Reducing Complexity Assumptions for Oblivious Transfer

K.Y. Cheong Takeshi Koshiba

Division of Mathematics, Electronics and Informatics,
Graduate School of Science and Engineering, Saitama University
255 Shimo-Okubo, Sakura, Saitama 338-8570, Japan.
Email: {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, is 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 but not trapdoor functions in general. In this paper, we enhance previous results and show one Oblivious Transfer protocol based on a collection of trapdoor functions with some extra properties. We also provide reasons for adding the extra properties and argue that the assumptions in the protocol are nearly minimum.

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 [24] 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 [25] 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 on σ_{1-i}. The former property is called *receiver's privacy* and the latter *sender's privacy.* Later, Crépeau [6] 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. [3, 4, 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 [20, 28]. By simple arguments it can be seen that, in 1-out-of-2 OT, 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 [26] implies that we have the freedom to choose which side to protect in which way when we are given a protocol.

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 [22] based on Diffie and Hellman [9] type of problems.

1.2 Complexity Assumptions of OT

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 imply the existence of one-way functions [18], which in particular implies $P \neq NP$.

On the other hand, since it may be impossible to avoid all the computational assumptions, we would like to construct protocols based upon as weak 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 [21], this assumption has been reduced in [15]. And finally, Haitner and Reingold [16] 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 [17], 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 combination of the oracle separation [19] between one-way permutations and key agreement and the construction [2, 24] of key agreements from OT suggests that black-box reductions from OT to one-way functions are impossible. 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 [13], 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 [23], 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 [13]. 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 that is necessary for OT. For example, Bellare et al. showed that many-to-one trapdoor functions with super-polynomial 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 [13] can be improved to make it applicable to *general* trapdoor functions. The permutation property is thus not essential. This fact is actually discussed in the concluding remarks of [13]. But the trapdoor functions used in our protocol have some extra properties with respect to pre-image size and length expansion, and we argue that these extra properties are necessary and are close to the minimum in black-box reductions. Consequently, we have an OT construction based on a weaker assumption than the previous results.

2 Preliminaries

2.1 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. In [14], it is shown that such a construction can be done in the black-box way, where the semi-honest protocol is used as a black-box.

These known constructions of protocols for the malicious model from the semi-honest model are based on commitment schemes and zero-knowledge proofs. Regarding to complexity assumptions, they also require the existence of one-way functions. Using the combination of these results, we can obtain OT in the general model simply by constructing a semi-honest OT protocol.

2.2 1-out-of-2 Bit OT

In this paper, we consider only the 1-out-of-2 bit OT in the semi-honest model. 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}, where 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. The privacy properties of OT can be described 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. On the other hand, as it is impossible to protect both parties perfectly, some computational assumptions must be introduced.

In our basic protocol, the receiver's privacy is protected in information theoretic sense. It is compatible with the standard definition, and our analysis is much simplified by the information theoretic arguments.

[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.

2.3 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,

$$|\Pr[M(V_R(\sigma_i, 1, i)) = 1] - \Pr[M(V_R(\sigma_i, 0, i)) = 1]| < \epsilon_2. \qquad (3)$$

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

$$|\Pr[M(V_S(\sigma_0, \sigma_1, 0)) = 1] - \Pr[M(V_S(\sigma_0, \sigma_1, 1)) = 1]| < \epsilon_3. \qquad (4)$$

Note that, under our definition, a $(neg(n), neg(n), neg(n))$-WOT is equal to OT, in either the semi-honest model or the general model.

2.4 Pairwise Independent Hash Functions

Let H_n be a family of functions where the length of input l_1 and length of output l_2 are both in polynomial in n. From [5] it is well known that, for any choice of l_1 and l_2, there exists an efficient family of pairwise independent hash functions H_n with the following properties.

1. There exists a polynomial-time algorithm to sample $h \in H_n$ uniformly.

2. There exists a polynomial-time algorithm to evaluate $h(x)$ given h and $x \in \{0,1\}^{l_1}$.

3. When h is uniformly sampled, for every distinct $x_1, x_2 \in \{0,1\}^{l_1}$ and every $y_1, y_2 \in \{0,1\}^{l_2}$,

$$\Pr[h(x_1) = y_1 \wedge h(x_2) = y_2] = \frac{1}{2^{2l_2}}. \qquad (5)$$

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.

3.1 Collection of Dense Trapdoor Functions

In general, a collection of (non-injective) trapdoor functions F_n, where n is the security parameter, have the following properties:

1. There exists an efficient algorithm which uniformly selects a function f_α in F_n, 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.

3.2 The Extra Properties

In this paper, in order to construct our OT protocol, we require the trapdoor functions to have a few more properties. We list them here and call them the Five Extra Properties, in order to distinguish our trapdoor functions from the general ones.

1. Without loss of generality, we assume $D_\alpha \subset \{0,1\}^n$. For all $x \in \{0,1\}^n$ we assume $f_\alpha(x)$ can be evaluated using the same algorithm evaluating the function, and the algorithm will halt in polynomial time, producing

some output. That is, even if $x \notin D_\alpha$ we assume the algorithm will still run and produce a string as output. As we do not assume that the algorithm can detect the fact of $x \notin D_\alpha$, we assume nothing about the output string.

2. For all $y \in \{0,1\}^m$, 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. The idea is similar to Property 1 above.

3. There exist a polynomial $p(n)$ such that, for all α, the set D_α is dense in $\{0,1\}^n$. That is,
$$\frac{|D_\alpha|}{2^n} > \frac{1}{p(n)}. \tag{6}$$

4. For all $x \in D_\alpha$ we have $f_\alpha(x) \in \{0,1\}^m$ for some fixed $m = n + O(\log n)$. That is, the expansion (in terms of the length of strings) of the function is in order of $\log n$. This assumption can be relaxed slightly that only a majority of $x \in D_\alpha$ have this property. To be more precise, as long as those $x \in D_\alpha$ having this property are dense in D_α, they are also dense in $\{0,1\}^n$ due to Property 3 above. In that case we can restrict the domain of the trapdoor function to this new set of x, without affecting any other property of the trapdoor function.

5. 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 exist a polynomial $q(n)$ that, for all α and y,
$$\begin{aligned} I_{\alpha,y} &= \{x \in D_\alpha : f_\alpha(x) = y\} && (7) \\ |I_{\alpha,y}| &\leq q(n). && (8) \end{aligned}$$

3.3 Reasons for Extra Properties

Among the Five Extra Properties, Property 1 and 2 are general clarifications and may be assumed to be true anyway. Property 3 is adopted from [13], and we find that in our protocol it is still necessary in order to sample the elements in the function domain.

Property 4, the expansion property, is related to [11], which proves that OT cannot be black-box reduced to public key encryption or trapdoor function without any assumption. The proof is constructed relative to a world with a PSPACE-complete oracle. In this world one special trapdoor function exists, but OT does not exist. The special trapdoor function 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.

Note that OT can be reduced to public key encryption if it is possible to sample its valid ciphertexts, separately from the corresponding plaintexts. Therefore, the impossibility results are shown relative to a world where the only public key encryption does not have this property.

As OT cannot be black-box reduced to trapdoor functions which is length-expanding in $O(n)$, we attempt to build the OT with a trapdoor function which is at most length-expanding in $O(\log n)$.

Property 5, the pre-image property, is due to [1], where non-injective trapdoor functions are studied. In [1], a trapdoor function with exponential pre-image size is black-box constructed from a one-way function. On the other hand, it is known that OT cannot be black-box reduced to one-way function [19]. This, combined with the recent results of black-box construction of OT from semi-honest OT [14], implies that semi-honest OT cannot be black-box constructed from a trapdoor function with exponential pre-image size.

In [1], it is also shown that a trapdoor function with polynomial pre-image size is sufficient to construct public key encryption. Therefore, we are motivated to build our OT protocol with a trapdoor function of polynomial pre-image size.

4 The Protocol

The construction of our OT protocol is similar to [13], that a semi-honest Weak OT protocol is first constructed. After that, the process to enhance it to a semi-honest OT is exactly the same as [13].

First of all, we select a collection of pairwise independent hash functions

H_n with domain $\{0,1\}^n$ and range $\{1,2,\ldots,g(n)p(n)q(n)\}$ where $g(n) > 1$ is a polynomial of our choice which will be discussed in the next two sections. 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 $r_{1-i} \in \{0,1\}^m$.

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 the following conditions are satisfied.

$$f_\alpha^{-1}(t, r_j) \in \{0,1\}^n \tag{9}$$

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

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

$$v_j = h(f_\alpha^{-1}(t, r_j)). \tag{11}$$

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, where the sender does not know which one.

8. For both $j = 0, 1$ the sender chooses $y_j \in \{0,1\}^n$ uniformly and sets

$$c_j = \sigma_j \oplus b(f_\alpha^{-1}(t, r_j), y_j) \tag{12}$$

   where $b(x, y)$ is the inner product of x, y modulus 2, a hardcore predicate.

9. The sender sends (c_0, c_1, y_0, y_1) to the receiver.

10. The receiver outputs $\sigma_i' = b(s, y_i) \oplus c_i$. This is the secret required.

5 Analysis of Protocol

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

$$\begin{aligned} D_\alpha' &= \{x \in D_\alpha : x = f_\alpha^{-1}(t, f_\alpha(x))\} && (13) \\ R_\alpha &= f_\alpha(D_\alpha) = f_\alpha(D_\alpha') && (14) \end{aligned}$$

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.

$$\begin{aligned} D_\alpha'' &= \{x \in \{0,1\}^n : x = f_\alpha^{-1}(t, f_\alpha(x)) \wedge f_\alpha(x) \in \{0,1\}^m\} && (15) \\ R_\alpha'' &= f_\alpha(D_\alpha''). && (16) \end{aligned}$$

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 $m = n + O(\log n)$, R_α is dense in $\{0,1\}^m$. To be more precise, in our protocol we have

$$\Pr(s \in D_\alpha') > \frac{1}{p(n)q(n)} \tag{17}$$

$$\Pr(r_{1-i} \in R_\alpha) > \frac{1}{p(n)q(n)n^c} \tag{18}$$

for some constant c.

In an 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 property and pre-image property 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 $s = f_\alpha^{-1}(t, r_i)$.

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

$$\begin{aligned}
\Pr(s = f_\alpha^{-1}(t, r_i)) &> \frac{1}{p(n)q(n)} && (19)\\
\Pr(s \neq f_\alpha^{-1}(t, r_i) \wedge h(s) = v_i) &< (1 - \frac{1}{p(n)q(n)})(\frac{1}{g(n)p(n)q(n)}) && (20)
\end{aligned}$$

and the remaining probability is that the iteration gets absorbed. Thus, the probability of correctness, given that the iteration is not absorbed, would be

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

$$> \quad 1 - \frac{1}{g(n)} \tag{21}$$

as $p(n) \geq 1$ and $q(n) \geq 1$. 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, $r_{1-i} \in R''_\alpha$ if the protocol is run through the end in an iteration. 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. As a result, the weakness parameter for receiver's privacy is bounded by the same events that determine correctness, and thus $\epsilon_3 \leq 1/g(n)$.

At this point, it is important to see that 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 the Weak OT protocol is on the sender's privacy. After all, r_0 and r_1 are finally not even guaranteed to be in R_α. We can assume nothing about the computational hardness of inverting function f_α in that case.

But if $r_{1-i} \in R_\alpha$, the sender's privacy is maintained. In this case it is easy to see that, if the receiver has non-negligible advantage in guessing σ_{1-i} then he also has non-negligible advantage in getting $f_\alpha^{-1}(t, r_{1-i})$, in violation of our computational assumption.

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

$$\epsilon_2 \leq 1 - \frac{1}{p(n)q(n)n^c} \tag{22}$$

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

6 Strengthening the Weak OT

As a result, we have a $(\frac{1}{g(n)} + neg(n), 1 - \frac{1}{t(n)}, \frac{1}{g(n)})$-WOT, where $t(n) = p(n)q(n)n^c$. In general, it is possible to strengthen a Weak OT [27] to a standard OT under some conditions, within either the semi-honest model or general model. For our protocol, the construction in [13] can be used, which involves a technique from [7]. The details of the process can be seen in the Appendix of this paper.

7 Concluding remarks

We believe the main contribution of this paper is two-fold. In some sense, we remove the permutation requirement in trapdoor functions for constructing OT. We show that trapdoor functions with some extra properties are sufficient. On the other hand, we argue that these extra properties may be hard to remove, considering previous black-box impossibility results.

References

[1] M. Bellare, S. Halevi, A. Sahai, and S. Vadhan: Many-to-one trapdoor functions and their relations to public-key cryptosystems, In *Advances in Cryptology – CRYPTO '98*, LNCS 1462, pp.283–299, 1998.

[2] M. Blum: How to exchange (secret) keys, *ACM Transactions of Computer Systems*, 1(2), pp.175–193, 1983.

[3] G. Brassard, C. Crépeau, and M. Santha: Oblivious transfers and intersecting codes, *IEEE Transactions on Information Theory*, 42(6), pp.1769–1780, 1996.

[4] G. Brassard, C. Crépeau, and S. Wolf: Oblivious transfers and privacy amplification, *Journal of Cryptology*, 16(4), pp.219–237, 2003.

[5] J. Carter and M. Wegman: Universal classes of hash functions, *Journal of Computer and System Sciences*, 18(2), pp.143–154, 1979.

[6] C. Crépeau: Equivalence between two flavours of oblivious transfer, In *Advances in Cryptology — CRYPTO '87*, LNCS 293, pp.350–354, 1988.

[7] C. Crépeau and J. Kilian: Weakening security assumptions and oblivious transfer, In *Advances in Cryptology — CRYPTO '88*, Springer, pp.2–7, 1990.

[8] 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.

[9] W. Diffie and M. Hellman: New directions in cryptography, *IEEE Transactions on Information Theory*, 22(6), pp.644–654, 1976.

[10] S. Even, O. Goldreich, and A Lempel: A randomized protocol for signing contracts, *Communications of the ACM*, 28(6), pp.637–647, 1985.

[11] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan: The relationship between public key encryption and oblivious transfer, In *Proc. 41st IEEE Symposium on Foundations of Computer Science*, pp.325–335, 2000.

[12] O. Goldreich: *Foundations of Cryptography, volume II*, Cambridge University Press, 2004.

[13] I. Haitner: Implementing oblivious transfer using collection of dense trapdoor permutations, In *Theory of Cryptography Conference 2004*, LNCS 2951, pp.394–409, 2004.

[14] I. Haitner: Semi-honest to malicious oblivious transfer - the black-box way, In *Theory of Cryptography Conference 2008*, LNCS 4948, pp.412–426, 2008.

[15] I. Haitner, O. Horvitz, J. Katz, C. Koo, R. Morselli, and R. Shaltiel: Reducing complexity assumptions for statistically-hiding commitment, In *Advances in Cryptology — EUROCRYPT 2005*, LNCS 3494, pp.58–77, 2005.

[16] I. Haitner and O. Reingold: Statistically-hiding commitment from any one-way function, In *Proc. 39th ACM Symposium on Theory of Computing*, pp.1–10, 2007.

[17] D. Harnik and M. Naor: On the compressibility of NP instances and cryptographic applications, In *Proc. 47th IEEE Symposium on Foundations of Computer Science*, pp.719–728, 2006.

[18] R. Impagliazzo and M. Luby: One-way functions are essential for complexity based cryptography, In *Proc. 30th IEEE Symposium on Foundations of Computer Science*, pp.230–235, 1989.

[19] R. Impagliazzo and S. Rudich, Limits on the provable consequences of one-way permutations, In *Proc. 21st ACM Symposium on Theory of Computing*, pp.44–61, 1989.

[20] J. Kilian: Founding cryptography on oblivious transfer, In *Proc. 20th ACM Symposium on Theory of Computing*, pp.20–31, 1988.

[21] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung: Perfect zero-knowledge arguments for NP using any one-way permutation, *Journal of Cryptology*, 11(2), pp.87–108, 1998.

[22] M. Naor and B. Pinkas: Efficient oblivious transfer protocols, In *Proc. 12th ACM-SIAM Symposium on Discrete Algorithms*, pp.448–457, 2001.

[23] C. Peikert and B. Waters: Lossy trapdoor functions and their applications, *Electronic Colloquium on Computational Complexity*, Report No.TR07-080, 2007.

[24] M. Rabin: How to exchange secrets by oblivious transfer, *Technical Report TR-81*, Aiken Computation Laboratory, Harvard University, 1981.

[25] S. Wiesner: Conjugate coding, *SIGACT News*, 15(1), pp.78–88, 1983.

[26] S. Wolf and J. Wullschleger: Oblivious transfer is symmetric, In *Advances in Cryptology — EUROCRYPT 2006*, LNCS 4004, pp.222–232, 2006.

[27] J. Wullschleger: Oblivious-transfer amplification, In *Advances in Cryptology — EUROCRYPT 2007*, LNCS 4515, pp.555–572, 2007.

[28] A. C.-C. Yao: Protocols for secure computations, *Proc. 23rd IEEE Symposium on Foundations of Computer Science*, pp.160–164, 1982.

A The Strengthening of the Weak OT

The following construction is designed to strengthen a $(\epsilon_1, \epsilon_2, \epsilon_3)$-WOT with $(\epsilon_1, \epsilon_2, \epsilon_3) = (\frac{1}{g(n)} + neg(n), 1 - \frac{1}{t(n)}, \frac{1}{g(n)})$. While ϵ_1 and ϵ_3 are subjected to our choice of $g(n)$, ϵ_2 depends on the density parameters of the trapdoor function. It is relatively larger and we handle it first. As the process is the same as [13], the choice of $g(n)$ can be the same. As illustrated in the following, it works for $g(n) = 3n^2t(n)$.

A.1 The Second Parameter

We enhance the sender's privacy by breaking his secrets into many parts by a secret sharing scheme. Each secret σ_j is split into $nt(n)$ parts $\{\omega_{j,k}\}$, for $1 \leq k \leq nt(n)$. The following conditions are satisfied:

1. $\omega_{j,1} \ldots \omega_{j,nt(n)-1}$ are uniformly chosen from $\{0, 1\}$.

2. $\omega_{j,nt(n)} = (\bigoplus_{k=1}^{nt(n)-1} \omega_{j,k}) \oplus \sigma_j$.

The pairs $\{\omega_{0,k}, \omega_{1,k}\}$ are then sent by the $(\frac{1}{g(n)} + neg(n), 1 - \frac{1}{nt(n)}, \frac{1}{g(n)})$-WOT system. As the receiver can only get the secret σ_i by getting $\{\omega_{i,k}\}$ for all k, this process enhances sender's privacy. It produces a $(\frac{nt(n)}{g(n)} + neg(n), neg(n), \frac{nt(n)}{g(n)})$-WOT system, where the second parameter is negligible. Note that the first and third parameters of the WOT are increased for no more than $nt(n)$ times. The running time of the protocol is also increased for $nt(n)$ times.

A.2 The First Parameter

Next, the correctness is enhanced by a repeated run of the WOT resulted from the last step, and the correct value is decided by the majority rule.

We get a $(neg(n), neg(n), \frac{n^2 t(n)}{g(n)})$-WOT protocol by running the $(\frac{nt(n)}{g(n)} + neg(n), neg(n), \frac{nt(n)}{g(n)})$-WOT protocol n times. While ϵ_1 becomes negligible, ϵ_3 increases no more than n times. The running time also increases n times.

A.3 The Third Parameter

The last step is a technique from [7] in which an OT system is constructed out of a repeated run of a WOT which is weak in terms of the third parameter only. At the end, only the XOR of all the receiver's choices is his real choice. The protocol is:

1. Sender chooses a constant μ and generates a list of $\mu - 1$ random bits $(\phi_{0,1} \ldots \phi_{0,\mu-1})$.
2. Sender sets $\phi_{0,\mu} = \sigma_0 \oplus \bigoplus_{k=1}^{\mu-1} \phi_{0,k}$.
3. Sender sets the second list of bits as $\phi_{1,k} = \phi_{0,k} \oplus \sigma_0 \oplus \sigma_1$ for all k.
4. The two parties use the $(neg(n), neg(n), \frac{n^2 t(n)}{g(n)})$-WOT for μ times to transfer each pair of $(\phi_{0,k}, \phi_{1,k})$.
5. The receiver makes the choices randomly, except that the XOR of all choices represents the real choice. That is, denoting the choices by i_k for $1 \leq k \leq \mu$, we have
$$i = \bigoplus_{k=1}^{\mu} i_k. \tag{23}$$
6. The final output of the receiver is σ_i, as it can be computed
$$\sigma_i = \bigoplus_{k=1}^{\mu} \phi_{i_k,k}. \tag{24}$$

In this protocol, if the sender tries to guess the final choice i of the receiver, he has to guess each of the i_k correctly. The probability of the sender being able to do so drops exponentially with μ. By selecting a suitable μ linear in n, we get a $(neg(n), neg(n), neg(n))$-WOT out of the $(neg(n), neg(n), \frac{n^2 t(n)}{g(n)})$-WOT. The running time is increased by μ times. This is our final OT protocol as all three weakness parameters are now negligible.