One-Round Witness Indistinguishability from Indistinguishability Obfuscation

Qihua Niu[1,2], Hongda Li[1], Bei Liang[1], Fei Tang[1]

[1] State Key Lab of Information Security, Institute of Information Engineering
Chinese Academy of Sciences, Beijing 100093
[2] School of Science, China University of Petroleum, Qingdao 266580
{niuqihua,lihongda,liangbei,tangfei}@iie.ac.cn

Abstract. In this work, we explore the connection between witness indistinguishability (WI) and indistinguishability obfuscation ($i\mathcal{O}$). We construct a one-round witness indistinguishable protocol for all of **NP** based on the the existence of indistinguishability obfuscator (the first candidate construction of indistinguishability obfuscator was recently put forward by Garg et.al. in 2013). Based on our one-round WI, we also construct a two-round oblivious transfer (OT) protocol and by a slight modification of our OT protocol, we get a noninteractive bit commitment scheme.

Key word: witness indistinguishability, indistinguishability obfuscation, commitment scheme, oblivious transfer.

1 Introduction

Witness Indistinguishability. The concept of witness indistinguishability, proposed by Feige and Shamir [13] is a meaningful relaxation of zero knowledge (ZK) which is introduced by Golewasser, Micali and Rackoff in their seminal paper [20].

ZK protocols allow proving a statement without revealing anything but its validity. That is, what the verifier can get from the execution of the ZK protocol can also be simulated from the statement itself. ZK maintains the most comprehensive privacy of the prover. Feige and Shamir [13] considered two relaxations of ZK: witness indistinguishable and witness hiding (WH) protocols. Sometimes, WI and WH are enough for many cryptographic applications. In this paper, we focus on WI protocols.

A protocol is WI if any two proofs for the same statement that use two different witnesses are indistinguishable. ZK protocols are WI protocols. However, WI protocols do not always guarantee witness secrecy, in particular, for statements with unique **NP** witness, WI is meaningless (protocols in which the prover just send to the verifier the unique witness are trivially WI).

WI protocols play important roles in the design of ZK protocols. One example is the FLS-type ZK protocols, this kind of protocol uses FLS technique which is introduced by Feige, Lapidot and Shamir [12]. FLS technique allows to reduce the problem of constructing a ZK proof (or argument) system to the problem of constructing two simpler objects: a witness indistinguishable proof (argument) system and a generation protocol. The seminal paper of Barak [2] also use FLS-type technique.

Indistinguishability Obfuscation. Indistinguishability obfuscation is the relaxation of program obfuscation.

In 2001, Barak, Goldreich, Impagliazzo, Rudich, Sahai, Vadhan and Yang [3, 4] initiated the formal study of program obfuscation, which aims to make computer program "unintelligible" while preserving their functionality. The compiler that takes programs and makes them difficult to understand is called obfuscator. Ideally, an obfuscated program should be a "virtual black box (VBB)", in the sense that anything one can compute from it one could also compute from the input-output behavior of the program. Unfortunately, Barak et.al. showed that general-purpose obfuscation in the sense of VBB is impossible.

Motivated by this impossibility result, Barak et.al. proposed the less intuitive, but potentially realizable, notion of $i\mathcal{O}$, which requires only that obfuscations of any two distinct program with the same functionality be computationally indistinguishable from each other. In a recent breakthrough, Garg, Gentry, Halevi, Raykova, Sahai and Waters [15] proposed the first candidate construction of an efficient indistinguishability obfuscator (for the sake of simplicity of notation, which is also denoted by $i\mathcal{O}$) for general programs and also showed how to apply $i\mathcal{O}$ to achieve powerful new functional encryption schemes for general circuits.

After the occurrence of $i\mathcal{O}$, many problems get great breakthrough. Sahai and Waters gave many applications of $i\mathcal{O}$ [29]. They resolved the 16-year-old open question of *Deniable Encryption* posed by Canetti, Dwork, Naor, and Ostrovsky in 1997 [10]. They also build up several core cryptographic primitives from $i\mathcal{O}$: public key encryption, non-Interactive zero knowledge proofs (NIZKs), injective trapdoor functions,and 2-round semi-honest oblivious transfer. In addition, Garg et.al. present a compiler that transform any MPC protocol into a 2-round protocol in the CRS model, the main tool they use is $i\mathcal{O}$. Hohenberger et.al. [26] utilized the advances in $i\mathcal{O}$ to construct specific hash functions.

Oblivious Transfer. Oblivious transfer, first introduced by Rabin [28], is a central primitive in modern cryptography. It serves as the basis of a wide range of cryptographic tasks.

Oblivious transfer is a protocol between a sender, holding two bits x_0 and x_1, and a receiver holding a choice bit b. At the end of the protocol the receiver should learn the bit of his choice (i.e., x_b) but learn nothing about the other bit. The sender, on the other hand, should learn nothing about the receivers choice b.

Constructing round-efficient oblivious transfer protocols is an important task. There are several 2-round OT protocols. [1, 11, 27, 5] constructed a 2-round OT protocol based on the various number theoretic assumptions (with weaker security guarantees than the simulation based security).

Commitment Scheme. Bit commitment schemes are basic primitives in cryptography. A commitment scheme defines a two-stage protocol between a sender S and a receiver R; informally, after the commit stage, S is bound to (at most) one value, which stays hidden from R, and in the reveal stage R learns this value. The two security properties hinted at in this informal description are known as binding (namely, that S is bound to at most one value after the commit stage) and hiding (namely, that R does not learn the value to which S commits before the reveal stage).

Commitment schemes consist of interactive and noninteractive schemes. In an interactive commitment scheme, the sender and the receiver are allowed to interact during the commitment and decommitment steps. The definition and examples of interactive commitment scheme can be found in [17]. In a noninteractive commitment scheme, both the two stages consist of one round, that is a message sent from the sender to the receiver. Noninteractive commitment has the minimum number of rounds, but it may depend on stronger assumption.

1.1 Our Results

In this work, we study the relation between WI and indistinguishability obfuscation. We construct a one-round witness indistinguishable protocol for all **NP** under the assumption of the existence of indistinguishability obfuscator for general circuit class. Our construction is different from that of Barak, Ong and Vadhan [5]. The one-round WI of [5] is based on the assumption of existence of efficient 1/2-HSG against co-nondeterministic circuits and the existence of trapdoor permutations. Barak et.al. [5] posed the issue: *Either constructions (of one-round WI) for specific problems based on specific assumptions or general constructions for all of **NP** based on alternative assumptions would be interesting.* Our construction is the second case.

Based on our one-round WI, we also construct a two-round oblivious transfer, and by a slightly modification of our OT protocol, we obtain a noninteractive bit commitment scheme.

1.2 Other Related Work

In light of the impossibility results of Barak et.al [3, 4], several followup works studied notions of obfuscation with relaxed security[22, 24, 25, 7]. There are also impossibility results with respect to auxiliary inputs [18, 19, 8]. An interesting phenomenon is that the possibility results of obfuscation drive the impossibility results. For example, in [19], their impossibility result is that: the existence of $i\mathcal{O}$ implies that all functions with sufficient pseudo-entropy cannot be obfuscated w.r.t dependent auxiliary input. The impossibility result in [8] is that: the existence of iO implies that all functions with sufficient pseudo-entropy cannot be obfuscated w.r.t independent auxiliary input.

The impossibility result of Barak et.al. stated that there exists an unobfuscatable (strong unobfuscatable) family of functions $\{f_k\}$ for which any program $\tilde{f}_k$ that computes approximate the same function as f_k leaks information that cannot be leaked, given only black-box access to f_k, assuming k is chosen at random. Bitansky and Paneth [9] put forth the notion of robust obfuscation and constructed robust unobfuscatable function family. That is, there exists family of functions $\{f_k\}$ for which any program $\tilde{f}_k$ that agree with the same function as f_k , with high-enough (noticeable) probability over inputs drawn from some specific distribution (such as uniform distribution) leaks information that cannot be leaked, given only black-box access to f_k, assuming k is chosen at random.

What's interesting is that Bitansky and Paneth [9] set up the relation between the negative results of obfuscation and positive results of ZK. They showed that unobfuscatable function family can be used to construct ZK protocols. In contrast, in this paper, we build the relation between the relaxations of obfuscation and ZK. Concretely, we find that the existence of $i\mathcal{O}$ implies one-round WI.

1.3 Outline

We start with reviewing the definitions of known cryptography primitives in section2. Later in section 3, we present our construction of a one-round WI protocol, and then, in section 4 we constructs a 2-round oblivious transfer protocol. Finally, a noninteractive commitment scheme is put forth in Section 5.

2 Preliminaries

In this paper, we use the following standard definitions and tools.

2.1 Proof (argument) system

An interactive proof [20] is an interactive protocol in which a prover (with unlimited computational powers) tries to convince a probabilistic polynomial-time verifier the validity of a certain statement. Since interactive protocols are probabilistic, the soundness and completeness criteria are also probabilistic. The formal definition of interactive proof follows.

Definition 1 (interactive proofs system [20]). *An interactive protocol (P, V) is called an interactive proof system for a language L if the following conditions holds*

1. ***Efficiency.*** *On common input x, the number and total length of messages exchanged between P and V are bounded by a polynomial in $|x|$, and V is a probabilistic polynomial-time machine.*

2. ***Completeness.*** *If $x \in L$, then $Pr[(P, V)(x) = 1] \geq 2/3$.*

3. ***Soundness.*** *If $x \notin L$, then for any P^*, $Pr[(P^*, V)(x) = 1] \leq 1/3$.*

If P is restricted to be polynomial-time machine, then this system is called interactive argument system.

We say that an interactive proof system has perfect completeness if the completeness condition holds with probability 1 instead of $2/3$. We say that a system has perfect soundness if the soundness condition holds with probability 0 instead of $1/3$.

An interactive proof (argument) system is called *public-coin* if the verifier's messages consist only of random strings and acceptance is computed as a deterministic polynomial-time function of the interaction's transcripts. An interactive proof (argument) system is *private-coin* if it is not public-coin.

The number of rounds in an interactive proof is the total number of messages exchanged in the interaction. A proof system with one round is called noninteractive.

2.2 Witness indistinguishable protocol

Let L be a language, R_L the corresponding relation of L. For a statement x, $x \in L$ if and only if there exists w such that $(x, w) \in R_L$, and in this case, we say that w is a witness for x. Recall that the class **NP** is the class of language L such that the corresponding relation

R_L is decidable in time polynomial in the first input. If L is an **NP** language then we say that R_L is a *witness relation* of L.

The concept of *witness indistinguishability* was proposed by Feige and Shamir [13] as a relaxation of zero-knowledge. Unlike the case with zero-knowledge, witness indistinguishability is closed under parallel and concurrent composition.

Definition 2 (witness indistinguishability, [13, 5]). *Let L be an **NP** language with witness relation R_L. Let (P, V) be a proof system for L. We say that (P, V) is witness indistinguishable if for any polynomial-time verifier V^*, for all $x \in L$, for all w_1, w_2 such that $(x, w_i) \in R_L, i \in \{1, 2\}$, the view of V^* when interactive with $P(x, w_1)$ is computationally indistinguishable from its view of V^* when interactive with $P(x, w_2)$.*

Theorem 1 ([13]). *Every zero-knowledge protocol is witness indistinguishable. Witness indistinguishable is preserved under parallel and concurrent composition of protocols.*

ZAPs. A *zap* [11] is a 2-round public-coin interactive proof system that is witness indistinguishable.

Dwork and Naor proved the following theorem.

Theorem 2. *If trapdoor permutations (secure against polynomial-size circuits) exist, then every language in **NP** has a ZAP.*

2.3 Indistinguishability obfuscation

Indistinguishability obfuscation is the weak version of program obfuscation. Firstly, let's recall the concept of *Program obfuscation*. The formal study of program obfuscation was initiated by Barak, Glodreich, Impagliazzo, Rudich, Sahai, Vadhan, and Yang in their seminal work [3, 4]. Roughly speaking, program obfuscation aims to make a computer program "unintelligible" while preserving its functionality. Unfortunately, Barak et.al. showed that the most natural simulation-based formulation of program obfuscation (a.k.a. "black-box obfuscation") is impossible to achieve for general program.

Due to the impossibility results of general program obfuscation, Barak et.al. [3, 4] suggested the weaker obfuscation, *indistinguishability obfuscation*. Informally, an indistinguishability obfuscator for a class of circuits $\mathcal{C}$ guarantees that for any two circuits C_0 and C_1 that are "functionally equivalent" (i.e., for all inputs x in the domain, $C_0(x) = C_1(x)$), the obfuscation of C_0 must be computationally indistinguishable from the obfuscation of C_1. Below we present the formal definition following the syntax of [15].

Definition 3 (indistinguishability obfuscation [15]). *A uniform PPT machine $i\mathcal{O}$ is called an indistinguishability obfuscator for a circuit class $\{\mathcal{C}_\lambda\}$ if the following conditions are satisfied:*

1. ***Preserving Functionality.*** *For all security parameters $\lambda \in \mathbb{N}$, for all $C \in \mathcal{C}_\lambda$, for all inputs x, we have that*

$$Pr[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(\lambda, C)] = 1$$

2. ***Indistinguishability.*** *For any (not necessarily uniform) PPT distinguisher D, there exists a negligible function $negl(\cdot)$ such that the following holds: For all security parameters $\lambda \in \mathbb{N}$, for all pairs of circuits $C_0, C_1 \in \mathcal{C}_\lambda$, we have that if $C_0(x) = C_1(x)$ for all inputs x, then*

$$|Pr[D(i\mathcal{O}(\lambda, C_0)) = 1] - Pr[D(i\mathcal{O}(\lambda, C_1)) = 1]| \leq negl(\lambda)$$

Garg et.al. [15], building upon a variant of the multilinear maps framework, gave the first candidate construction for a general-purpose obfuscator satisfying this notion.

2.4 Witness encryption

Here we recall the notion of Witness Encryption for **NP** recently introduced by Garg et.al. [16]. Just like what they stated, a witness encryption scheme is defined for an **NP** language L (with corresponding witness relation R_L). In such a scheme, a user can encrypt a message M to a particular problem instance x (which acts as encryption key) to produce a ciphertext. A recipient of a such ciphertext is able to decrypt the message if x is in the language and the recipient knows a witness w (the decryption key) of x (i.e. $(x, w) \in R_L$). However, if x is not in the language, then no polynomial-time attacker can distinguish between encryptions of any two messages of equal length. The encrypter himself may have no idea whether x is actually in the language.

Definition 4 (witness encryption). *A witness encryption scheme for an **NP** language L (with corresponding witness relation R_L) consists of the following two polynomial-time algorithms:*

- ***Encryption***. *The algorithm Encrypt($1^\lambda, x, M$) take as input a security parameter 1^λ, a statement $x \in L$, and a bit $b \in \{0, 1\}$, and outputs a ciphertext CT.*
- ***Decryption***. *The algorithm Decrypt(CT,w) takes as inputs a ciphertext CT and an string w, and outputs a bit b' or the symbol $\perp$.*

The algorithms satisfy the following two conditions:

- ***Correctness***. *For any security parameter λ, for any $b \in \{0, 1\}$, and for any $x \in L$ such that $(x, w) \in R_L$, we have that*

$$Pr[Decrypt(Encrypt(1^\lambda, x, b), w) = b] = 1 - negl(\lambda).$$

- ***Soundness Security***. *For any $x \notin L$, for any PPT adversary $\mathcal{A}$:*

$$Pr[\mathcal{A}(Encrypt(1^\lambda, x, 0)) = 1] - Pr[\mathcal{A}(Encrypt(1^\lambda, x, 1)) = 1] \leq negl(\lambda).$$

where $negl(\cdot)$ is some negligible function.

It is clear that the above scheme can be used to encrypt a string. Let $b = b_1 b_2 ... b_\lambda$ is a string of length λ, then the ciphertext of b is $c = c_1 c_2 ... c_\lambda$ where c_i is the ciphertext of b_i. The decryption is similar.

We can treat a witness encryption scheme as a special public encryption scheme. The public key is a string x. If $x \in L$, the corresponding secret key is any w such that $(x, w) \in R_L$;

if $x \notin L$, there is no secret key (this is a bit different from the traditional public key cryptosystem (PKC), in traditional PKC, the public/secret key pair is generated by a key generation algorithm, every public key corresponds to a secret key). There is no secrecy guarantee for cipertext under public key x in L.

Witness encryption scheme is implied by indistinguishability obfuscation. Garg et al. [15] presented a construction of witness encryption for an **NP**-Complete language from indistinguishability obfuscation by applying it on "point filter function" [18].

3 One-round Witness indistinguishable from indistinguishability obfuscation

In this section, we construct a one-round (noninteractive but without common reference strings) witness indistinguishable proof system for **NP** from indistinguishability obfuscation. Witness indistinguishable means that the prover proves the validity of a statement with some witness, but the verifier cannot distinguish which witness the prover uses. Indistinguishability obfuscator of a circuit class is a program that takes as input a circuit of the circuit class such that when the input circuits have the same functionality, it is hard to distinguish which input $i\mathcal{O}$ has used. The same meaning of indistinguishability in the two concepts motive us to explore the relation between "witness indistinguishability" and "indistinguishability obfuscator".

In the following one-round witness indistinguishable protocol for **NP** language L, we need to use a witness encryption scheme for L (with the corresponding witness relation R_L) and a indistinguishability obfuscator $i\mathcal{O}$ for the decryption circuit class. We denote the encryption and decryption algorithms of the witness encryption scheme by $\mathit{Encrypt}(1^\lambda, \cdot, \cdot)$ and $\mathit{Decrypt}(\cdot, \cdot)$ respectively and we denote $i\mathcal{O}$ the indistinguishability obfuscator of the circuit class $\{\mathcal{C}_\lambda\}$ where $\mathcal{C}_\lambda$ consists of the circuits $C_{x,w}$ with the functionality $C_{x,w}(\cdot) = \mathit{Decrypt}(\cdot, w)$ and $|x| = \lambda$. Notice here, if $(x, w) \notin R_L$, $C_{x,w}(\cdot)$ is a constant circuit that always outputs $\perp$, it has no decryption functionality. For the convenience of expression, we call the circuit class $\{\mathcal{C}_\lambda\}$ as decryption circuit class.

Now, we present our one-round witness indistinguishable protocol.

One-round WI Proof for $L \in$ NP. On common input $x \in \{0,1\}^\lambda$ and auxiliary input w for the prover, such that $(x, w) \in R_L$, do the following.

Prover's message
1. Construct the circuit $C_{x,w}(\cdot)$ with the functionality $C_{x,w}(\cdot) = \mathit{Decrypt}(\cdot, w)$.
2. Compile the circuit $C_{x,w}(\cdot)$ by the indistinguishability obfuscator $i\mathcal{O}$ of $\{\mathcal{C}_\lambda\}$ and get $T_w = i\mathcal{O}(C_{x,w})$.
3. Send to the verifier T_w.

Verifier's Test
1. Choose randomly the string $r \in \{0,1\}^\lambda$ and compute $c = \mathit{Encrypt}(1^\lambda, x, r)$.
2. Feed c to T_w and get the output $T_w(c)$. Accept if $T_w(c) = r$, and reject otherwise.

Theorem 3. *The above protocol is a one-round witness indistinguishable proof system if indistinguishability obfuscator for general circuit class exists.*

Proof. The assumption that indistinguishability obfuscator for general circuit class exists implies that witness encryption scheme exists, and the indistinguishability obfuscator of the decryption circuit class $\{\mathcal{C}_\lambda\}$ exists, we denote it by $i\mathcal{O}$.

We need to prove that the protocol is (1) complete, (2) sound, and (3) witness indistinguishable.

Firstly, if the prover and the verifier are both honest, then for ciphertext c under the key x of any plaintext r that the verifier chooses, the verifier can get the exact decryption of c by feeding c to the received circuit T_w which has the same functionality as the decryption circuit $C_{x,w}(\cdot)$ except for a negligible probability.

Secondly, the verifier test procedure is executed by feeding to the obfuscated decryption circuit a ciphertext of a random string r of length λ. By the soundness security of the witness encryption scheme, if $x \notin L$, then for any prover's message, the verifier rejects with overwhelming probability.

At last, for witnesses w_0, w_1 of x, that is $(x, w_0), (x, w_1) \in R_L$, if the verifier can distinguish $T_{w_0} = i\mathcal{O}(C_{w_0})$ from $T_{w_1} = i\mathcal{O}(C_{w_1})$ with noticeable probability, then the verifier algorithm can be used as the distinguisher to distinguish $i\mathcal{O}(C_{w_0})$ from $i\mathcal{O}(C_{w_1})$, and this contradicts the indistinguishability property of $i\mathcal{O}$.

Remark. It is clear that both of the prover and the verifier can not get additional benefits by resetting the other party. So our WI protocol has the property of resettable-soundness as well as resettable-witness indistinguishability.

4 Two-round oblivious transfer from one-round WI

In this section, we construct a two-round 1-out-of-2 oblivious transfer (OT) protocol. Informally speaking, a 1-out-of-2 OT protocol consists of two parties, one is called Sender with input $x_0, x_1 \in \{0,1\}$ and the other is called Receiver with input $b \in \{0,1\}$. The goal of the protocol for the Receiver is to receive the bit x_b without the Sender knowing Receiver's input b, the goal of the protocol for the Sender is to let the Receiver receive at most one of x_0 and x_1.

Dwork and Naor [11] presented a 3-round OT protocol in the standard model, their OT protocol is constructed based on the *Quadratic Residuosity Assumption* [21]. The first two rounds of Dwork and Naor's 3-round OT protocol are their 2-round WI protocol. Barak, Ong, and Vadhan [5] gave their 2-round OT protocol by replacing Dwork-Naor's 2-round WI sub-protocol with a 1-round WI sub-protocol which is constructed by Barak,Ong, and Vadhan based on the existence of trapdoor permutations and the existence of an efficient 1/2-*HSG* against co-nondeterministic circuits. Our 2-round OT protocol is obtained by replacing Dwork, and Naor's 2-round WI with our own 1-round WI that is based on the existence of indistinguishability obfuscator for general circuit class.

Now, we formally give the definition of 1-out-of-2 oblivious transfer [5], let $output_S(S(x_0, x_1), R(b))$ be the output of Sender S (on input $x_0, x_1 \in \{0,1\}$) after interacting with Receiver R (on input $b \in \{0,1\}$), we define $output_R(S(x_0, x_1), R(b))$ similarly.

Definition 5 (1-out-of-2 OT [5]). *An 1-out-of-2 OT protocol consists of a polynomial-time Sender S and a polynomial Receiver R, satisfying the following conditions.*

- ***completeness***. *For all $x_0, x_1, b \in \{0,1\}$, we have that $Pr[output_R(S(x_0, x_1), R(b)) = x_b] > 1 - negl(k)$, where $negl(\cdot)$ is a negligible function and k is the security parameter and the probability is over the random choices of S and R.*

- ***computational privacy of receiver***. *For all probabilistic polynomial-time S^*, we have that $output_{S^*}(S^*, R(0))$ is computationally indistinguishable from $output_{S^*}(S^*, R(1))$.*

- ***statistical privacy of sender***. *For every deterministic receiver strategy R^*, one of the two following conditions holds:*

  1. *$output_{R^*}(S(0, x), R^*)$ is statistically indistinguishable from $output_{R^*}(S(1, x), R^*)$ for any $x \in \{0,1\}$, or*
  2. *$output_{R^*}(S(x, 0), R^*)$ is statistically indistinguishable from $output_{R^*}(S(x, 1), R^*)$ for any $x \in \{0,1\}$.*

In what follows, we show our 2-round OT protocol which is obtained by replacing the 2-round WI with 1-round WI in Dwork-Naor's OT protocol.

Two-round OT Protocol. The Sender has inputs $x_0, x_1 \in \{0,1\}$ and the Receiver has input $b \in \{0,1\}$, k is security parameter.

Receiver's message
1. Choose two random odd primes p, q of length k and let $N = pq$ and choose two random strings $y_0, y_1 \in \mathbb{Z}_N^*$ such that y_{1-b} is a quadratic residue (QR) modulo N and y_b is a non-residue with Jacobi symbol 1.
2. Make a WI proof π of the statement: " y_0 is a QR mod N or y_1 is a QR mod N".
3. Send to the Sender N, y_0, y_1 and π.

Sender's Message
1. Verify the validity of π. Abort if the verification fails; Otherwise, choose $z_0, z_1 \in_R \mathbb{Z}_N^*$.
2. Send to the Receiver the following two values in any order: $\{t_0 = y_0^{x_0} z_0^2 \bmod N, t_1 = y_1^{x_1} z_1^2 \bmod N\}$.

Verifier's decision
Determine whether the values t_0, t_1 received form the sender are quadratic residue or not (Notice here, the verifier knows the decomposition of N). If both of t_0 and t_1 are quadratic residue, it outputs 0, if one is residue and the other is non-residue, it outputs 1. Otherwise, it outputs $\perp$.

Theorem 4. *Suppose that there exists indistinguishability obfuscator for general circuit class and that the Quadratic Residuosity Assumption (QRA) holds, then the above protocol is a two-round oblivious transfer protocol.*

Proof. Firstly, assume that both Receiver and Sender follow the protocol correctly. Let y_i be the only quadratic non-residue modulo N among y_0, y_1, then $y_i^{x_i} z_i^2$ is a quadratic residue modulo N if and only if $x_i = 0$, the other value t_{1-i} is always quadratic residue. Because the Receiver have the decomposition of N, it can efficiently output x_i with probability 1.

Secondly, assume the Receiver follow the protocol correctly but the Sender does not, that is, the Sender is a probabilistic polynomial-time adversary denoted by S^*. We need to show that S^* can not distinguish the case when Receiver's input is 0 from the case when Receiver's input is 1. Because what S^* get from the Receiver is just the Receiver's message: N, y_0, y_1 and π where one of y_0 and y_1 is a quadratic residue and the other is a non-residue with Jacobi symbol 1 and π is a WI proof for the statement "y_0 is a QR mod N or y_1 is a QR mod N". By the *Quadratic Residue Assumption* and witness indistinguishability property of π, it is hard for polynomial-time S^* to determine whether y_0 and y_1 are quadratic non-residues, so it is hard for S^* to distinguish the case when Receiver's input is 0 from the case when Receiver's input is 1.

At last, assume that the Sender is honest, but the Receiver is not, that is, the Receiver R^* is polynomial-time adversary. By the soundness of the WI, at least one of y_0, y_1 is quadratic residue module N. Assume y_j is a quadratic residue module N, then $y_j^{b_j} z_j^2$ is always a quadratic residue module N, independent of x_j, and independent of how N is chosen. Thus R^* can learn at most one of x_0 x_1, that is, at least one of the following two cases hold:

1. $output_{R^*}(S(0,x), R^*)$ is statistically indistinguishable from $output_{R^*}(S(1,x), R^*)$ for any $x \in \{0,1\}$, or
2. $output_{R^*}(S(x,0), R^*)$ is statistically indistinguishable from $output_{R^*}(S(x,1), R^*)$ for any $x \in \{0,1\}$.

5 Noninteractive bit commitment scheme from one-round WI

By slightly modifying the OT protocol in section 4, we present a noninteractive bit commitment scheme in this section.

Commitment schemes are basic ingredients in many cryptographic protocols. They are used to enable a party to commit itself to a value while keeping it secret. In a later stage the commitment is opened, and it is guaranteed that the opening can yield only a single value determined in the commitment phase.

Loosely speaking, a commitment scheme is an efficient two-part protocol through which one party, called the sender, can commit itself to a value such that the following two conflicting requirements are satisfied. (1) hiding: at the end of the first phase, the other party, called the receiver, does not gain any knowledge of the sender's value. This requirement has to be satisfied even if the receiver tries to cheat. (2) binding: given the transcript of the interaction in the first phase, there exists at most one value that the receiver can later (i.e., in the second phase) accept as a legal opening of the commitment. This requirement has to be satisfied even if the sender tries to cheat.

There are two types of commitments: interactive commitment scheme and noninteractive commitment scheme. In an interactive commitment scheme, the sender and the receiver are allowed to interactive during the commitment and decommitment steps. The formal definition of interactive one can be found in [17]. In what follows, let's see the definition of noninteractive commitment scheme [5].

Definition 6 (noninteractive bit commitment [5]). *A noninteractive bit commitment scheme is a polynomial-time algorithm S which takes a bit $b \in \{0,1\}$ and a random $r_S \leftarrow \{0,1\}^{poly(k)}$. Where k is the security parameter, and outputs a commitment $C = S(b; r_S)$. The algorithm S must satisfy the following conditions:*

1. *(Hiding) The commitments to 0 and 1 are computationally indistinguishable. That is, the distributions $\{S(0; r_S)_{r_S \leftarrow \{0,1\}^{poly(k)}}\}$ and $\{S(1; r_S)_{r_S \leftarrow \{0,1\}^{poly(k)}}\}$ are computationally indistinguishable by probabilistic polynomial-time algorithms.*
2. *(Binding) There exists at most one value that the receiver can later (i.e., in the second phase) accept as a legal opening of the commitment.*

There are noninteractive bit commitment schemes constructed by Blum [6] based on any 1-1 one-way function and by Barak, Ong, and Vadhan [5] based on the assumption that there exists an efficient 1/2-HSG against co-nondeterministic uniform algorithms and that one-way functions exist. In what follows, we propose a noninteractive bit commitment scheme based on the existence of one-round WI and quadratic residue assumption. In our commitment scheme, even though our assumption is strong, but in the setting that our commitment scheme is a sub-protocol of another protocol (such as zero-knowledge protocol), and the outer protocol need the QRA assumption and one-round WI (e.g. for the purpose of minimizing the rounds of the protocol), our scheme is fine.

In what follows, we propose our noninteractive bit commitment scheme.

Noninteractive bit commitment scheme
Input to receiver R: 1^k, where k is the security parameter.
Input to sender S: 1^k and a bit $b \in \{0,1\}$.

Commitment stage:
1. Choose two random odd primes p, q of length k and let $N = pq$ and two random strings $y_0, y_1 \in \mathbb{Z}_N^*$ such that y_{1-b} is a quadratic residue (QR) modulo N and y_b is a non-residue with Jacobi symbol 1.
2. Make a WI proof π of the statement: "y_0 is a QR mod N or y_1 is a QR mod N".
3. Send to the Sender N, y_0, y_1 and π.

Decommitment stage:
S reveals the bit b and the decomposition of $N = pq$.

Theorem 5. *If there exists one-round WI and QRA holds, then the above protocol is a noninteractive commitment scheme.*

Sketch of the proof: From the soundness of WI there is at most one of y_0, y_1 is quadratic non-residue, so the biding property holds. The hiding property is from the QRA.

References

1. W. Aiello, Y. Ishai, O. Reingold. Priced Oblivious Transfer: How to Sell Digital Goods. In EUROCRYPT, pp 119-135, 2001.

2. B.Barak How to go beyond the black-box simulation barrier. In FOCS, pp 106-115, 2001.
3. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A Sahai, S.P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In CRYPTO, pp 1-18, 2001.
4. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S.P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. J.ACM, 59(2):6, 2012.
5. B. Barak, S. J. Ong and S. Vadhan. Derandomization in Crytography. IN CRYPTO, pp 299-315, 2003.
6. M. Blum. Coin flipping by phone. In 24th IEEE Computer Conference (CompCon). pp 133-137, 1982.
7. N. Biransky, and R. Canetti. On strong simulation and composable point obfuscation . IN CRYPTO, pp 520-537, 2010.
8. N.Bitansky, R.Canetti, O.Paneth, and A.Rosen. More on the impossibility of VBB obfuscation with auxiliary input. IACR Cryptology ePrint Archive, 2013:701, 2013.
9. N. Biransky, and O. Paneth. On the impossibility of approximate obfuscation and applications to resettable cryptography. IN STOC, pp 241-250, 2013.
10. R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky. Deniable Encrypton. In CRYPTO, pp 90-104, 1997.
11. C. Dwork and M. Naor. Zaps and their applications. In proceedings of the 41th Annual symposium on Foundations of Computer Science, Pages 283-293. ACM, 2000.
12. U. Feige, D. Lapidot and A. Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. SICOMP: SIAM Journal on Computing,29, 1999.
13. U. Feige and A. Shamir. Witness Indistinguishable and Witness Hiding Protocols. Proc. 22nd ACM Symposium on the Theorey of Computing, pp. 416-426, 1990.
14. S. Garg, C. Gentry, S.Halevi, M.Raykova. Two-round MPC from indistinguishability obfuscation. In TCC, pp 74-94, 2014.
15. S. Garg, C. Gentry, S.Halevi, M.Raykova, A.Sahai and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In FOCS, pp 40-49, 2013.
16. S. Garg, C. Gentry, A. Sahai and B. Waters. Witness encryption and its applications. In STOC, pp 467-476, 2013.
17. O. Goldreich. Foundations of Cryptography - Basic Tools. Cambridge University Press, 2001.
18. S. Goldwasser and Y. T. Kalai: On the impossibility of obfuscation with auxiliary input. In FOCS, pp 553-562, 2005.
19. S. Goldwasser, Y. T .Kalai. A Note on the Impossibility of Obfuscation with Auxiliary Inputs. IACR Cryptology ePrint Archive, 2013:665, 2013.
20. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. SIAM Journal on computing, 1989, 18(16):186-208.
21. S.Goldwasser, S.Micali. Probabilistic Encryption. Journal of Computer and System Sciences, Vol.28, April 1984, pp. 270-299.
22. S. Goldwasser, and G.N. Rothblum. On best-possible obfuscation. In TCC, pp 194-213, 2007.
23. O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. J. of the ACM, 1991, 38(3):691-729.
24. D. Hofheinz, J.Malone-Lee, and M. Stam. Obfuscation for cryptographic purposes. In TCC, pp 2214-232, 2007.
25. A. Hohenberger, G. N. Rothblum. A.Shelat, and V. Vaikuntanathan. Securely obfuscating reencryption. In TCC, pp 233-252, 2007.
26. A. Hohenberger, A. Sahai, B. Waters. Replacing a random oracle: full domain hash from indistinguishability obfuscation. In EUROCRYPT, 2014.
27. M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In SODA 2001, pp 448-457,2001.
28. M. O. Rabin. How to Exchange Secrets by Oblivious Transfer. TR-81, Har- vard, 1981.
29. A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. IACR Cryptology ePrint Archive: 2013:454, 2013.