

A Black-Box Construction of Fully-Simulatable, Round-Optimal Oblivious Transfer from Strongly Uniform Key Agreement

Daniele Friolo^{1*}, Daniel Masny^{2†}, and Daniele Venturi^{1*}

¹ Department of Computer Science, Sapienza University of Rome, Rome, IT

² VISA Research, Palo Alto, CA

Abstract. We show how to construct maliciously secure oblivious transfer (M-OT) from a strengthening of key agreement (KA) which we call *strongly uniform* KA (SU-KA), where the latter roughly means that the messages sent by one party are computationally close to uniform, even if the other party is malicious. Our transformation is black-box, almost round preserving (adding only a constant overhead of up to two rounds), and achieves standard simulation-based security in the plain model.

As we show, 2-round SU-KA can be realized from cryptographic assumptions such as low-noise LPN, high-noise LWE, Subset Sum, DDH, CDH and RSA—all with polynomial hardness—thus yielding a black-box construction of fully-simulatable, round-optimal, M-OT from the same set of assumptions (some of which were not known before).

Keywords: Oblivious transfer · Malicious security · LPN

1 Introduction

Oblivious transfer (OT) is a very simple functionality between two parties: a sender with input two strings (s_0, s_1) , and a receiver with input a choice bit b ; the output for the receiver equals s_b , while the sender learns nothing (i.e., the receiver’s choice bit remains hidden) [51,15]. The standard security definition for OT compares an execution of the protocol in the real world—where either the sender or the receiver might act maliciously—with an execution in the ideal world where a trusted third party simply implements the above functionality. Following previous work, we call “*fully simulatable*” an OT protocol that meets this notion.

Surprisingly, OT turned out to be sufficient for constructing secure multiparty computation (MPC) for *arbitrary* functionalities [53,54,38,33,32,6,20]. For

*Supported in part by the research projects “PRIVacy-preserving, Security, and MACHine-learning techniques for healthcare applications (PRISMA)” and “Protect yourself and your data when using social networks”, both funded by Sapienza University of Rome, and in part by the MIUR under grant “Dipartimenti di eccellenza 2018-2022” of the Computer Science Department of Sapienza University of Rome.

†Part of the work was done at UC Berkeley, funded by the Center for Long-Term Cybersecurity (CLTC, UC Berkeley).

this reason, constructing OT has been an important objective and received much attention. Nevertheless, previous constructions of fully-simulatable OT suffer from diverse shortcomings (cf. also §1.4): (i) They require *trusted setup*, or are based on *random oracles* (as, e.g., in [34,50]); (ii) They have *high round complexity* (as, e.g., in [27]), while the optimal number of rounds would be 4 [32,19]; (iii) They are *non-black-box*, in that they are obtained by generically transforming semi-honestly secure OT (SH-OT)—which in turn can be constructed from special types of PKE [21]—to fully-simulatable OT via (possibly interactive) zero-knowledge proofs (*à la* GMW [22]); (iv) They are tailored to *specific hardness assumptions* (as, e.g., in [41,7]).

One exception is the work of Ostrovsky, Richelson and Scafuro [49], that provide a black-box construction of 4-round, fully-simulatable OT in the plain model from *certified trapdoor permutations* (TDPs) [5,45,10], which in turn can be instantiated from the RSA assumption under some parameter regimes [35,10]. This draws our focus to the question:

Can we obtain 4-round, fully-simulatable OT in a black-box way from minimal assumptions, without assuming trusted setup or relying on random oracles?

1.1 Our Contribution

We give a positive answer to the above question by leveraging a certain type of key agreement (KA) protocols, which intuitively allow two parties to establish a secure channel in the presence of an eavesdropper. The influential work by Impagliazzo and Rudich [31] showed a (black-box) separation between secret-key cryptography and public-key cryptography and KA. Ever since, it is common sense that public-key encryption (PKE) requires stronger assumptions than the existence of one-way functions, and thus secure KA is the weakest assumption from which public-key cryptography can be obtained. More recent research efforts have only provided further confidence in this conviction [18].

In more details, our main contribution is a construction of fully-simulatable OT (a.k.a. *maliciously secure* OT, or M-OT) from a strengthening of KA protocols, which we term *strongly uniform* (SU); our protocol is fully *black-box* and essentially *round-preserving*, adding only a constant overhead of at most two rounds. In particular, we show:

Theorem 1. *For any odd $t \in \mathbb{N}$, with $t > 1$, there is a black-box construction of a $(t + 1)$ -round, fully-simulatable oblivious transfer protocol in the plain model, from any t -round strongly uniform key agreement protocol and a perfectly binding commitment scheme.³*

Since, as we show, 2-round and 3-round SU-KA can be instantiated from several assumptions, including low-noise (ring) LPN, high-noise (ring) LWE, Subset

³Statistically binding commitment schemes are implied by perfectly-correct KA protocols [44]. Both LWE and low-noise LPN imply statistically binding commitment schemes as well [25].

Sum, CDH, DDH, and RSA—all with polynomial hardness—a consequence of our result is that we obtain round-optimal M-OT in the plain model under the same set of assumptions (in a black-box way). In particular, this yields the *first* such protocols from LPN, LWE (with modulus noise ratio \sqrt{n}), CDH, and Subset Sum.⁴ Note that our LWE parameter setting relates to an approximation factor of $n^{1.5}$ for SIVP in lattices of dimension n [52], which is the weakest LWE assumption known to imply PKE.

In our construction, we use a special kind of “*commit-and-open*” protocols which were implicitly used in previous works [39,49]. As a conceptual contribution, we formalize their security properties, which allows for a more modular presentation and security analysis.

1.2 Technical Overview

We proceed to a high level overview of the techniques behind our main result, starting with the notion of strong uniformity and the abstraction of commit-and-open protocols, and landing with the intuition behind our construction of M-OT (cf. Fig. 1).

Strong uniformity. As an important stepping stone to our main result, in §3, we introduce the notion of strong uniformity. Recall that a KA protocol allows Alice and Bob to share a key over a public channel, in such a way that the shared key is indistinguishable from uniform to the eyes of a passive eavesdropper. Strong uniformity here demands that, even if Bob is malicious, the messages sent by Alice are computationally close to uniform over an efficiently sampleable group.⁵ This flavor of security straightforwardly translates to SH-OT and PKE, yielding so-called SUSH-OT and SU-PKE. In the case of SUSH-OT, it demands that all messages of the receiver have this property (even if the sender is malicious). For SU-PKE, we distinguish two types, which are a strengthening of the types defined by Gertner *et al.* [21].⁶

- **Type-A PKE:** The distribution of the public key is computationally indistinguishable from uniform. This type of PKE is known to exist under

⁴We can also base our construction on Factoring when relying on the hardness of CDH over the group of signed quadratic residues [30], but this requires a trusted setup of this group which is based on a Blum integer.

⁵We call a group efficiently sampleable if we can efficiently sample uniform elements from the group and, given a group element, we can simulate this sampling procedure. A reverse sampleable group [23] would suffice. In the context of public-key encryption a similar property is called oblivious key generation [14]. In our construction, we require a stronger property where the public keys are additionally computationally indistinguishable from uniform.

⁶The difference is that the notions in [21] only ask for oblivious sampleability, rather than our stronger requirement of computational uniformity over efficiently sampleable groups.

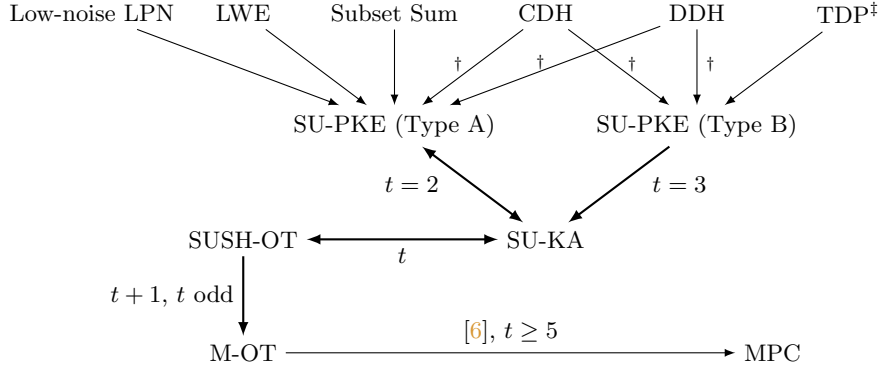


Fig. 1: Overview over equivalence and implications of the notion of strong uniformity. The value $t \in \mathbb{N}$ denotes the round complexity. \dagger This holds over efficiently sampleable groups. \ddagger We need an enhanced certified TDP.

DDH [17] and CDH [24] over efficiently sampleable groups,⁷ LWE [52], low-noise LPN [2], and Subset Sum [46].

- **Type-B PKE:** The encryption of a uniformly random message w.r.t. a maliciously chosen public key is computationally close to the uniform distribution over the ciphertext space. This type of PKE is harder to obtain, and can be constructed from enhanced certified TDPs, and from CDH and DDH over efficiently sampleable groups. In case of a TDP f , a ciphertext has the form $(f(r), h(r) \oplus m)$, where h is a hardcore predicate for f , and r is a random element from the domain of f . Under CDH or DDH, a ciphertext is defined as $(g^r, h(g^{xr}) \cdot m, g^{xr} \cdot m)$ respectively, where g^r is a uniform group element, and g^x is the public key. Clearly, for a uniform message m , these ciphertexts are uniform even under maliciously chosen public keys.

In §3, we show that SU Type-A and SU Type-B PKE imply, respectively, 2-round and 3-round SU-KA, whereas 2-round SU-KA implies SU Type-A PKE. Further, we prove that SU-KA is equivalent to SUSH-OT. The latter implies that strong uniformity is a sufficiently strong notion to bypass the black-box separation between OT and KA, in a similar way as Type-A and Type-B PKE bypass the impossibility of constructing OT from PKE [21].

Commit-and-open protocols. A 1-out-of-2 commit-and-open (C&O) protocol is a 3-round protocol with the following structure: (1) In the first round, the prover,

⁷These are groups for which one can directly sample a group element without knowing the discrete logarithm with respect to some generator. The latter requires non black-box access to the group, which is also needed when using ElGamal with messages that are encoded as group elements and not as exponents. Though we need the stronger property of sampleability of elements that are computationally close to uniform.

with inputs two messages m_0, m_1 and a bit d , sends a string γ (called “commitment”) generated with m_d but independent of m_{1-d} to the verifier; (2) In the second round, the verifier sends a value β to the prover (called “challenge”); (3) In the third round, the prover sends a tuple (δ, m_0, m_1) to the verifier (called “opening”). Security requires two properties. The first property, called *existence of a committing branch*, demands that a malicious prover must be committed to at least one message, i.e. m_d , already after having sent γ . The second property, called *committing branch indistinguishability*, asks that a malicious verifier cannot learn the committing branch, i.e. d , of an honest prover.

A construction of C&O protocols for single bits is implicit in Kilian [39]. This has been extended to strings by Ostrovsky *et al.* [49]. Both constructions make black-box use of a statistically binding commitment scheme, and allow a prover to equivocally open one of the messages.

M-OT from SUSH-OT: A warm up. In order to explain the main ideas behind our construction of M-OT, we describe below a simplified version of our protocol for the special case of $t = 2$, i.e. when starting with a 2-round SUSH-OT (S', R') ; here, we denote with ρ the message sent by the receiver, and with σ the message sent by the sender, and further observe that for the case of 2 rounds the notion of strong uniformity collapses to standard semi-honest security with the additional property that the distribution of ρ is (computationally close to) uniform to the eyes of an eavesdropper. We then construct a 4-round OT protocol (S, R) , as informally described below:

1. $(R \rightarrow S)$: The receiver picks a uniformly random value $m_{1-b} \in \mathcal{M}$, where b is the choice bit, and runs the prover of the C&O protocol upon input m_{1-b} , obtaining a commitment γ that is forwarded to the sender.
2. $(S \rightarrow R)$: The sender samples a challenge β for the C&O protocol, as well as uniformly random elements $r_0, r_1 \in \mathcal{M}$. Hence, it forwards (β, r_0, r_1) to the receiver.
3. $(R \rightarrow S)$: The receiver runs the receiver R' of the underlying 2-round OT protocol with choice bit fixed to 0, obtaining a value ρ_b which is used to define the message $m_b = \rho_b - r_b$ required to complete the execution of the C&O protocol in the non-committing branch b . This results in a tuple (δ, m_0, m_1) that is forwarded to the sender.
4. $(S \rightarrow R)$: The sender verifies that the transcript $T = (\gamma, \beta, (\delta, m_0, m_1))$ is accepting for the underlying C&O protocol. If so, it samples $u_0, u_1 \in \mathcal{M}$ uniformly at random, and runs the sender S' of the underlying 2-round OT protocol twice, with independent random tapes: The first run uses input strings (s_0, u_0) and message $m_0 + r_0$ from the receiver, resulting in a message σ_0 , whereas the second run uses input strings (s_1, u_1) and message $m_1 + r_1$ from the receiver, resulting in a message σ_1 . Hence, it sends (σ_0, σ_1) to the receiver.
5. Output: The receiver runs the receiver R' of the underlying 2-round OT protocol, upon input message s_b from the sender, thus obtaining s_b .

Correctness is immediate. In order to prove simulation-based security we proceed in two steps. In the first step, we show the above protocol achieves a weaker security flavor called *receiver-sided simulatability* [48,49] which consists of two properties: (1) The existence of a simulator which by interacting with the ideal OT functionality can fake the view of any efficient adversary corrupting the receiver in a real execution of the protocol (i.e., standard simulation-based security w.r.t. corrupted receivers); (2) Indistinguishability of the protocol transcripts with choice bit of the receiver equal to zero or one, for any efficient adversary corrupting the sender in a real execution of the protocol (i.e., game-based security w.r.t. corrupted senders). In the second step, we rely on a *round-preserving* black-box transformation given in [49], which allows to boost receiver-sided simulatability to fully-fledged malicious security. To show (1), we consider a series of hybrid experiments:

- In the first hybrid, we run the first 3 rounds of the protocol, yielding a partial transcript $\gamma, (\beta, r_0, r_1), (\delta, m_0, m_1)$. Hence, after verifying that $T = (\gamma, \beta, (\delta, m_0, m_1))$ is a valid transcript of the C&O protocol, we rewind the adversary to the end of the first round and continue the execution of the protocol from there using a fresh challenge (β', r'_0, r'_1) , except that after the third round we artificially abort if there is no value $\hat{b} \in \{0, 1\}$ such that $m_{\hat{b}} = m'_{\hat{b}}$, where (δ', m'_0, m'_1) is the third message sent by the adversary after the rewinding.

Notice that an abort means that it is not possible to identify a committing branch for the C&O protocol, which however can only happen with negligible probability; thus this hybrid is computationally close to the original experiment.

- In the second hybrid, we modify the distribution of the value r'_{1-b} (right after the rewinding) to $r''_{1-b} = \rho_{1-b} - m_{1-b}$, where we set $1 - b \stackrel{\text{def}}{=} \hat{b}$ from the previous hybrid, and where ρ_{1-b} is obtained by running the receiver R' of the underlying 2-round OT protocol with choice bit fixed to 1.

To argue indistinguishability, we exploit the fact that the distribution of m_{1-b} is independent from that of r'_{1-b} , and thus by strong uniformity we can switch $r'_{1-b} + m_{1-b}$ with ρ_{1-b} from the receiver R' .

- In the third hybrid, we use the simulator of the underlying 2-round SH-OT protocol to compute the messages σ_{1-b} sent by the sender. Note that in both the third and the second hybrid the messages $(\rho_{1-b}, \sigma_{1-b})$ are computed by the honest sender, and thus any efficient algorithm telling apart the third and the second hybrid violates semi-honest security of (S', R') .

In the last hybrid, a protocol transcript is independent of s_{1-b} but still yields a well distributed output for the malicious receiver, which immediately implies a simulator in the ideal world.

To show (2), we first use the strong uniformity property of (S', R') to sample m_b uniformly at random at the beginning of the protocol. Notice that this implies that the receiver cannot recover the value s_b of the sender anymore. Finally, we use the committing branch indistinguishability of the C&O protocol to argue that the transcripts with $b = 0$ and $b = 1$ are computationally indistinguishable.

M-OT from SUSH-OT: The general case. There are several difficulties when trying to extend the above protocol to the general case where we start with a t -round SUSH-OT. In fact, if we would simply iterate sequentially the above construction, where one iteration counts for a message from R' to S' and back, the adversary could use different committing branches from one iteration to the other. This creates a problem in the proof, as the simulator would need to be consistent with both choices of possible committing branches from the adversary, which however requires knowing both inputs from the sender.

We resolve this issue by having the receiver sending all commitments γ_i for the C&O protocol in the first round, where each value γ_i is generated including a random message m_{1-b}^i concatenated with the full history $m_{1-b}^{i-1}, \dots, m_{1-b}^1$. Hence, during each iteration, the receiver opens one commitment as before. As we show, this prevents the adversary from switching committing branch from one iteration to the next one. We refer the reader to §4 for a formal description of our protocol, and for a somewhat detailed proof intuition.

1.3 Application to Round-Efficient MPC

Since M-OT implies maliciously secure MPC [6,20] and very recently, the work of Choudhuri et al. [11], a direct consequence of Theorem 1 is the following:

Corollary 1. *For any odd $t \in \mathbb{N}$, there is a non-black-box construction of a $(t + 1)$ -round maliciously secure multi-party computation protocol in the plain model, from any t -round strongly uniform key agreement protocol.*

Corollary 1 yields 4-round maliciously secure MPC from any of low-noise LPN, high-noise LWE, Subset Sum, CDH, DDH, and RSA, all with polynomial hardness. Previously to our work, it was known how to get maliciously secure MPC in the plain model, for arbitrary functionalities:

- Using 5 rounds, via interactive ZK proofs and SH-OT [6], assuming polynomially-hard LWE with super-polynomial noise ratio and adaptive commitments [8], polynomially-hard DDH [3], and enhanced certified trapdoor permutations (TDP) [49,6];
- Using 4 rounds, assuming sub-exponentially-hard LWE with super-polynomial noise ratio and adaptive commitments [8], polynomially-hard LWE with a SIVP approximation factor of $n^{3.5}$ [7], sub-exponentially-hard DDH and one-way permutations [3], polynomially-hard DDH/QR/DCR [4], and either polynomially-hard QR or QR together with any of LWE/DDH/DCR (all with polynomial hardness) [29].

1.4 Related Work

Maliciously secure OT. Jarecki and Shamtikov [34], and Peikert, Vaikuntanathan, and Waters [50], show how to construct 2-round M-OT in the common reference string model.

A result by Haitner *et al.* [27,28] gives a black-box construction of M-OT from SH-OT. While being based on weaker assumptions (i.e., plain SH-OT instead of SUSH-OT), assuming the starting OT protocol has round complexity t , the final protocol requires 4 additional rounds for obtaining an intermediate security flavor known as “defensible privacy”, plus 4 rounds for cut and choose, plus 2 times the number of rounds required for running coin tossing, plus a final round to conclude the protocol. Assuming coin tossing can be done in 5 rounds [37], the total accounts to $t + 19$ rounds, and thus yields 21 rounds by setting $t = 2$.

Lindell [41] gives constructions of M-OT with 7 rounds, under the DDH assumption, the N th residuosity assumption, and the assumption that homomorphic PKE exists. Camenish, Neven, and shelat [9], and Green and Hohenberger [26], construct M-OT protocols, some of which even achieve adaptive security, using computational assumptions over bilinear groups.

There are also several efficient protocols for OT that guarantee only privacy (but not simulatability) in the presence of malicious adversaries, see, e.g. [40,47,1,36,7].

Round-optimal MPC. Katz and Ostrovsky [37] proved that 5 rounds are necessary and sufficient for realizing general-purpose two-party protocols, without assuming a simultaneous broadcast channel (where the parties are allowed to send each other messages in the same round). Their result was later extended by Garg *et al.* [19] who showed that, assuming simultaneous broadcast, 4 rounds are optimal for general-purpose MPC. Together with a result by Ishai *et al.* [32]—yielding *non-interactive* maliciously secure two-party computation for arbitrary functionalities, in the OT-hybrid model—the latter implies that 4 rounds are optimal for constructing fully-simulatable M-OT in the plain model.

Ciampi *et al.* [13] construct a special type of 4-round M-OT assuming certified TDPs,⁸ and show how to apply it in order to obtain (fully black-box) 4-round two-party computation with simultaneous broadcast. In a companion paper [12], the same authors further give a 4-round MPC protocol for the specific case of multi-party coin-tossing.

2 Preliminaries

2.1 Standard Notation

We use $\lambda \in \mathbb{N}$ to denote the security parameter, sans-serif letters (such as \mathbf{A} , \mathbf{B}) to denote algorithms, caligraphic letters (such as \mathcal{X} , \mathcal{Y}) to denote sets, and bold-face letters (such as \mathbf{v} , \mathbf{A}) to denote vectors and matrices; all vectors are by default row vectors, and \mathbf{v}^T denotes a column vector. An algorithm is *probabilistic polynomial-time* (PPT) if it is randomized, and its running time can be bounded by a polynomial in its input length. By $y \leftarrow_s \mathbf{A}(x)$, we mean that the value y is assigned to the output of algorithm \mathbf{A} upon input x and fresh random coins.

⁸They also claim [13, Footnote 3] that their OT protocol can be instantiated using PKE with special properties, however no proof of this fact is provided.

We implicitly assume that all algorithms are given the security parameter 1^λ as input.

A function $\nu : \mathbb{N} \rightarrow [0, 1]$ is negligible in the security parameter (or simply negligible) if it vanishes faster than the inverse of any polynomial in λ , i.e. $\nu(\lambda) \in O(1/p(\lambda))$ for all positive polynomials $p(\lambda)$. We often write $\nu(\lambda) \in \text{negl}(\lambda)$ to denote that $\nu(\lambda)$ is negligible.

For a random variable X , we write $\mathbb{P}[X = x]$ for the probability that X takes on a particular value $x \in \mathcal{X}$ (with \mathcal{X} being the set where X is defined). The statistical distance between two random variables X and X' defined over the same set \mathcal{X} is defined as $\Delta(X; X') = \frac{1}{2} \sum_{x \in \mathcal{X}} |\Pr[X = x] - \Pr[X' = x]|$. Given two ensembles $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$, we write $X \equiv Y$ to denote that they are identically distributed, $X \approx_s Y$ to denote that they are statistically close (i.e., $\Delta(X_\lambda; Y_\lambda) \in \text{negl}(\lambda)$), and $X \approx_c Y$ to denote that they are computationally indistinguishable—i.e., for all PPT distinguishers D there exists a negligible function $\nu : \mathbb{N} \rightarrow [0, 1]$ such that $|\Pr[D(X_\lambda) = 1] - \Pr[D(Y_\lambda) = 1]| \leq \nu(\lambda)$.

We call a group efficiently sampleable if and only if there is a PPT sampling procedure Samp for the uniform distribution over the group, and moreover there exists a PPT simulator SimSamp that given an element of the group, outputs the randomness used by Samp . More precisely, $(r, \text{Samp}(1^\lambda, r)) \approx_c (r', \text{Samp}(1^\lambda, r'))$ where $r' \leftarrow_s \text{SimSamp}(1^\lambda, \text{Samp}(1^\lambda, r))$ and $r \leftarrow_s \{0, 1\}^*$.⁹ A group that is efficiently reverse sampleable (as in [23]) suffices.

2.2 Oblivious Transfer

An interactive protocol Π for the Oblivious Transfer (OT) functionality, features two interactive PPT Turing machines S , R called, respectively, the sender and the receiver. The sender S holds a pair of strings $s_0, s_1 \in \{0, 1\}^\lambda$, whereas the receiver R is given a choice bit $b \in \{0, 1\}$. At the end of the protocol, which might take several rounds, the receiver learns s_b (and nothing more), whereas the sender learns nothing.

Typically, security of OT is defined using the real/ideal paradigm. Specifically, we compare a real execution of the protocol, where an adversary might corrupt either the sender or the receiver, with an ideal execution where the parties can interact with an ideal functionality. The ideal functionality, which we denote by \mathcal{F}_{OT} , features a trusted party that receives the inputs from both the sender and the receiver, and then sends to the receiver the sender's input corresponding to the receiver's choice bit. We refer the reader to Fig. 2 for a formal specification of the \mathcal{F}_{OT} functionality.

In what follows, we denote by $REAL_{\Pi, R^*(z)}(\lambda, s_0, s_1, b)$ (resp., $REAL_{\Pi, S^*(z)}(\lambda, s_0, s_1, b)$) the distribution of the output of the malicious receiver (resp., sender) during a real execution of the protocol Π (with s_0, s_1 as inputs of the sender, b as choice bit of the receiver, and z as auxiliary input for the adversary), and

⁹The existence of a simulator is crucial for constructing SUSH-OT from SU-KA; we solely use it for this purpose.

Ideal Functionality \mathcal{F}_{OT} :

The functionality runs with Turing machines (S, R) and adversary Sim, and works as follows:

- Upon receiving message (**send**, s_0, s_1, S, R) from S, where $s_0, s_1 \in \{0, 1\}^\lambda$, store s_0 and s_1 and answer **send** to R and Sim.
- Upon receiving a message (**receive**, b) from R, where $b \in \{0, 1\}$, send s_b to R and **receive** to S and Sim, and halt. If no message (**send**, \cdot) was previously sent, do nothing.

Fig. 2: Oblivious transfer ideal functionality

by $IDEAL_{\mathcal{F}_{\text{OT}}, \text{Sim}^{\text{R}^*}(z)}(\lambda, s_0, s_1, b)$ (resp., $IDEAL_{\mathcal{F}_{\text{OT}}, \text{Sim}^{\text{S}^*}(z)}(\lambda, s_0, s_1, b)$) the output of the malicious receiver (resp., sender) in an ideal execution where the parties (with analogous inputs) interact with \mathcal{F}_{OT} , and where the simulator is given black-box access to the adversary.

Definition 1 (OT with full simulation). *Let \mathcal{F}_{OT} be the functionality from Fig. 2. We say that a protocol $\Pi = (S, R)$ securely computes \mathcal{F}_{OT} with full simulation if the following holds:*

- (a) *For every non-uniform PPT malicious receiver R^* , there exists a non-uniform PPT simulator Sim such that*

$$\{REAL_{\Pi, \text{R}^*(z)}(\lambda, s_0, s_1, b)\}_{\lambda, s_0, s_1, b, z} \approx_c \{IDEAL_{\mathcal{F}_{\text{OT}}, \text{Sim}^{\text{R}^*}(z)}(\lambda, s_0, s_1, b)\}_{\lambda, s_0, s_1, b, z}$$

where $\lambda \in \mathbb{N}$, $s_0, s_1 \in \{0, 1\}^\lambda$, $b \in \{0, 1\}$, and $z \in \{0, 1\}^*$.

- (b) *For every non-uniform PPT malicious sender S^* , there exists a non-uniform PPT simulator Sim such that*

$$\{REAL_{\Pi, \text{S}^*(z)}(\lambda, s_0, s_1, b)\}_{\lambda, s_0, s_1, b, z} \approx_c \{IDEAL_{\mathcal{F}_{\text{OT}}, \text{Sim}^{\text{S}^*}(z)}(\lambda, s_0, s_1, b)\}_{\lambda, s_0, s_1, b, z}$$

where $\lambda \in \mathbb{N}$, $s_0, s_1 \in \{0, 1\}^\lambda$, $b \in \{0, 1\}$, and $z \in \{0, 1\}^*$.

Game-based security. One can also consider weaker security definitions for OT, where simulation-based security only holds when either the receiver or the sender is corrupted, whereas when the other party is malicious only game-based security is guaranteed. Below, we give the definition for the case of a corrupted sender, which yields a security notion known as *receiver-sided* simulatability. Intuitively, the latter means that the adversary cannot distinguish whether the honest receiver is playing with choice bit 0 or 1.

Definition 2 (OT with receiver-sided simulation). *Let \mathcal{F}_{OT} be the functionality from Fig. 2. We say that a protocol $\Pi = (S, R)$ securely computes \mathcal{F}_{OT} with receiver-sided simulation if the following holds:*

- (a) *Same as property (a) in Definition 1.*

(b) For every non-uniform PPT malicious sender S^* it holds that

$$\left\{ \text{VIEW}_{\Pi, S^*(z)}^R(\lambda, s_0, s_1, 0) \right\}_{\lambda, s_0, s_1, z} \approx_c \left\{ \text{VIEW}_{\Pi, S^*(z)}^R(\lambda, s_0, s_1, 1) \right\}_{\lambda, s_0, s_1, z}$$

where $\lambda \in \mathbb{N}$, $s_0, s_1 \in \{0, 1\}^\lambda$, and $z \in \{0, 1\}^*$, and where $\text{VIEW}_{\Pi, S^*(z)}^R(\lambda, s_0, s_1, b)$ is the distribution of the view of S^* (with input s_0, s_1 and auxiliary input z) at the end of a real execution of protocol Π with the honest receiver R given b as input.

Receiver-sided simulatability is a useful stepping stone towards achieving full simulatability. In fact, Ostrovsky *et al.* [49] show how to compile any 4-round OT protocol with receiver-sided simulatability to a 4-round OT protocol with full simulatability. This transformation can be easily extended to hold for any t -round protocol, with $t \geq 3$; the main reason is that the transform only relies on an extractable commitment scheme, which requires at least 3 rounds.

Theorem 2 (Adapted from [49]). *Assuming $t \geq 3$, there is a black-box transformation from t -round OT with receiver-sided simulation to t -round OT with full simulation.*¹⁰

2.3 Commit-and-Open Protocols

We envision a 3-round protocol between a prover and a verifier where the prover takes as input two messages $m_0, m_1 \in \mathcal{M}$ and a bit $d \in \{0, 1\}$. The prover speaks first, and the protocol is public coin, in the sense that the message of the verifier consists of uniformly random bits. Intuitively, we want that whenever the prover manages to convince the verifier, he must be committed to at least one of m_0, m_1 already after having sent the first message.

More formally, a 1-out-of-2 commit-and-open (C&O) protocol is a tuple of efficient interactive Turing machines $\Pi_{c\&o} \stackrel{\text{def}}{=} (\mathsf{P} = (\mathsf{P}_0, \mathsf{P}_1), \mathsf{V} = (\mathsf{V}_0, \mathsf{V}_1))$ specified as follows. (i) The randomized algorithm P_0 takes m_d and returns a string $\gamma \in \{0, 1\}^*$ and auxiliary state information $\alpha \in \{0, 1\}^*$; (ii) The randomized algorithm V_0 returns a random string $\beta \leftarrow_{\$} \mathcal{B}$; (iii) The randomized algorithm P_1 takes $(\alpha, \beta, \gamma, m_{1-d})$ and returns a string $\delta \in \{0, 1\}^*$; (iv) The deterministic algorithm V_1 takes a transcript $(\gamma, \beta, (\delta, m_0, m_1))$ and outputs a bit.

We write $\langle \mathsf{P}(m_0, m_1, d), \mathsf{V}(1^\lambda) \rangle$ for a run of the protocol upon inputs (m_0, m_1, d) to the prover, and we denote by $T \stackrel{\text{def}}{=} (\gamma, \beta, (\delta, m_0, m_1))$ the random variable corresponding to a transcript of the interaction. Note that the prover does not necessarily need to know m_{1-d} before computing the first message. We say that $\Pi_{c\&o}$ satisfies completeness if honestly generated transcripts are always accepted by the verifier, i.e. for all $m_0, m_1 \in \mathcal{M}$ and $d \in \{0, 1\}$, we have $\Pr[\mathsf{V}_1(T) = 1 : T \leftarrow_{\$} \langle \mathsf{P}(m_0, m_1, d), \mathsf{V}(1^\lambda) \rangle] = 1$, where the probability is over the randomness of $\mathsf{P}_0, \mathsf{V}_0$, and P_1 .

¹⁰They also need the existence of one-way functions. Since OT implies OT extension which implies one-way functions [42,43], OT implies one-way functions.

Security properties. Roughly, a C&O protocol must satisfy two security requirements. The first requirement is that at the end of the first round, a malicious prover is committed to at least one message. This can be formalized by looking at a mental experiment where we first run the protocol with a malicious prover, yielding a first transcript $T = (\gamma, \beta, (\delta, m_0, m_1))$; hence, we rewind the prover to the point it already sent the first message, and give it a fresh challenge β' which yields a second transcript $T' = (\gamma, \beta', (\delta', m'_0, m'_1))$. The security property now states that, as long as the two transcripts T and T' are valid, it shall exist at least one “committing branch” $\hat{d} \in \{0, 1\}$ for which $m_{\hat{d}} = m'_{\hat{d}}$. The second requirement says that no malicious verifier can learn any information on the committing branch of the prover. See the full version [16] for formal definitions.

3 Strong Uniformity at a Glance

This section contains a brief overview over the notion of strongly uniform OT and KA. We refer to the full version [16] for detailed definitions and for the implications of these notions.

In KA, Alice and Bob interact with the goal of establishing a shared key which remains hidden to an eavesdropper. We strengthen this notion by asking that Alice’s messages are computationally close to uniform over an efficiently sampleable group, even when Bob is malicious. We call this security feature *strong uniformity*.

Strong uniformity straightforwardly translates to OT. We call an OT protocol strongly uniform if the receiver’s messages are computationally close to uniform over an efficiently sampleable group, even when the sender is malicious. An important consequence of strong uniformity is that strongly uniform secure KA and strongly uniform semi-honestly secure OT are equivalent.

Theorem 3. *There is a black-box construction of strongly uniform semi-honestly secure OT from strongly uniform secure KA and vice versa, with the same round complexity.*

Intuitively, one can construct a KA protocol from OT by using the first of the sender’s inputs as key, and setting the receiver’s choice bit to 0, such that the receiver learns this key. Gertner *et al.* [21] already described this protocol, and it turns out that it preserves strong uniformity.

To construct strongly uniform semi-honestly secure OT from strongly uniform secure KA, one can use strong uniformity to let the receiver sample uniform messages rather than follow the KA protocol. More precisely, the sender and receiver will run two instances of the KA protocol, and the sender will use the two shared keys as one-time pad masks for his inputs. The receiver, depending on his choice bit, will run one of the two KA instances according to the protocol description, whereas, for the other one, he will sample uniform messages. Hence, the receiver will learn only one of the shared keys and inputs of the sender.

4 From SUSH-OT to M-OT

Let $\Pi_{\text{c\&o}} = (\mathsf{P}_0, \mathsf{P}_1, \mathsf{V}_0, \mathsf{V}_1)$ be a 1-out-of-2 C&O protocol and $\Pi' = (\mathsf{S}', \mathsf{R}')$ be a $(2t' + 1)$ -round OT protocol, where the first message σ^1 might be the empty string. Our OT protocol $\Pi = (\mathsf{S}, \mathsf{R})$ is depicted in Fig. 3 on page 19. The protocol consists of $(2t' + 2)$ rounds as informally described below.

1. The receiver samples $m_{1-b,i} \in \mathcal{M}$ for all $i \in [t']$, where b is the choice bit. Then he runs the prover of the C&O protocol upon input $(m_{1-b,j})_{j \in [i]}$ for all $i \in [t']$, obtaining $(\gamma_i)_{i \in [t']}$ which are forwarded to the sender.
2. The sender samples uniform values $u_0, u_1 \leftarrow^* \mathcal{M}$. Then, he runs the underlying $(2t' + 1)$ -round OT twice with inputs (s_0, u_0) and (s_1, u_1) to generate the first messages σ_0^1 and σ_1^1 . Further, the sender samples a challenge β_1 for the C&O protocol, as well as two uniformly random group elements $r_{0,1}, r_{1,1}$ from \mathcal{M} , and forwards $(\beta_1, r_{0,1}, r_{1,1})$ to the receiver together with the first messages of the OTs (i.e. σ_0^1 and σ_1^1).
3. Repeat the following steps for each $i \in [t']$:
 - (a) ($\mathsf{R} \rightarrow \mathsf{S}$): The receiver runs the receiver R' of the underlying $(2t' + 1)$ -round OT protocol with choice bit fixed to 0, and upon input message σ_b^i from the sender, obtaining a message ρ_b^i which is used to define the message $m_{b,i} = \rho_b^i - r_{b,i}$ required to complete the execution of the C&O protocol in the non-committing branch b . This results in a tuple $(\delta_i, m_{0,i}, m_{1,i})$ that is forwarded to the sender.
 - (b) ($\mathsf{S} \rightarrow \mathsf{R}$): The sender verifies that the transcript $T_i = (\gamma_i, \beta_i, (\delta_i, (m_{0,j})_{j \in [i]}, (m_{1,j})_{j \in [i]}))$ is accepting for the underlying C&O protocol. If so, he continues the two runs of the sender S' for the underlying $(2t' + 1)$ -round OT protocol. The first run uses state $\alpha_{\mathsf{S},0}^i$ and message $m_{0,i} + r_{0,i}$ from the receiver resulting in a message σ_0^{i+1} and state $\alpha_{\mathsf{S},0}^{i+1}$, whereas the second run uses state $\alpha_{\mathsf{S},1}^i$ and message $m_{1,i} + r_{1,i}$ from the receiver resulting in a message σ_1^{i+1} and state $\alpha_{\mathsf{S},1}^{i+1}$. Finally, the sender samples a challenge β_{i+1} for the C&O protocol, as well as another two uniformly random group elements $r_{0,i+1}, r_{1,i+1}$ from \mathcal{M} , and forwards $(\sigma_0^{i+1}, \sigma_1^{i+1})$ and $\beta_{i+1}, r_{0,i+1}, r_{1,i+1}$ to the receiver.
4. Output: The receiver runs the receiver R' of the underlying $(2t' + 1)$ -round OT protocol, upon input the $(t' + 1)$ -th message $\sigma_b^{t'+1}$ from the sender, thus obtaining an output $\rho_b^{t'+1}$.

Correctness follows by the fact that, when both the sender and the receiver are honest, by correctness of the C&O protocol the transcripts T_i are always accepting, and moreover the messages produced by the sender σ_b^i are computed using message $m_{b,i} + r_{b,i} = \rho_b^i$ from the receiver, so that each pair (ρ_b^i, σ_b^i) corresponds to the i -th interaction of the underlying $(2t' + 1)$ -round OT protocol with input strings (s_b, u_b) for the sender and choice bit 0 for the receiver, and thus at the end the receiver outputs s_b . As for security, we have:

Theorem 4 (Receiver-sided simulatability of Π). *Assuming that Π' is a $(2t' + 1)$ -round strongly uniform semi-honestly secure OT protocol, and that $\Pi_{\text{c\&o}}$*

is a secure 1-out-of-2 commit-and-open protocol, then the protocol Π from Fig. 3 securely realizes \mathcal{F}_{OT} with receiver-sided simulation.

We give a detailed proof in the full version [16], and here provide some intuition. In order to show receiver-sided simulatability we need to prove two things: (1) The existence of a simulator Sim which by interacting with the ideal functionality \mathcal{F}_{OT} can fake the view of any efficient adversary corrupting the receiver in a real execution of the protocol; (2) Indistinguishability of the protocol transcripts with choice bit of the receiver equal to zero or one, for any efficient adversary corrupting the sender in a real execution of the protocol.

To show (1), we consider a series of hybrid experiments that naturally lead to the definition of a simulator in the ideal world. In order to facilitate the description of the hybrids, it will be useful to think of the protocol as a sequence of t' iterations, where each iteration consists of 2 rounds, as depicted in Fig. 3 on page 19.

- In the first hybrid, we run a malicious receiver twice after he has sent his commitments. The purpose of the first run is to learn a malicious receiver's input bit, i.e. on which branch he is not committed. If he is committed on both branches, simulation will be easy since he will not be able to receive any of the sender's inputs. We use the second run to learn the output of a malicious receiver. We describe the two runs now.
 1. The first round of each iteration yields an opening $(\delta_i, m_{0,i}, m_{1,i})$. Hence, after verifying that the opening is valid, we rewind the adversary to the end of the first round of the i -th iteration to receive another opening $(\delta'_i, m'_{0,i}, m'_{1,i})$.
Now, let $b \in \{0, 1\}$ such that $m_{b,i} \neq m'_{b,i}$. By the security of the C&O protocol, there can be at most one such b . If there is no b we continue the first run. Otherwise, if there is such a b , we have learned the equivocal branch and start the second run.
 2. We execute the second run according to the protocol with the difference that we now know the equivocal branch, i.e. b , from the very beginning, which will help us later to simulate correctly right from the start. Notice that by the security of the C&O protocol, a malicious receiver cannot change the equivocal branch in the second run. Obviously, he cannot change it during the same iteration since then he would be equivocal on both branches and contradict the security of the C&O protocol. He can also not change the equivocal branch of one of the later rounds $j > i$, since in the j -th commitment δ_j he cannot be committed to both $m_{b,i}$ and $m'_{b,i}$, so he needs to equivocally open δ_j as well. Thus, he needs to be committed on the other branch, i.e. branch $1 - b$.
- The values $m'_{k,i}$ (right after the rewinding) of each iteration of the first run for $k \in \{0, 1\}$, and second run for $k = 1 - b$, are identical to $m_{k,i}$. Moreover, $m'_{k,i} \neq m_{k,i}$ holds only for the second run for branch $k = b$. Therefore, in the second hybrid, we can change the distribution of $r'_{k,i}$ to $r'_{k,i} = \rho_k^i - m_{k,i}$ for $k \in \{0, 1\}$, and both runs except branch $k = b$ during the

second run. The value ρ_k^i is obtained by running the simulator for the receiver of the underlying strongly uniform semi-honest OT protocol with choice bit 1 and input u_k . We can use the messages generated by this simulator on the sender's side as well.

We will use the strong uniformity of the OT to argue that a malicious receiver cannot distinguish $r'_{k,i} = \rho_k^i - m_{k,i}$ from uniform. By the semi-honest security, the messages generated by the simulator are indistinguishable from the actual semi-honest OT. At the same time this simulator is independent of the sender's inputs s_0 and s_1 . Note that in this hybrid, we only need to know s_b for the second run after having learned b .

In the last hybrid, a protocol transcript is independent of s_{1-b} but still yields a well distributed output for the malicious receiver, which directly yields a simulator in the ideal world.

To show (2), we first use the strong uniformity of the underlying OT protocol to sample $m_{b,i}$ uniformly at random at the beginning of the protocol. Notice that this implies that the receiver cannot recover the value s_b of the sender anymore. Further, we need the strong uniformity property here, since the receiver is interacting with a malicious sender who could influence the distribution of $m_{b,i}$ sent by the receiver. Once both messages, $m_{0,i}$ and $m_{1,i}$ for all iterations are known before the start of the protocol, we can challenge the choice bit indistinguishability of the C&O protocol. As a consequence, we can argue that the transcripts with $b = 0$ and $b = 1$ are computationally indistinguishable, which implies game-based security against a malicious sender.

5 Conclusions

We have shown a construction of maliciously secure oblivious transfer (M-OT) protocol from a certain class of key agreement (KA) and semi-honestly secure OT (SH-OT) protocols that enjoy a property called *strong uniformity* (SU), which informally means that the distribution of the messages sent by one of the parties is computationally close to uniform, even in case the other party is malicious.

When starting with 2-round or 3-round SUSH-OT or SU-KA, we obtain 4-round M-OT, and thus, invoking [11], 4-round maliciously secure MPC from standard assumptions including low-noise LPN, LWE, Subset Sum, CDH, DDH, and RSA (all with polynomial hardness).

Also, it is a natural question to see whether SU-KA with $t \geq 4$ rounds can be instantiated from concrete assumptions that do not imply PKE.

6 Acknowledgments

We would like to thank Silas Richelson for a discussion on their commit-and-open protocol. We also thank the anonymous reviewers who helped removing wrong claims and clarifying the presentation of our results.

References

1. Aiello, W., Ishai, Y., Reingold, O.: Priced oblivious transfer: How to sell digital goods. In: EUROCRYPT. pp. 119–135 (2001)
2. Alekhnovich, M.: More on average case vs approximation complexity. In: IEEE FOCS. pp. 298–307 (2003)
3. Ananth, P., Choudhuri, A.R., Jain, A.: A new approach to round-optimal secure multiparty computation. In: CRYPTO. pp. 468–499 (2017)
4. Badrinarayanan, S., Goyal, V., Jain, A., Kalai, Y.T., Khurana, D., Sahai, A.: Promise zero knowledge and its applications to round optimal MPC. In: CRYPTO. pp. 459–487 (2018)
5. Bellare, M., Yung, M.: Certifying cryptographic tools: The case of trapdoor permutations. In: CRYPTO. pp. 442–460 (1992)
6. Benhamouda, F., Lin, H.: k -round multiparty computation from k -round oblivious transfer via garbled interactive circuits. In: EUROCRYPT. pp. 500–532 (2018)
7. Brakerski, Z., Döttling, N.: Two-message statistically sender-private OT from LWE. In: TCC. pp. 370–390 (2018)
8. Brakerski, Z., Halevi, S., Polychroniadou, A.: Four round secure computation without setup. In: TCC. pp. 645–677 (2017)
9. Camenisch, J., Neven, G., Shelat, A.: Simulatable adaptive oblivious transfer. In: EUROCRYPT. pp. 573–590 (2007)
10. Canetti, R., Lichtenberg, A.: Certifying trapdoor permutations, revisited. In: TCC. pp. 476–506 (2018)
11. Choudhuri, A.R., Ciampi, M., Goyal, V., Jain, A., Ostrovsky, R.: Round optimal secure multiparty computation from minimal assumptions. Cryptology ePrint Archive, Report 2019/216 (2019), <https://eprint.iacr.org/2019/216>
12. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Delayed-input non-malleable zero knowledge and multi-party coin tossing in four rounds. In: TCC. pp. 711–742 (2017)
13. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Round-optimal secure two-party computation from trapdoor permutations. In: TCC. pp. 678–710 (2017)
14. Damgård, I., Nielsen, J.B.: Improved non-committing encryption schemes based on a general complexity assumption. In: CRYPTO. pp. 432–450 (2000)
15. Even, S., Goldreich, O., Lempel, A.: A randomized protocol for signing contracts. In: CRYPTO. pp. 205–210 (1982)
16. Friolo, D., Masny, D., Venturi, D.: A black-box construction of fully-simulatable, round-optimal oblivious transfer from strongly uniform key agreement. Cryptology ePrint Archive, Report 2018/473 (2018), <https://eprint.iacr.org/2018/473>
17. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: CRYPTO. pp. 10–18 (1984)
18. Garg, S., Mahmoody, M., Masny, D., Meckler, I.: On the round complexity of OT extension. In: CRYPTO. pp. 545–574 (2018)
19. Garg, S., Mukherjee, P., Pandey, O., Polychroniadou, A.: The exact round complexity of secure computation. In: EUROCRYPT. pp. 448–476 (2016)
20. Garg, S., Srinivasan, A.: Two-round multiparty secure computation from minimal assumptions. In: EUROCRYPT. pp. 468–499 (2018)
21. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: IEEE FOCS. pp. 325–335 (2000)

22. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM* **38**(3), 691–729 (1991)
23. Goldreich, O., Rothblum, R.D.: Enhancements of trapdoor permutations. *J. Cryptology* **26**(3), 484–512 (July 2013)
24. Goldwasser, S., Micali, S.: Probabilistic encryption. *J. Comput. Syst. Sci.* **28**(2), 270–299 (1984)
25. Goyal, R., Hohenberger, S., Koppula, V., Waters, B.: A generic approach to constructing and proving verifiable random functions. In: *TCC*. pp. 537–566 (2017)
26. Green, M., Hohenberger, S.: Blind identity-based encryption and simulatable oblivious transfer. In: *ASIACRYPT*. pp. 265–282 (2007)
27. Haitner, I.: Semi-honest to malicious oblivious transfer - the black-box way. In: *TCC*. pp. 412–426 (2008)
28. Haitner, I., Ishai, Y., Kushilevitz, E., Lindell, Y., Petrank, E.: Black-box constructions of protocols for secure computation. *SIAM J. Comput.* **40**(2), 225–266 (2011)
29. Halevi, S., Hazay, C., Polychroniadou, A., Venkatasubramanian, M.: Round-optimal secure multi-party computation. In: *CRYPTO*. pp. 488–520 (2018)
30. Hofheinz, D., Kiltz, E.: The group of signed quadratic residues and applications. In: *CRYPTO*. pp. 637–653 (2009)
31. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: *CRYPTO*. pp. 8–26 (1988)
32. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Prabhakaran, M., Sahai, A.: Efficient non-interactive secure computation. In: *EUROCRYPT*. pp. 406–425 (2011)
33. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer – Efficiently. In: *CRYPTO*. pp. 572–591 (2008)
34. Jarecki, S., Shmatikov, V.: Efficient two-party secure computation on committed inputs. In: *EUROCRYPT*. pp. 97–114 (2007)
35. Kakvi, S.A., Kiltz, E., May, A.: Certifying RSA. In: *ASIACRYPT*. pp. 404–414 (2012)
36. Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. In: *EUROCRYPT*. pp. 78–95 (2005)
37. Katz, J., Ostrovsky, R.: Round-optimal secure two-party computation. In: *CRYPTO*. pp. 335–354 (2004)
38. Kilian, J.: Founding cryptography on oblivious transfer. In: *ACM STOC*. pp. 20–31 (1988)
39. Kilian, J.: A note on efficient zero-knowledge proofs and arguments (extended abstract). In: *ACM STOC*. pp. 723–732 (1992)
40. Kushilevitz, E., Ostrovsky, R.: Replication is NOT needed: SINGLE database, computationally-private information retrieval. In: *IEEE FOCS*. pp. 364–373 (1997)
41. Lindell, Y.: Efficient fully-simulatable oblivious transfer. *Chicago J. Theor. Comput. Sci.* **2008** (2008)
42. Lindell, Y., Zarusim, H.: On the feasibility of extending oblivious transfer. In: *TCC*. pp. 519–538 (2013)
43. Lindell, Y., Zarusim, H.: On the feasibility of extending oblivious transfer. *J. Cryptology* **31**(3), 737–773 (2018)
44. Lombardi, A., Schaeffer, L.: A note on key agreement and non-interactive commitments. *Cryptology ePrint Archive, Report 2019/279* (2019), <https://eprint.iacr.org/2019/279>
45. Lysyanskaya, A., Micali, S., Reyzin, L., Shacham, H.: Sequential aggregate signatures from trapdoor permutations. In: *EUROCRYPT*. pp. 74–90 (2004)

46. Lyubashevsky, V., Palacio, A., Segev, G.: Public-key cryptographic primitives provably as secure as subset sum. In: TCC. pp. 382–400 (2010)
47. Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: SODA. pp. 448–457 (2001)
48. Naor, M., Pinkas, B.: Computationally secure oblivious transfer. *J. Cryptology* **18**(1), 1–35 (2005)
49. Ostrovsky, R., Richelson, S., Scafuro, A.: Round-optimal black-box two-party computation. In: CRYPTO. pp. 339–358 (2015)
50. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: CRYPTO. pp. 554–571 (2008)
51. Rabin, M.O.: How to exchange secrets by oblivious transfer. Tech. rep., Harvard University (1981)
52. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: ACM STOC. pp. 84–93 (2005)
53. Yao, A.C.: Protocols for secure computations (extended abstract). In: IEEE FOCS. pp. 160–164 (1982)
54. Yao, A.C.: How to generate and exchange secrets (extended abstract). In: IEEE FOCS. pp. 162–167 (1986)

Sender $S(s_0, s_1)$	Receiver $R(b)$
$w_0, w_1 \leftarrow \mathcal{M}$	$\alpha_{R,b}^0 = 0$
$\alpha_{S,0}^0 = (s_0, u_0)$	$\forall i \in [t'] :$
$\alpha_{S,1}^0 = (s_1, u_1)$	$m_{1-b,i} \leftarrow \mathcal{M}$
$(\alpha_{S,0}^1, \sigma_0^1) \leftarrow \mathcal{S}'(\alpha_{S,0}^0)$	$(\gamma_i, \alpha_i) \leftarrow \mathcal{P}_0((m_{1-b,j})_{j \in [i]})$
$(\alpha_{S,1}^1, \sigma_1^1) \leftarrow \mathcal{S}'(\alpha_{S,1}^0)$	$\xleftarrow{(\gamma_i)_{i \in [t'()]}}$
$\beta_1 \leftarrow \mathcal{V}_0(1^\lambda)$	$\xleftarrow{(\beta_1, (r_{k,1}, \sigma_k^1)_{k \in \{0,1\}})}$
$r_{0,1}, r_{1,1} \leftarrow \mathcal{M}$	
..... Repeat for each $i \in [t']$	
	$(\alpha_{R,b}^i, \rho_b^i) \leftarrow \mathcal{R}'(\alpha_{R,b}^{i-1}, \sigma_b^i)$
	$m_{b,i} = \rho_b^i - r_{b,i}$
	$\delta_i \leftarrow \mathcal{P}_1(\alpha_i, \beta_i, \gamma_i, (m_{b,j})_{j \in [i]})$
return \perp	
$(\alpha_{S,0}^{i+1}, \sigma_0^{i+1}) \leftarrow \mathcal{S}'(\alpha_{S,0}^i, m_{0,i} + r_{0,i})$	
$(\alpha_{S,1}^{i+1}, \sigma_1^{i+1}) \leftarrow \mathcal{S}'(\alpha_{S,1}^i, m_{1,i} + r_{1,i})$	
$\beta_{i+1} \leftarrow \mathcal{V}_0(1^\lambda)$	
$r_{0,i+1}, r_{1,i+1} \leftarrow \mathcal{M}$	$\xleftarrow{(\beta_{i+1}, (r_{k,i+1}, \sigma_k^{i+1})_{k \in \{0,1\}})}$
.....	$(\alpha_{R,b}^{t'+1}, \rho_b^{t'+1}) \leftarrow \mathcal{R}'(\alpha_{R,b}^{t'}, \sigma_b^{t'+1})$
	output $s_b = \rho_b^{t'+1}$

Fig. 3: $(2t' + 2)$ -round OT protocol achieving receiver-sided simulatability from $(2t' + 1)$ -round strongly uniform semi-honestly secure OT. Note that the initial state information $\alpha_{S,0}^0, \alpha_{S,1}^0$ and $\alpha_{R,b}^0$ is set to be equal, respectively to the inputs used by the sender and the receiver during the runs of the underlying OT protocol ($\mathcal{S}', \mathcal{R}'$). The values $\beta_{t'+1}, r_{0,t'+1}, r_{1,t'+1}$ are not needed and can be removed, but we avoided to do that in order to keep the protocol description more compact.