is, the sender cannot know the receiver's choice $i \in \{0, 1\}$ and the receiver cannot know any information

T. Takagi and M. Mambo (Eds.): IWSEC 2009, LNCS 5824, pp. 110–124, 2009.
© Springer-Verlag Berlin Heidelberg 2009

on σ_{1-i}. The former property is called *receiver's privacy* and the latter *sender's privacy*. Later, Crépeau [7] showed that Rabin's OT and the 1-out-of-2 OT are equivalent. Furthermore, the more general 1-out-of-N OT (where the sender has N secrets), the more specific 1-out-of-2 *bit* OT (where the secrets are one bit long), are similarly defined and the reductions among the variants of OT have been discussed in the literature, e.g. [2,3,8].

OT protocols are fundamental building blocks of modern cryptography. Most notably, it is known that any multi-party secure computation can be based on OT [22,14]. Various implementations of OT protocols have been proposed, and they are all based on some computational assumptions. As an efficient implementation, Naor and Pinkas has proposed a protocol [24] based on Diffie and Hellman [9] type of problems. More recently, a universally composable [4] OT protocol has been constructed based on a variety of assumptions [25].

1.2 Complexity Assumptions of OT

With limited exceptions such as one-time-pad encryption [30] and secret sharing scheme [29], most cryptographic primitives rely on certain computational assumptions. In 1-out-of-2 OT, by simple arguments it can be seen that either sender's privacy or receiver's privacy must be protected by some computational assumptions, where the other party may be protected in the information theoretic sense. The symmetry of 1-out-of-2 bit OT [32] implies that we have the freedom to choose which side to protect in which way when we are given a protocol.

We are interested to know the minimum computational assumptions necessary for building OT. Unavoidably, for each OT protocol proposed, we may have to rely on some unproven computational assumptions for its security. To some extent, this is acceptable, since most cryptographic protocols require the existence of one-way functions [20]. This in particular implies $P \neq NP$, which is unproven.

On the other hand, since it is impossible to avoid all the computational assumptions, we would like to construct protocols based upon as little assumptions as possible. In any cryptographic protocol, less underlying assumptions means more confidence on the security. Therefore, the study of minimum computational assumptions of various cryptographic primitives is an important part in cryptographic research. For example, while one-way permutation is known to imply statistically-hiding commitment [23], this assumption has been reduced in [17]. And finally, Haitner and Reingold [18] recently proved that statistically-hiding commitment can be constructed from any one-way function. That enables us to rely on one-way functions to use zero-knowledge arguments.

The situation for OT is more complicated. From the discussion in [19], it is known that OT can be based on one-way functions if there exists a *witness retrievable compression algorithm* for some type of SAT formulas. But on the other hand, the oracle separation [21] between one-way permutations and OT rules out the possibility of blackbox reductions from OT to one-way functions. In general, it is believed that it will be very difficult, if not impossible, to build OT with one-way functions only.

In the original paper of [10], trapdoor permutations with some extra properties are used to construct OT. In [15], Haitner proposed a similar protocol which in theory reduced the computational assumptions required by [10]. The protocol uses a collection of *dense* trapdoor permutations. In [26], another construction of [10] is made from a new type of trapdoor functions (called *lossy trapdoor functions*) with some specific properties. However, the definition comes rather from concrete problems such as the Diffie-Hellman problem and lattice problems than from the theoretical origin.

In this paper, we focus on two issues. We explore the possibility to further reduce the computational assumptions of OT as stated in [15]. We like to know if trapdoor functions, rather than trapdoor permutations, can be used to construct OT. Also, we investigate the essential properties of trapdoor functions necessary for OT. For example, Bellare et al. showed that many-to-one trapdoor functions with exponential pre-image size can be constructed from one-way functions [1]. This fact says that many-to-one trapdoor functions with polynomial pre-image size may have very different properties from those of super-polynomial pre-image size. It also suggests that OT may not be constructible from many-to-one trapdoor functions with super-polynomial pre-image size.

While public key encryptions can be constructed from many-to-one trapdoor functions with polynomial pre-image size as stated in [1], there exists an oracle separation in [11] between public key encryptions and OT. Thus, it is natural to ask whether OT can be constructed from many-to-one trapdoor functions with polynomial pre-image size.

As the main result of this paper, we show that the protocol of [15] can be improved to make it applicable to *general* trapdoor functions. The permutation property is thus not essential. This possibility is actually discussed in the concluding remarks of [15]. But the trapdoor functions used in our protocol have some extra properties (and restrictions) with respect to pre-image size and length expansion. Consequently, we have an OT construction based on a weaker assumption than the previous results, because a trapdoor permutation is a trapdoor function with strictly single pre-image and zero length expansion. Also, we provide arguments that these extra properties are necessary, and are close to the minimum in blackbox reductions.

1.3 Relation to Previous Results

The original paper of [10] for 1-out-of-2 OT opens the discussion for the minimum computational assumptions of OT. In [10], a public key encryption scheme with an extra property is used to construct OT. Stated explicitly, the property is that a valid ciphertext can be uniformly sampled from the plaintext domain. This condition is explained in [15] such that, in general, a trapdoor permutation suffices for OT if it is possible to sample an image of it without knowing the pre-image. The term Enhanced Trapdoor Permutation is used in [12] to represent such a trapdoor permutation.

Following the construction of [10] and discussion of [12], Haitner reduced the assumption further by using a set of dense trapdoor permutations [15]. This

essentially establishes the sampling property without requiring it explicitly. This seems to be close to minimal, as [21] shows the impossibility for blackbox reduction from OT to one-way permutations. In this paper, we follow the insights and techniques of [15] to further reduce the computational assumptions such that trapdoor functions, rather than trapdoor permutations, may be used to construct OT. Near the end of [15], the possibility of using trapdoor functions for OT has been considered, but the further assumptions required for such a trapdoor function are not clearly discussed.

Taking the impossibility results implied by [1] and [11], we see that the preimage size and length-expansion of the trapdoor function are vital for OT possibility. Therefore, we consider these issues and try to build OT with what may be regarded as minimum assumptions in this framework.

2 Preliminaries

2.1 Blackbox Reduction

Our work is about basing OT on a primitive with as few assumptions as possible. We focus on blackbox reductions only, where the primitive used as the building block is treated as a blackbox. This means the protocol only deals with the input and output of the underlying primitive, but not its internal calculations. Most known reductions and impossibility results are based on blackbox reductions. The impossibility results initiated by [21] shows that OT cannot be based on one-way functions in blackbox reductions. In [1] and [11], other impossibility results concerning OT are also shown based on [21]. These results are related to our protocol.

As discussed in [21], blackbox reductions may be divided into fully-blackbox reductions and semi-blackbox reductions. In a fully-blackbox reduction, any adversary who breaks the constructed primitive can be used as a blackbox for another algorithm which breaks the building-block primitive. The semi-blackbox reduction basically does not have this requirement. Therefore, a fully-blackbox reduction seems to imply a closer relation between the constructing and constructed primitives. On the other hand, [28] shows that the difference between fully-blackbox and semi-blackbox reductions is not as great as what may be perceived in [21].

In this paper, we focus on fully-blackbox reductions. Any adversary who breaks our protocol can be used as a blackbox to break the trapdoor function used. In fact, in our OT protocol, only the sender's privacy is protected computationally. The receiver's privacy is protected in information theoretic sense. Therefore it is the sender's privacy that is equivalent to the security of the trapdoor function.

2.2 Semi-honest Model

We limit ourselves to the semi-honest model in our OT protocol. In a semi-honest protocol, all parties are assumed to follow the protocol properly, except that

they may try to extract extra information from the communications, possibly by performing some computations afterwards. In [12] it is shown that a protocol for semi-honest model can be used to construct an equivalent protocol in the general malicious model, where nothing is assumed about the parties. Moreover, in [16] and [6] it is further shown that such a construction can be done in the blackbox way, where the semi-honest protocol is used as a blackbox.

These known constructions of protocols for the malicious model from the semi-honest model are based on commitment schemes or zero-knowledge proofs. Regarding to complexity assumptions, they also require the existence of one-way functions, which is a rather basic assumption for most cryptographic primitives including OT. Using the combination of these results, we can obtain OT in the general model simply by constructing a semi-honest OT protocol. The use of semi-honest model can simplify both the definition and the construction of OT.

2.3 1-out-of-2 Bit OT

In this paper, we consider only the 1-out-of-2 bit OT. It is known that other versions of OT can be constructed using 1-out-of-2 bit OT as building blocks. The sender has two secret bits (σ_0, σ_1) and the receiver has a choice bit i. In the correct output, the receiver will get σ_i and not σ_{1-i}, whereas the sender will get no information about i. More formally, let $V_S(\sigma_0, \sigma_1, i)$ and $V_R(\sigma_i, \sigma_{1-i}, i)$ be the random variables for the sender's and receiver's view of the protocol respectively, given the receiver's choice i and the sender's secrets σ_0 and σ_1. Note that the notation of $V_R(\sigma_i, \sigma_{1-i}, i)$ is informal because the order of parameters is not fixed. This is not a problem because the receiver always knows i and the order of the other two parameters are decided accordingly. Also, these variables have to exist because we assume the OT protocol is run in a semi-honest way. The privacy properties of OT can then be defined as, for all possible i, σ_0 and σ_1:

1. *Sender's privacy*: Receiver gains no computational knowledge about σ_{1-i}. That is, for any probabilistic polynomial time algorithm M,

$$|\Pr[M(V_R(\sigma_i, 1, i)) = 1] - \Pr[M(V_R(\sigma_i, 0, i)) = 1]| < neg(n) \qquad (1)$$

   where $neg(n)$ stands for a negligible function of n.[1]
2. *Receiver's privacy*: Sender gains no computational knowledge about i.

$$|\Pr[M(V_S(\sigma_0, \sigma_1, 0)) = 1] - \Pr[M(V_S(\sigma_0, \sigma_1, 1)) = 1]| < neg(n) \qquad (2)$$

   for any probabilistic polynomial time algorithm M.

The standard definition of OT above requires that both parties are at least protected computationally. Nonetheless, in an OT system, it is known that at most one party's privacy can be perfectly protected in information theoretic sense. In that case, even if the other party is computationally unbounded, the first party's privacy is still maintained.

[1] A negligible function of n, denoted by $neg(n)$, is defined as a function of n where $|neg(n)| < |\frac{1}{g(n)}|$ for any polynomial $g(n)$, for large enough n.

2.4 Weak OT

A Weak OT protocol (WOT) is a relaxed version of OT. The weakness is described by three parameters. In a $(\epsilon_1, \epsilon_2, \epsilon_3)$-WOT, the secret required by the receiver is only guaranteed to pass correctly with a probability no less than $1 - \epsilon_1$. This is called the *correctness* of the protocol. On the other hand, the receiver does not gain more computational advantage about σ_{1-i} than ϵ_2, and the sender does not gain more computational advantage about i than ϵ_3. Similar to the normal OT, we have:

1. *Sender's privacy*: For any probabilistic polynomial time algorithm M,

$$|\Pr[M(V_R(\sigma_i, 1, i)) = 1] - \Pr[M(V_R(\sigma_i, 0, i)) = 1]| \leq \epsilon_2. \tag{3}$$

2. *Receiver's privacy*: For any probabilistic polynomial time algorithm M,

$$|\Pr[M(V_S(\sigma_0, \sigma_1, 0)) = 1] - \Pr[M(V_S(\sigma_0, \sigma_1, 1)) = 1]| \leq \epsilon_3. \tag{4}$$

Note that, under our definition, a $(neg(n), neg(n), neg(n))$-WOT is equal to OT.

2.5 Pairwise Independent Universal Hash Functions

In this paper we also need a construction called the pairwise independent universal hash function. For a parameter n, let there be two sets $L_1 = \{1, 2, \ldots, 2^n\}$ and $L_2 = \{1, 2, \ldots, l\}$ such that $l \leq 2^n$. From [5] it is known that, for any choice of l, there exists an efficient family of hash functions H_n with the following properties:

1. Any function $h \in H_n$ has domain L_1 and range L_2.
2. There exists a polynomial-time algorithm to sample $h \in H_n$ uniformly.
3. There exists a polynomial-time algorithm to evaluate $h(x)$ given h and $x \in L_1$.
4. When h is uniformly sampled, for every distinct $x_1, x_2 \in L_1$ and every $y_1, y_2 \in L_2$,

$$\Pr[h(x_1) = y_1 \wedge h(x_2) = y_2] = \frac{1}{l^2}. \tag{5}$$

3 Trapdoor Functions for OT

In this paper we are constructing OT based on a special type of trapdoor function. We first define the normal trapdoor function, and add some extra restrictions suitable for our purpose. At the same time, we try to minimize the assumptions we make. In general, a collection of (non-injective) trapdoor functions F have the following properties:

1. There exists an efficient algorithm which uniformly selects a function f_α in F, represented by α, and generates the trapdoor t at the same time.

2. Denote the domain of the function by D_α. If $x \in D_\alpha$ then $f_\alpha(x)$ can be computed efficiently.
3. Without the trapdoor t, for a uniformly chosen $x \in D_\alpha$, when given $f_\alpha(x)$ it is computationally infeasible to obtain any $x' \in D_\alpha$ such that $f_\alpha(x') = f_\alpha(x)$.
4. For any $x \in D_\alpha$, given $f_\alpha(x)$ and t, there exist an efficient algorithm to find one $x' \in D_\alpha$ such that $f_\alpha(x') = f_\alpha(x)$. That is, we can calculate $x' = f_\alpha^{-1}(t, y)$ where $y = f_\alpha(x')$, if in the first place $y = f_\alpha(x)$ for some x in the domain.

In any practical use of such a trapdoor function, we can assume either $D_\alpha = \{0,1\}^n$ or $D_\alpha \subset \{0,1\}^n$ for some parameter n. The former is called full domain while the latter normally further requires a sampling algorithm for finding elements in D_α. For our trapdoor function, the full domain is preferred, but we can relax the assumption a bit, due to the results of [15]. Without loss of generality, $D_\alpha \subset \{0,1\}^n$. But we also assume that D_α is dense in $\{0,1\}^n$. This *dense domain assumption* is the first assumption we add to our otherwise general trapdoor function. It means there exist a polynomial $p(n)$ such that, for all α, we have

$$\frac{|D_\alpha|}{2^n} > \frac{1}{p(n)}. \tag{6}$$

Next, for all $x \in \{0,1\}^n$ we assume $f_\alpha(x)$ can be evaluated in general using the same algorithm evaluating the function, and the algorithm will halt in polynomial time, producing some output. In practice, this has to be justified by adding a measure which terminates the algorithm when the running time exceeds some fixed value, and gives a default output. That is, even if $x \notin D_\alpha$ the algorithm will still run and produce a string as output. The definition of $f_\alpha(x)$ is extended to handle any $x \in \{0,1\}^n$. As we do not assume we can detect $x \notin D_\alpha$, nothing is assumed about the output string in this case.

In the same way, for all $x \in D_\alpha$ we assume $f_\alpha(x) \in \{0,1\}^m$ for some fixed m. And for all $y \in \{0,1\}^m$, we assume the function $f_\alpha^{-1}(t, y)$ can be evaluated using the same algorithm evaluating the inverse function, and the algorithm will halt in polynomial time, producing some output. In other words, the definition of $f_\alpha^{-1}(t, y)$ is extended for all $y \in \{0,1\}^m$.

3.1 Extra Assumptions

In order to construct our OT protocol, we require the trapdoor functions to have a few more properties. We call them the Extra Assumptions, in order to distinguish our trapdoor functions from the general ones.

1. *Pre-image assumption*: For any α, when $x \in D_\alpha$ and $y = f_\alpha(x)$, the number of pre-images of y is bounded by a polynomial. That is, there exists a polynomial $q_1(n)$ such that, for all α and y,

$$I_{\alpha,y} = \{x \in D_\alpha : f_\alpha(x) = y\} \tag{7}$$

$$|I_{\alpha,y}| \leq q_1(n). \tag{8}$$

2. *Expansion assumption*: For $x \in D_\alpha$ we have $f_\alpha(x) \in \{0,1\}^m$ with $m = n + \log q_2(n)$ where $q_2(n)$ is a polynomial in n. That is equal to saying that the expansion (in terms of the length of strings) of the function is in $O(\log n)$.

3.2 Necessity of the Extra Assumptions

We clarify that our aim is to define a general set of trapdoor functions with specific restrictions, such that any trapdoor functions meeting these restrictions can be used to construct OT. Therefore, when we investigate a particular set of such restrictions, one single counterexample of OT impossibility under a trapdoor function meeting these restrictions suffices to indicate that the set of restrictions in question is not tight enough. The counterexamples can be specially designed for this purpose, and may only exist theoretically.

To see the necessity of the Extra Assumptions, first look at the pre-image assumption due to [1], where non-injective trapdoor functions are studied. The following trapdoor function with exponential pre-image size can be blackbox constructed from a one-way permutation.

1. A one-way permutation $f_1(x)$ is given for $x \in \{0,1\}^n$.
2. Choose a trapdoor value $t \in \{0,1\}^n$. Let $\alpha = f_1(t)$. For $v, u, x \in \{0,1\}^n$ we define
$$f_2(v,u,x) = \begin{cases} v & \text{if } f_1(u) = \alpha \\ f_1(x) & \text{otherwise.} \end{cases} \tag{9}$$
3. This is a trapdoor function in the sense that, if t is known, we can calculate from an image y a value (y,t,x) as a pre-image, using any x. The function f_2 is also one-way because when t is unknown, its inversion requires the inversion of f_1 on either y or α.

On the other hand, it is known that no OT (including semi-honest model) can be blackbox reduced to one-way permutation [21]. This implies that semi-honest OT cannot be blackbox constructed from a trapdoor function with exponential pre-image size.

The expansion assumption is related to [11], which shows an example of a trapdoor function with linear length expansion. Arguments are presented relative to a world with a PSPACE-complete oracle. The following random (oracle) functions are constructed as the only source of computational hardness, but OT does not exist in this world. This implies that OT cannot be blackbox constructed from any such functions in the real world.

1. $\alpha = f_3(t)$ is a uniformly distributed, length-tripling function. It generates an identifier α by inputting trapdoor t, an arbitrary string, to the function.
2. $y = f_4(x,r,\alpha)$ is an injective, uniformly distributed, length-tripling function on the set of valid inputs. Input α is valid if there exists t such that $\alpha = f_3(t)$. Also, x and r are valid if $|x| = |r| = |t|$. On any invalid input the function outputs $\perp$.
3. f_5 is a function basically for inverting f_4, such that $x = f_5(y,t)$ whenever $y = f_4(x,r,f_3(t))$ for some (x,r). There is at most one such x, as f_4 is injective. When there is no such x, $f_5(y,t) = \perp$.

An injective trapdoor function can be based on f_4 simply by fixing $r = 0$ all the time. It is length-expanding in $O(n)$. The length-expanding property of this trapdoor function makes it difficult to sample valid images of the function without knowing the pre-image. This is one main reason why OT cannot be based on it.

Note that in these two examples of OT impossibility, the trapdoor function with exponential pre-image size is not length-expanding, and the length-expanding trapdoor function is injective, as shown above. That means in our trapdoor function for OT possibility, both the pre-image assumption and the expansion assumption are required at the same time.

Moreover, our trapdoor function f_2 with exponential pre-image size is a full domain trapdoor function, as it takes any string (in the right format) as input. The length-expanding trapdoor function f_4 also has a full domain as we only consider x as the input. This further shows the necessity of the pre-image assumption and the expansion assumption, regardless of the dense domain assumption.

On the other hand, we do not rule out OT possibilities based on other assumptions. For instance, in [11] it is implied that if α can be sampled independent of t, then OT may be based on such a trapdoor function, regardless of length expansion. Although we do not see it as a minimal assumption in general, this assumption is indeed rather independent of ours.

We also note that there is still space between our construction and the known impossibility results, for both the pre-image assumption and the expansion assumption. A possible gap between super-polynomial and exponential functions is neglected up to this point. For pre-image size, while impossibility results are known for the exponential, our construction is for the polynomial. Similarly, for length expansion, while impossibility results are for the linear, our construction is for the logarithm of polynomial. In this sense, we say that our Extra Assumptions are close, but may not be equal to the real minimum.

4 The Protocol

We point out that the construction of our OT protocol is mostly same as [15]. Every step is basically the same, while there are some modifications only due to the differences of the trapdoor functions involved. A semi-honest WOT protocol is first constructed. After that, the process to enhance it to a semi-honest OT is exactly the same as [15].

First of all, we select a collection of pairwise independent universal hash functions H_n with domain $\{0,1\}^n$ and range $\{1, 2, \ldots, g(n)p(n)q_1(n)\}$ where $g(n) > 1$ is a relatively large polynomial of our choice. The actual choice of $g(n)$ is related to the WOT parameters and will be discussed later. The sender has secret bits (σ_0, σ_1) and the receiver has the choice bit i. The protocol is:

1. The sender uniformly selects a trapdoor function (α, t) and a hash function $h \in H_n$.
2. The sender sends (h, α) to the receiver.

3. The receiver selects uniformly $s \in \{0,1\}^n$ and calculates $f_\alpha(s)$. If $f_\alpha(s) \notin \{0,1\}^m$ another s is selected iteratively until $f_\alpha(s) \in \{0,1\}^m$. After that the receiver sets $r_i = f_\alpha(s)$ and selects uniformly $r_{1-i} \in \{0,1\}^m$ where $r_i \neq r_{1-i}$.
4. The receiver sends $\{r_0, r_1\}$ in random order to the sender.
5. Not knowing the order of $\{r_0, r_1\}$, for both $j = 0, 1$ the sender checks that the following conditions are satisfied.

$$f_\alpha^{-1}(t, r_j) \in \{0,1\}^n \tag{10}$$
$$f_\alpha(f_\alpha^{-1}(t, r_j)) = r_j. \tag{11}$$

   If the answer is negative, the sender aborts the current iteration and restarts the protocol. Otherwise the protocol continues with the sender setting for $j = 0, 1$

$$v_j = h(f_\alpha^{-1}(t, r_j)). \tag{12}$$

6. The sender sends $\{v_0, v_1\}$ in the same order as he received $\{r_0, r_1\}$ from the receiver before.
7. Receiver checks that $v_i = h(s)$. If the result is negative, the current iteration aborts and the protocol is restarted. Otherwise, the receiver reveals the true order of (r_0, r_1) to the sender. From here, both r_0 and r_1 are thought to be good candidates as the keys in the OT protocol. The receiver is thought to know the pre-image of exactly one of them, whereas the sender does not know which one.
8. For both $j = 0, 1$ the sender chooses $z_j \in \{0,1\}^n$ uniformly and sets

$$c_j = \sigma_j \oplus b(f_\alpha^{-1}(t, r_j), z_j) \tag{13}$$

   where $b(x, y)$ is the inner product of x, y modulus 2, a hardcore predicate.
9. The sender sends (c_0, c_1, z_0, z_1) to the receiver.
10. The receiver outputs $\sigma'_i = b(s, z_i) \oplus c_i$. This is the secret required.

5 Analysis of Protocol

To make the analysis easier, we define the following sets before we proceed.

$$D'_\alpha = \{x \in D_\alpha : x = f_\alpha^{-1}(t, f_\alpha(x))\} \tag{14}$$
$$R_\alpha = f_\alpha(D_\alpha) = f_\alpha(D'_\alpha) \tag{15}$$

where R_α is the range of the trapdoor function. Also, there is a one-to-one relationship between D'_α and R_α. Next, we define the following sets, acting as an extension of the domain of the trapdoor function.

$$D''_\alpha = \{x \in \{0,1\}^n : x = f_\alpha^{-1}(t, f_\alpha(x)) \wedge f_\alpha(x) \in \{0,1\}^m\} \tag{16}$$
$$R''_\alpha = f_\alpha(D''_\alpha). \tag{17}$$

Naturally, there is also a one-to-one relationship between elements in D''_α and R''_α. Also we see that $D'_\alpha = D_\alpha \cap D''_\alpha$.

5.1 Running Time

Observe that, due to the dense property of D_α in $\{0,1\}^n$ and D'_α in D_α, D'_α is also dense in $\{0,1\}^n$. As $|D'_\alpha| = |R_\alpha|$ and $m = n + \log q_2(n)$, R_α is dense in $\{0,1\}^m$. To be more precise, in our protocol we have, in each iteration,

$$\Pr(s \in D'_\alpha) > \frac{1}{p(n)q_1(n)} \tag{18}$$

$$\Pr(r_{1-i} \in R_\alpha) > \frac{1}{p(n)q_1(n)q_2(n)}. \tag{19}$$

In one iteration, if $s \in D'_\alpha$ and $r_{1-i} \in R_\alpha$ then the protocol will reach the end successfully. It is easy to see that the total expected number of iterations is polynomial in n. Thus, we say the protocol runs in expected polynomial time. To be precise, in order to guarantee that the protocol will come to a halt, we need to set a counter for the number of iterations. The protocol is terminated when the counter exceeds some predetermined number. In this case, the running time will be polynomial, while the weakness parameter for correctness in WOT will be increased by a negligible amount.

Also, we see how the properties of the trapdoor function affect the running of the protocol. Both the expansion and pre-image size affect the density of usable elements in the domain and range of the trapdoor function. Here they are required for the running time to be polynomial.

5.2 Correctness

With the discussion above, the protocol will be prematurely terminated with a negligible probability. If this does not happen, the protocol is executed to the last step. In the last iteration of the protocol, the receiver can get the required secret correctly if $s = f_\alpha^{-1}(t, r_i)$.

For any initial choice of s and r_{1-i}, failure occurs if $s \neq f_\alpha^{-1}(t, r_i)$ and at the same time $h(s) = v_i$. This is independent of the choice of r_{1-i}, even though r_{1-i} may lead to an aborted round in the protocol. For probability we write:

$$\Pr(s = f_\alpha^{-1}(t, r_i)) > \frac{1}{p(n)q_1(n)} \tag{20}$$

$$\Pr(s \neq f_\alpha^{-1}(t, r_i) \wedge h(s) = v_i) < (1 - \frac{1}{p(n)q_1(n)})(\frac{1}{g(n)p(n)q_1(n)}) \tag{21}$$

and the remaining probability is that the iteration does not reach the end of the protocol. Thus, the probability of correctness, given that the protocol is completely finished, would be

$$\begin{aligned} 1 - \epsilon_1 &> \frac{\frac{1}{p(n)q_1(n)}}{\frac{1}{p(n)q_1(n)} + (1 - \frac{1}{p(n)q_1(n)})(\frac{1}{g(n)p(n)q_1(n)})} \\ &= \frac{g(n)}{g(n) + (1 - \frac{1}{p(n)q_1(n)})} \end{aligned}$$

$$> 1 - \frac{1}{g(n)} \tag{22}$$

as $p(n) \geq 1$ and $q_1(n) \geq 1$. This gives the required result that $\epsilon_1 < 1/g(n)$. If we also consider the minor case that the protocol may not run through the end, we have $\epsilon_1 < 1/g(n) + neg(n)$.

5.3 Privacy of Receiver

First of all we argue that, when $s = f_\alpha^{-1}(t, r_i)$, we have $s \in D''_\alpha$. On the other hand, if the protocol is run through the end in an iteration, then it must be that $r_{1-i} \in R''_\alpha$. Due to the one-to-one relation between elements of D''_α and R''_α, we conclude in this case that both r_0 and r_1 will appear uniformly distributed in R''_α, protecting the privacy of the receiver. This is guaranteed at the time the order of (r_0, r_1) is revealed to the sender. As a result, the only problem occurs when $s \neq f_\alpha^{-1}(t, r_i)$. Thus the weakness parameter for receiver's privacy is bounded by the same events that determine correctness, giving $\epsilon_3 < 1/g(n)$.

At this point, it is important to see that when $s = f_\alpha^{-1}(t, r_i)$ the receiver's privacy is protected in information theoretic sense, without requiring permutation properties in the trapdoor functions. In previous works, the permutation property in trapdoor permutations is usually needed to protect the receiver's privacy in information theoretic sense, while the sender's privacy is protected by computational hardness of the inverse function.

5.4 Privacy of Sender

The main weakness of our WOT protocol is on the sender's privacy. After all, r_{1-i} is finally not even guaranteed to be in R_α with high probability. We can assume nothing about the computational hardness of inverting f_α in that case.

But if $r_{1-i} \in R_\alpha$, the sender's privacy should be protected. In this case we can see that if the receiver has non-negligible advantage in guessing σ_{1-i} then he also has non-negligible advantage guessing $b(f_\alpha^{-1}(t, r_{1-i}), z_{1-i})$. From the theory for this hardcore predicate [13], this means the receiver has a non-negligible advantage to compute $f_\alpha^{-1}(t, r_{1-i})$.

Note that the receiver is holding r_{1-i} and $h(f_\alpha^{-1}(t, r_{1-i}))$ to help his computation. But if there is such an efficient algorithm M to find $f_\alpha^{-1}(t, r_{1-i})$ in this case, then we also have a polynomial time algorithm solving $f_\alpha^{-1}(t, r_{1-i})$ from r_{1-i} alone, by running M with the setting of $h(f_\alpha^{-1}(t, r_{1-i})) = y$ for each $y \in \{1, 2, \ldots, g(n)p(n)q_1(n)\}$. Each iteration is terminated at a reasonable time limit if it does not give an output. Any potential solution x for $f_\alpha^{-1}(t, r_{1-i})$ can be checked by $f_\alpha(x)$.

Finally, if $f_\alpha^{-1}(t, r_{1-i})$ can be calculated from r_{1-i} in our protocol with non-negligible probability, the computational hardness of the trapdoor function must be violated because r_{1-i} is generated by uniform sampling in the first place. This results in a contradiction. Therefore, we conclude that when $r_{1-i} \in R_\alpha$, the sender's privacy is maintained.

The event $r_{1-i} \in R_\alpha$ is only related to the density of R_α in $\{0,1\}^m$. Thus we have

$$\epsilon_2 < 1 - \frac{1}{p(n)q_1(n)q_2(n)} \tag{23}$$

where again we see that the privacy of sender depends on all properties of our trapdoor function: the dense property $p(n)$, the pre-image property $q_1(n)$ and expansion property $q_2(n)$.

6 Strengthening the Weak OT

As a result, we have a WOT with $\epsilon_1 < \frac{1}{g(n)} + neg(n)$, $\epsilon_2 < 1 - \frac{1}{G(n)}$ and $\epsilon_3 < \frac{1}{g(n)}$, where $G(n) = p(n)q_1(n)q_2(n)$. The value of $g(n)$ is of our choice. It is possible to strengthen WOT to standard OT [33] under some conditions in general. In our protocol, exactly the same method of [15] can be used to strengthen the WOT to OT in the semi-honest model. From [15], it works with $g(n) = 3n^2G(n)$. The WOT is used as a blackbox a number of times to suppress the weakness parameters until they become negligible. This completes the last step of the construction of standard OT with blackbox usage of our trapdoor functions.

7 Concluding Remarks

We believe the main contribution of this paper is two-fold. In the constructive sense, we follow [15] and continue the work to remove the strict permutation requirement in trapdoor functions for constructing OT. We show that trapdoor functions with three extra properties are sufficient. They are the dense assumption, pre-image assumption and expansion assumption.

On the other hand, through the known blackbox impossibility results, we argue that the pre-image assumption and expansion assumption are hard to remove. The one question remains is about OT possibility if the dense assumption is removed, keeping only the other two assumptions. This question can be divided into two cases. The first case is that the trapdoor function is not required to be a permutation. Then the answer is negative, as a counterexample can easily be constructed by setting $D_\alpha = \{0,1\}^{\frac{m}{15}}$ with $R_\alpha \subset \{0,1\}^m$ and following exactly the same arguments for linear expansion mentioned in this paper. If a trapdoor permutation is used, then we are back to an old question. We know that the Enhanced Trapdoor Permutation [12] suffices, but OT based on trapdoor permutation only is an interesting open question, and the answer is still being awaited.

References

1. Bellare, M., Halevi, S., Sahai, A., Vadhan, S.P.: Many-to-one trapdoor functions and their relation to public-key cryptosystems. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 283–299. Springer, Heidelberg (1998)

2. Brassard, G., Crépeau, C., Santha, M.: Oblivious transfers and intersecting codes. IEEE Transactions on Information Theory 42(6), 1769–1780 (1996)
3. Brassard, G., Crépeau, C., Wolf, S.: Oblivious transfers and privacy amplification. Journal of Cryptology 16(4), 219–237 (2003)
4. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: Proc. 42nd IEEE Symposium on Foundations of Computer Science, pp. 136–145 (2001)
5. Carter, J., Wegman, M.: Universal classes of hash functions. Journal of Computer and System Sciences 18(2), 143–154 (1979)
6. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Simple, black-box constructions of adaptively secure protocols. In: Theory of Cryptography Conference 2009. LNCS, vol. 5444, pp. 387–402 (2009)
7. Crépeau, C.: Equivalence between two flavours of oblivious transfers. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 350–354. Springer, Heidelberg (1988)
8. Crépeau, C., Savvides, G.: Optimal reductions between oblivious transfers using interactive hashing. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 201–221. Springer, Heidelberg (2006)
9. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)
10. Even, S., Goldreich, O.: A Lempel: A randomized protocol for signing contracts. Communications of the ACM 28(6), 637–647 (1985)
11. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: Proc. 41st IEEE Symposium on Foundations of Computer Science, pp. 325–335 (2000)
12. Goldreich, O.: Foundations of Cryptography, vol II. Cambridge University Press, Cambridge (2004)
13. Goldreich, O., Levin, L.: A hard-core predicate for all one-way functions. In: Proc. 21st ACM Symposium on Theory of Computing, pp. 25–32 (1989)
14. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: Proc. 19th ACM Symposium on Theory of Computing, pp. 218–229 (1987)
15. Haitner, I.: Implementing oblivious transfer using collection of dense trapdoor permutations. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 394–409. Springer, Heidelberg (2004)
16. Haitner, I.: Semi-honest to malicious oblivious transfer—the black-box way. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 412–426. Springer, Heidelberg (2008)
17. Haitner, I., Horvitz, O., Katz, J., Koo, C.-Y., Morselli, R., Shaltiel, R.: Reducing complexity assumptions for statistically-hiding commitment. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 58–77. Springer, Heidelberg (2005)
18. Haitner, I., Reingold, O.: Statistically-hiding commitment from any one-way function. In: Proc. 39th ACM Symposium on Theory of Computing, pp. 1–10 (2007)
19. Harnik, D., Naor, M.: On the compressibility of NP instances and cryptographic applications. In: Proc. 47th IEEE Symposium on Foundations of Computer Science, pp. 719–728 (2006)
20. Impagliazzo, R., Luby, M.: One-way functions are essential for complexity based cryptography. In: Proc. 30th IEEE Symposium on Foundations of Computer Science, pp. 230–235 (1989)

21. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Proc. 21st ACM Symposium on Theory of Computing, pp. 44–61 (1989)
22. Kilian, J.: Founding cryptography on oblivious tranfer. In: Proc. 20th ACM Symposium on Theory of Computing, pp. 20–31 (1988)
23. Naor, M., Ostrovsky, R., Venkatesan, R., Yung, M.: Perfect zero-knowledge arguments for NP using any one-way permutation. Journal of Cryptology 11(2), 87–108 (1998)
24. Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: Proc. 12th ACM-SIAM Symposium on Discrete Algorithms, pp. 448–457 (2001)
25. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008)
26. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Proc. 40th ACM Symposium on Theory of Computing, pp. 187–196 (2008)
27. Rabin, M.: How to exchange secrets by oblivious transfer, Technical Report TR-81, Aiken Computation Laboratory, Harvard University (1981)
28. Reingold, O., Trevisan, L., Vadhan, S.P.: Notions of reducibility between cryptographic primitives. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 1–20. Springer, Heidelberg (2004)
29. Shamir, A.: How to share a secret. Communications of the ACM 22(11), 612–613 (1979)
30. Shannon, C.: Communication theory of secrecy systems. Bell System Technical Journal 28(4), 656–715 (1949)
31. Wiesner, S.: Conjugate coding. SIGACT News 15(1), 78–88 (1983)
32. Wolf, S., Wullschleger, J.: Oblivious transfer is symmetric. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 222–232. Springer, Heidelberg (2006)
33. Wullschleger, J.: Oblivious-transfer amplification. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 555–572. Springer, Heidelberg (2007)