Two-Message, Oblivious Evaluation of Cryptographic Functionalities

Nico Döttling[1], Nils Fleischhacker[2(✉)], Johannes Krupp[2], and Dominique Schröder[2,3]

[1] University of California, Berkeley, USA
[2] CISPA, Saarland University, Saarbrücken, Germany
fleischhacker@cs.uni-saarland.de
[3] Friedrich-Alexander-University, Nuremberg, Germany

Abstract. We study the problem of two round oblivious evaluation of cryptographic functionalities. In this setting, one party P_1 holds a private key sk for a provably secure instance of a cryptographic functionality $\mathcal{F}$ and the second party P_2 wishes to evaluate $\mathcal{F}_{sk}$ on a value x. Although it has been known for 22 years that *general* functionalities cannot be computed securely in the presence of malicious adversaries with only two rounds of communication, we show the existence of a round optimal protocol that obliviously evaluates *cryptographic* functionalities. Our protocol is provably secure against malicious receivers under standard assumptions and does not rely on heuristic (setup) assumptions. Our main technical contribution is a novel *non*black-box technique, which makes *non*black-box use of the security reduction of $\mathcal{F}_{sk}$. Specifically, our proof of malicious receiver security uses *the code* of the reduction, which reduces the security of $\mathcal{F}_{sk}$ to some hard problem, in order to break that problem directly. Instantiating our framework, we obtain the first two-round oblivious pseudorandom function that is secure in the standard model. This question was left open since the invention of OPRFs in 1997.

1 Introduction

An oblivious evaluation protocol of a cryptographic functionality $\mathcal{F}$, is a two-party protocol in which one party P_1, called the sender, holds a function $\mathcal{F}_{sk} \in \mathcal{F}$ and the second party P_2, called the receiver, wishes to evaluate $\mathcal{F}_{sk}$ on x. Sender security says that P_1 remains oblivious of x while receiver security guarantees that the security of $\mathcal{F}_{sk}$ is preserved, i.e., evaluating $\mathcal{F}_{sk}$ obliviously should be as secure as having direct access to $\mathcal{F}$, even if a malicious party deviates from the protocol arbitrarily. Although it has been known for 22 years that *general* functionalities cannot be computed securely in the presence of malicious adversaries with only two rounds (messages) of communication [29], we show the existence of a two message protocol that obliviously evaluates *cryptographic* functionalities. The functionalities covered by our framework have the following properties:

- There is a security experiment Exp that characterizes the security of $\mathcal{F}$.
- The experiment Exp gives the adversary access to an oracle $\mathcal{O}$.

M. Robshaw and J. Katz (Eds.): CRYPTO 2016, Part III, LNCS 9816, pp. 619–648, 2016.
DOI: 10.1007/978-3-662-53015-3_22

– There is a black-box reduction $\mathcal{B}$ with certain properties that reduces the security of $\mathcal{F}_{sk}$ to a hard problem π.

Our framework subsumes popular two-party protocols, such as blind signatures and oblivious pseudorandom functions (OPRF). In fact, our framework yields the first OPRF with only two rounds of communication in the standard model — a problem that has been open since their invention in 1997 [49].

Technical Contribution. Our main technical contribution is a *non*black-box proof technique, which is *non*black-box *in the reduction*. To explain what being *non*black-box means, consider an instance P of a cryptographic functionality $\mathcal{F}$. Assume further that this instance is provably secure, i.e., there is a reduction $\mathcal{B}$ that turns any adversary $\mathcal{A}$ breaking the security of P into an algorithm solving the underlying hard problem π. Our protocol then shows that P can be evaluated securely. The corresponding proof of malicious receiver security makes *non*black-box use of the underlying *code* of the reduction $\mathcal{B}$. This proof does *not* reduce the security to P but to the underlying hard problem exploiting the code of $\mathcal{B}$. To best of our knowledge, this is the first result that shows how to make *non*black-box use of the code of a given security reduction. We call this class of reductions *oblivious reductions*.

1.1 Impossibility of Malicious Security and Induced Game-Based Security

Ideally one would like to achieve the standard notion of simulation based security against malicious adversaries. This notion says that the malicious receiver and sender learn only $f(x)$ (except what can trivially be learned from $f(x)$) and that the private input of the other party remains hidden. Unfortunately, it is well known that standard simulation based security notions cannot be achieved for two-round secure function evaluation (SFE) [29]. In fact, if one uses black-box techniques only, then at least five rounds of communication are necessary [36].

Since there is no hope in achieving malicious simulation-based security, we propose an alternative definition of malicious security for the setting of secure evaluation of cryptographic primitives: On a high-level, our security notions for malicious receiver says that the security properties of the underlying cryptographic primitive is preserved even against malicious adversaries. More precisely, we consider the secure evaluation of cryptographic functionalities, which are equipped with a game-based security notion. In our formalization the adversary in the corresponding security experiment has black-box access to the primitive. Then, we define an induced security notion by replacing black-box calls to the primitive in the security game with instances of the two-round SFE protocol. I.e., instead of giving the adversary black-box access to the primitive, it acts as a malicious receiver in an SFE session with the sender. Achieving this notion and showing that the underlying security guarantees are preserved is non-trivial, because the adversary is *not* semi-honest and may not follow the protocol.

Regarding security against corrupted senders, we show that malicious sender security *and* induced game-based security against malicious receivers cannot be achieved under (standard) non-interactive assumptions. In fact, our result is more general as it rules out protocols with three moves. Our impossibility is constructive and shows that our notion captures the standard definition of blind signatures. But for blind signatures it is well known that a large class of three-move blind signture schemes cannot be proven secure in the standard model under non-interactive assumptions [16]. Since our blind signature scheme belongs to this class, it follows that achieving both notions of malicious security is impossible. Thus, we also need to weakening the security against malicious senders and we stick to the standard notion of semi-honest security.

1.2 Oblivious Reductions: A Nonblack-Box Proof Technique

We give a high-level overview of our protocol and proof strategy. Our starting point is an instance $\mathcal{F}_{sk}$ of some cryptographic functionality $\mathcal{F}$ (such as the pseudorandom function functionality). The corresponding security proof is a *black-box* reduction $\mathcal{B}$ to some underlying hard problem π. Our goal is to obliviously compute $\mathcal{F}_{sk}$ in a secure two-party protocol Π with only two rounds of communication. Our protocol is simple and uses a certain type of homomorphic encryption and works as follows: The receiver encrypts its input x using the homomorphic encryption scheme, it sends the ciphertext $c \leftarrow \mathsf{Enc}(x)$ to the sender. The sender evaluates the function $\mathcal{F}_{sk}$ on c computing $c' \leftarrow \mathsf{Eval}(c, \mathcal{F}_{sk})$ and returns c' to the receiver, who obtains $\mathcal{F}_{sk}(x)$ by simply decrypting c'. Using fully homomorphic encryption as the underlying encryption scheme, it is well known that this protocol is secure against *semi-honest* adversaries [23].

However, we are interested in achieving *malicious* security and we achieve our goal using a specific type of homomorphic encryption scheme in combination with our novel *non*black-box proof technique. We provide an efficient reduction from the security of the homomorphically evaluated primitive to the underlying problem π directly using the code of the reduction $\mathcal{B}$. Our proof technique works for a large and natural class of black-box reductions that we call *oblivious*. Loosely speaking, a reduction is oblivious, if it only knows an upper bound on the number of the oracle queries, but does neither learn the query nor the answer. We give several examples of known oblivious reductions in Sect. 2.2 and we sketch the basic ideas of this technique in the following.

In the first step of our proof (see Fig. 1), we run a security experiment where the malicious receiver $\mathcal{A}$ has oracle access to a homomorphically evaluated functionality $\mathsf{Eval}(c, \mathcal{F}_{sk})$. In the second step, the experiment is transformed in the following way. First, the adversary's oracle inputs are extracted via an unbounded extractor, the functionality is evaluated on the extracted input, and finally the output is encrypted (with the right distribution). Assuming that the homomorphic encryption is (statistically) circuit private, we show that this modification does not change the success probability of the adversary. While extracting an input x from a ciphertext c is not possible in polynomial-time, it does not change the success probability of $\mathcal{A}$.

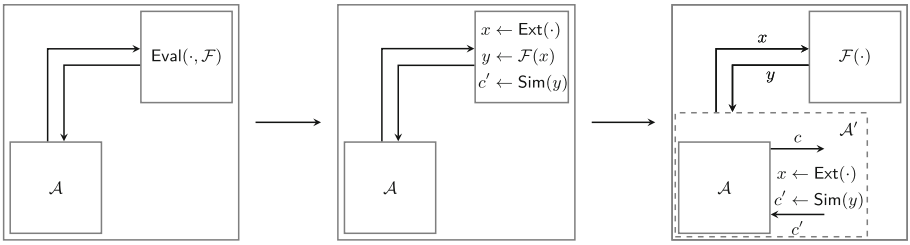

Fig. 1. Oblivious reduction part 1 of 2.

In the third step (see right picture of Fig. 1), we move the extraction and simulation procedures from the security experiment into the adversary itself, obtaining an unbounded adversary $\mathcal{A}'$. That is, the modified attacker $\mathcal{A}'$ runs $\mathcal{A}$ as a black-box. Whenever $\mathcal{A}$ sends c to its oracle, then $\mathcal{A}'$ extract x from c, invokes its own oracle obtaining $y \leftarrow F(x)$, and returns the encryption of y to $\mathcal{A}$. Obviously, the adversary $\mathcal{A}'$ does not run in polynomial-time, but this does not change its success probability, as we have only re-arranged the algorithms from one machine into another, but the overall composed algorithm remained the same.

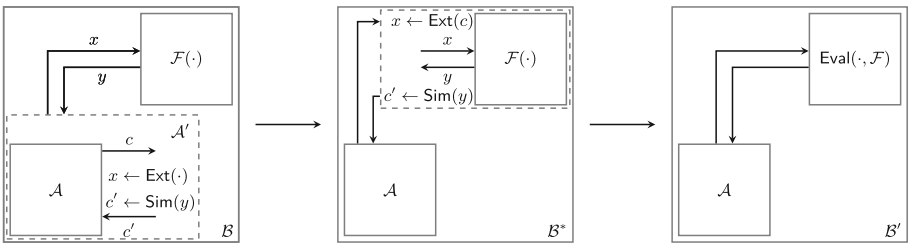

Fig. 2. Oblivious reduction part 2 of 2.

Consider the three steps shown in Fig. 2. In the first part, the unbounded adversary is plugged into the oblivious black-box reduction $\mathcal{B}$, which reduces the security of $\mathcal{F}$ to some hard problem π. This step is legitimate because the reduction only makes black-box use of the adversary. Observe that a black-box reduction cannot tell the difference between a polynomial-time adversary and an unbounded adversary, but only depends on the adversary's advantage in the security experiment. Thus, $\mathcal{B}^{\mathcal{A}'}$ is an inefficient adversary against the problem π. In our next modification we move the extraction and simulation algorithms from the adversary $\mathcal{A}'$ into the oracle-circuit. While this is just a bridging step, the inefficient algorithms for extraction and simulation are now part of the reduction. That is, whenever $\mathcal{A}$ queries c to its oracle, then the reduction $\mathcal{B}^*$ first extracts x from c and runs the simulation of $\mathcal{B}$ afterwards in order to compute the simulated answer $y \leftarrow \mathcal{F}_{sk}(x)$. Subsequently, $\mathcal{B}^*$ encrypts y as c' and sends this answer to $\mathcal{A}$. As a result, we obtain an inefficient reduction $\mathcal{B}^*$ that uses the code of the underlying reduction.

In the last step of our proof, we turn $\mathcal{B}^*$ into an *efficient* reduction $\mathcal{B}'$ against the underlying hard problem π (last picture in Fig. 2). Here, we again exploit the statistical circuit privacy of the homomorphic encryption scheme and replace the inefficient computation by the homomorphic evaluation of $\mathcal{F}$.

Running-Time of the Reduction. One may have the impression that we cheated in our proof by building a reduction that is not efficiently computable. This is not the case. A closer look at the formal proof reveals that the computationally inefficient steps are happening "inside" of the parts where we exploit the statistical circuit privacy. Thus, in some sense one may view this step as a game "in the head" while running an efficient reduction.

1.3 Our Contribution

The main contributions of this work are the following:

- We put forward the study of two-message secure evaluation of cryptographic functionalities.
- We propose a novel security model which says that the underlying security properties of the cryptographic functionality must be preserved, even if the malicious receiver does not follow the protocol.
- We show that security against malicious receivers with respect to our notion of induced game-based security and malicious senders cannot be achieved simultaneously in the standard model. In fact, our impossibility result is more general as it covers protocols with three moves.
- We suggest a protocol that is provably secure in this model under standard assumptions. The corresponding security proof relies on a novel *non*black-box technique that is *nonblack*-box in the reduction. We believe that this technique might be of independent interest.
- As an instance of our protocol, we present the first two-move oblivious pseudorandom function and solve a problem that was open since their invention in 1997.

1.4 Related Work

In this section, we discuss related works in the areas of secure two-party computation, round optimal oblivious PRFs and blind signatures.

Secure Two-Party Computation. The seminal works of Yao [58] and Goldreich et al. [28] show that any polynomial-time function can be securely computed in various settings. Recent works have shown protocols for secure two- and multi-party computation with practical complexity such as [7,13,44,51]. A central measure of efficiency for interactive protocols is the round complexity. It was shown that secure two-party computation of arbitrary functionalities cannot be realized with only two rounds [29,42,43], and if the security proof uses

black-box techniques only, then 5 rounds are needed [36]. On the other hand, several meaningful functionalities can be realized with only two (resp. less than five) rounds. Research in this area has gained much attention in the past and upper and lower bounds for many cryptographic protocols were discovered, such as for (concurrent) zero-knowledge proofs and arguments [5,15,26,27,29,56] and [10,39,54], blind signatures [16,19,20], as well as two- and multi-party computation [3,4,21,32,41,58] and [12,22,37].

Round Optimal Oblivious PRFs. Oblivious pseudorandom functions are in essence pseudorandom functions (PRFs) that are obliviously evaluated in a two-party protocol. This means that the sender $\mathcal{S}$ holds a key k of a PRF F and the receiver $\mathcal{R}$ a value x and wishes to learn $F(k, x)$. OPRFs have many applications, such as private key-word search [17], or secure computation of set intersection [34]. However, besides the popularity of this primitive, no scheme in the standard model is known with only two-rounds of communication. The first OPRF scheme was proposed by Naor and Reingold and it requires $\mathcal{O}(\lambda)$ rounds [49]. Freedman et al. [17] used previous work of Naor and Pinkas [46,47] to extend this to a constant round protocol assuming the hardness of DDH. Note that this protocol realizes a "weak PRF", which allows the receiver to learn information about the key k as long as this information does not change the pseudorandomness of future queries. Jarecki and Liu suggested the first round optimal OPRFs in the random oracle model [34].

Round Optimal Blind Signatures. A blind signature scheme [11] allows a signer to interactively issue signatures for a user such that the signer learns nothing about the message being signed (blindness) while the user cannot compute any additional signature without the help of the signer (unforgeability). Constructing round-optimal blind signature schemes in the standard model has been a long standing open question. Fischlin and Schröder showed that all previously known schemes having at most three rounds of communication, cannot be proven secure under non-interactive assumptions in the standard model via black-box reductions [16]. Subsequently, several works used a technique called "complexity leveraging" to circumvent this impossibility result [19,20] and recently, Fuchsbauer, Hanser, and Slamanig suggested a round optimal blind signature scheme that is secure in the generic group model [18]. In fact, it is still unknown if round optimal blind signatures, based on standard assumptions, exist in the standard model.

1.5 Outlook

Our work also shows that the "quality" of the proof has implication on the usability of the primitive in other contexts. In particular, having an oblivious black-box reduction, in contrast to a non-oblivious one, implies that the primitive can be securely evaluated in our framework while the underlying security is preserved. In fact, our results show a certain degree of composability of cryptographic functionalities and round optimal secure function evaluation.

Outline. We define our security model in Sect. 2. Our protocol is then given in Sect. 3. Section 4 shows how our result can be applied to achieve oblivious pseudorandom functions. The impossibility result is given in Sect. 4.

Notations. The security parameter is λ. By $y \leftarrow A(x;r)$ we refer to a (PPT) algorithm A that gets as input some value x and some randomness r and returns an output y. If X is a set, then $x \xleftarrow{\$} X$ means that x is chosen uniformly at random from X. The statistical distance $\Delta(A,B)$ of two probability distributions A and B is defined as $\Delta(A,B) = \frac{1}{2}\sum_v |Pr(A=v) - Pr(B=v)|$.

2 Secure Computation of Cryptographic Functionalities

In the following section, we formalize experiments, the corresponding notion of security of an experiment, oblivious black-box reduction, and our notion of secure computation of cryptographic primitives. Our formalization of experiments is similar to the one by Bellare and Rogaway [6], but our goal is to formalize oblivious reduction, i.e., reduction that only knows an upper number on the number of oracle queries made by an adversary and which does not see the actual queries to the oracle.

Please note that in the literature the term "round" has been used both to refer to a single message (either from A to B *or* from B to A) and to refer to two messages (one from A to B *and* one from B to A). Since none of the two seems to be favoured over the other, in this work we will stick to the former usage, i.e., a "round" refers a *single* message despite its direction.

2.1 Cryptographic Security Experiment

In this section, we formalize security experiments for cryptographic primitives P, where we view P as a collection of efficient algorithms. The basic idea of our notion is to define a framework, similar to the one of Bellare and Rogaway [6], for cryptographic experiments. Our framework provides some basic algorithm, such as initialization, an update mechanism, and a method to test if the adversary succeeds in the experiment. Moreover, it also define oracles that may be queried by the attacker. The most important aspect of our formalization is that the experiment is oblivious about the adversary's queries to its oracle. This means that the experiment may know an upper bound on the total number of queries, but does not learn the queries, or the corresponding answers.

Formally, the experiment consists of four algorithm. The first algorithm, Init, initializes the environment of the security experiment and computes publicly available informations pp and private informations st that may be hardcoded into the oracle that will be used by the attacker in the corresponding security notion. The algorithm Init receives a upper bound q on number of oracle queries as input. This is necessary because several security experiments, such as the one of blind signatures, require a concrete bound on the number of queries. This

oracle, denoted by OA, obtains $(\mathsf{pp}, \mathsf{st})$ and some query x, and it either returns an answer y, or $\perp$ to indicate failure. The update algorithm Update allows to re-program the oracle. The test algorithm Test checks the validity of some value out with respect to public and private informations pp and st, respectively.

Definition 1 (Security Experiment). *A* security experiment *for a cryptographic primitive* P *is a tuple of four algorithms defined as follows:*

Initialization. *The initialization algorithm* $\mathsf{Init}(1^\lambda, q)$ *gets as input the security parameter* 1^λ *and an upper bound* q *on the number queries. It outputs some public information* pp *together with some private information* st.

Oracle. *The oracle algorithm* $\mathsf{OA}(\mathsf{pp}, \mathsf{st}, x)$ *gets as input a string* pp*, state information* st*, and a query* x*. It answers with special symbol* $\perp$ *if the query is invalid, and otherwise with a value* y.

Update. *The stateful algorithm* $\mathsf{Update}(\mathsf{st}, \mathsf{resp})$ *takes as input some state information* st *and a string* resp*. It outputs some updated information* st.

Testing. *The* $\mathsf{Test}(\mathsf{pp}, \mathsf{st}, \mathsf{out})$ *algorithm gets as input the input of the attacker* pp*, state information* st*, the output of the attacker* out*, and outputs a bit* b *signifying whether the attacker was successful.*

In almost all cases, the oracle OA embeds an algorithm from the primitive P, such as the signing algorithm in case of signature, or the encryption algorithm in case of the CPA (resp. CCA) security game. Given the formalization of a security experiment, we are ready to formalize the corresponding notion of security. Loosely speaking, a cryptographic primitive is secure, if the success probability of the adversary in this experiment is only negligible bigger than the guessing probability. Since our notions covers both computational and decisional cryptographic experiments, we follow the standard way of introducing a function ν that serves as a security threshold and which corresponds to the guessing probability. In our formalization, the adversary $\mathcal{A}$ is a stateful algorithm that runs r rounds of the security experiment. This algorithm is initially intitialized with an empty state $\mathsf{st}_\mathcal{A} := \emptyset$. Our formalization could also handle non-uniform adversaries by setting this initial state to some string.

Definition 2 (Security of a Cryptographic Primitive). *Let* $\mathsf{Exp} = (\mathsf{Init}, \mathcal{O}, \mathsf{Update}, \mathsf{Test})$ *be a security experiment for a cryptographic primitive* P*, and let* $\mathcal{A}$ *be an adversary having a state* $\mathsf{st}_\mathcal{A}$ *querying the oracle exactly once per invocation. Further let* $\nu : \mathbb{N} \to [0, 1]$ *be a function. In abuse of notation, we denote by* $\mathsf{Exp}^\mathsf{P}(\mathcal{A})$ *the following cryptographic security experiment:*

Game $\mathsf{Exp}^\mathsf{P}(\mathcal{A})$
- $(\mathsf{pp}, \mathsf{st}) \leftarrow \mathsf{Init}(1^\lambda, q)$
- $\mathsf{st}_\mathcal{A} := \emptyset$
- *for* $i = 1$ *to* q *do*
  - $(\mathsf{resp}_i, \mathsf{st}_\mathcal{A}) \leftarrow \mathcal{A}^{\mathcal{O}(\mathsf{pp}, \mathsf{st}, \cdot)}(\mathsf{pp}, \mathsf{st}_\mathcal{A})$
  - $(\mathsf{pp}, \mathsf{st}) \leftarrow \mathsf{Update}(\mathsf{st}, \mathsf{resp}_i)$
- $\mathsf{out} := \mathsf{resp}_q$
- $b \leftarrow \mathsf{Test}(\mathsf{pp}, \mathsf{st}, \mathsf{out})$
- *Return* b

Oracle $\mathcal{O}(\mathsf{pp}, \mathsf{st}, x)$
- $y \leftarrow \mathsf{OA}(\mathsf{pp}, \mathsf{st}, x)$
- *Return* y

We define the advantage of the adversary $\mathcal{A}$ as

$$\mathbf{Adv}^{P}(\mathcal{A}) := \left|\mathrm{Prob}\left[\mathsf{Exp}^{\mathsf{P}}(\mathcal{A}) = 1\right] - \nu(\lambda)\right|.$$

A cryptographic primitive is secure *with respect to $\mathsf{Exp}^{\mathsf{P}}(\mathcal{A})$ if the advantage $\mathbf{Adv}^{P}(\mathcal{A})$ is negligible (in λ).*

Remark 1. Observe that in our formalization of a cryptographic security experiment, all algorithms, except for the adversary, are oblivious of the queries to the oracle. The reason is that the output of the oracle is returned to the adversary only and no other algorithm obtains this value. In particular, the update algorithm does not receive the output as an input and also the test algorithm, which determines if the attacker is successful, only receives pp, st, and out as an input and no input or output from OA.

The CCA Secure Encryption Experiment. Our formalization of cryptographic experiments covers standard security notions, such as CCA security for public-key encryption schemes (obviously, the adaption to CCA secure private-key encryption is trivial). Recall that a public-key encryption scheme $\mathsf{HE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ consists of a key generation algorithm $(ek, dk) \leftarrow \mathsf{Kg}(1^{\lambda})$, an encryption algorithm $c \leftarrow \mathsf{Enc}(ek, m)$, and a decryption algorithm $m \leftarrow \mathsf{Dec}(dk, c)$ and the corresponding security experiment of CCA is a two stage game. In the first stage, the attacker has access to a decryption oracle and may query this oracle on arbitrary values. Subsequently, the attacker outputs two messages of equal length and receives a challenge ciphertext that encrypts one of the messages depending on a randomly chosen bit b. In the second stage of the experiment, the attacker gets access to a modified decryption oracle that answers all queries, except for the challenge ciphertext. Eventually, the attacker outputs a bit b' trying to predict b and it wins the security experiment if its success probability is non-negligibly bigger than $1/2$.

In our formalization, the game of CCA security is a 2-round experiment. The initialization algorithm Init generates a key-pair (ek, dk) of a public-key encryption scheme, it chooses a random bit b, and sets $i = 1, r = 2$ and $c_b = \emptyset$. The public parameters pp contain (ek, i, r, c_b) and the private state is (dk, b). The input of the oracle OA is (pp, x), it parses pp as (ek, i, r, c_b) and behaves as follows: If $i = 1$, then it returns the decryption of x, i.e., it outputs $y = \mathsf{Dec}(dk, x)$. If $i = 2$, then OA outputs $\mathsf{Dec}(dk, x)$ if $x \neq c_b$, and $\perp$ otherwise. At some point, the adversary $\mathcal{A}$ outputs as its response $\mathsf{resp} = (m_0, m_1, \mathsf{st}_{\mathcal{A}})$ two challenges messages m_0, m_1 and some state information $\mathsf{st}_{\mathcal{A}}$. The update algorithm $\mathsf{Update}(\mathsf{st}, \mathsf{resp}, \mathsf{cnt})$ extracts b from st and updates the public parameters pp by replacing c_b with $c_b \leftarrow \mathsf{Enc}(ek, m_b)$ and by setting $i = 2$. Moreover, it stores the messages m_0 and m_1 in st. In the next stage of the experiment, the oracle OA returns $\perp$ when queried with c_b. Eventually, $\mathcal{A}$ outputs a bit b' as its response resp. The test algorithm Test extracts m_0, m_1, and b from st and b' from resp. It returns 0 if $|m_0| \neq |m_1|$ or if $b' \neq b$. Otherwise, it outputs 1.

The Unforgeability Experiment. The classical security experiment of existential unforgeability under chosen messages attacks for signature schemes is not covered by our formalization. The reason is that the testing algorithm outputs 1 if the forged message m^* is different from all queries $m_1, \ldots, m_i$ the attacker $\mathcal{A}$ queried to OA. Thus, the testing algorithm is clearly not oblivious of $\mathcal{A}$'s queries to OA. However, one can easily define a modified experiment that is implied by the classical experiment. Similar to the unforgeability notion of blind signatures, we let the attacker query the signing oracle q times and the attacker succeeds if it outputs $q+1$ messages-signature pairs such that all messages are distinct and all signatures are valid. Clearly, giving a successful adversary against this modified game, one can easily break the classical notion by guessing which of the $q+1$ pairs is the forgery.

2.2 Oblivious Black-Box Reductions

Hard Computational Problem. We recall the definition of hard computational problems due to Naor [45].

Definition 3 (Hard Problem). *A computational problem $\pi = (Ch, t)$ is defined by a machine Ch (the challenger) and a threshold function $t = t(\lambda)$. We say that an adversary $\mathcal{A}$ breaks the problem π with advantage ϵ, if*

$$\Pr[\langle Ch, \mathcal{A} \rangle = 1] \geq t(\lambda) + \epsilon(\lambda),$$

over the randomness of Ch and $\mathcal{A}$. If π is non-interactive, then the interaction between $\mathcal{A}$ consists of Ch providing an input instance to $\mathcal{A}$ and $\mathcal{A}$ providing an output to Ch. The problem π is hard *if ϵ is negligible for all efficient adversaries $\mathcal{A}$.*

All standard hardness assumptions used in cryptography can be modeled in this way, for instance the DDH assumption. The goal of a reduction is to show that the security of a cryptographic primitive P can be reduced to some underlying hard assumption. This is shown by contraposition assuming that the cryptographic primitive is insecure with respect to some security experiment. Then, the reduction gets as input an instance of the underlying hard problem, it runs a black-box simulation of the attacker and shows, via simulation of the security experiment, that it can use the adversary to solve the underlying hard problem. Since the problem is assumed to be hard, such an attacker cannot exist. A reduction is black-box if it treats the adversary as a black-box and does not look at the code of the attacker. A comprehensive discussion about the different types of black-box reductions and techniques is given in [55]. For our purposes we need a specific class of black-box reductions that we call *oblivious*. Loosely speaking, a black-box reduction is oblivious if it only knowns an upper bound on the number of oracle queries made by the attacker, but does neither know the query nor the answer. Intuitively, this motion allows the reduction to program the oracle once for each round of the security game.

Definition 4 (Oblivious Black-Box Reductions). *Let* P *be a cryptographic primitive with an associated security experiment* Exp*. Moreover, let* π *be a hard problem. Let* $\mathcal{B}$ *be an oracle algorithm with the following syntax.*

- $\mathcal{B}$ *is an adversary against the problem* π
- $\mathcal{B}$ *has* restricted *black-box access to a machine* $\mathcal{A}$*, which is an adversary for the security experiment* Exp
- $\mathcal{B}$ *gets as auxiliary input an upper bound* q *on the number of oracle queries* $\mathcal{A}$ *makes in each invocation.*

By restricted black-box access to $\mathcal{A}$ *we mean that* $\mathcal{B}$ *is allowed to program an oracle* $\mathcal{O}_{\mathcal{B}}$*, choose inputs* $\mathsf{pp}, \mathsf{st}_{\mathcal{A}}$ *and get the output* $(\mathsf{resp}, \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}}(\cdot)}(\mathsf{pp}, \mathsf{st}_{\mathcal{A}})$*. As before, we assume that* $\mathcal{A}$ *queries its oracle exactly once per invocation (We stress that* $\mathcal{B}$ *does* not *see* $\mathcal{A}$*'s oracle queries).*

We say that $\mathcal{B}$ *is an oblivious black-box reduction from the security of* Exp *to* π *if it holds for every (possibly inefficient) adversary* $\mathcal{A}$ *against* Exp *that if* $\mathbf{Adv}^{\mathsf{Exp}}_{\mathcal{A}}(\lambda)$ *is non-negligible, then* $\mathbf{Adv}^{\pi}_{\mathcal{B}^{\mathcal{A}}}(\lambda)$ *is also non-negligible.*

2.3 Secure Function Evaluation for Cryptographic Primitives

In this section, we propose our security notions for two-round secure function evaluation of cryptographic primitives P. A two-round SFE protocol is a protocol between two parties, a sender $\mathcal{S}$ and a receiver $\mathcal{R}$. The sender provides as input a function f from a family $\mathcal{F}$ and the receiver an input x to the function. At the end of the protocol, the sender gets no output (except for a signal that the protocol is over), whereas the receiver's output is $f(x)$. The function that is realized by our SFE protocols is a function of the primitive P. Since we view P as a collection of algorithms, our SFE protocol evaluates the underlying functionality. For example, in the case of signature schemes this collection consists of a key generation, a signing, and a verification algorithm. Securely evaluating this primitive means to securely evaluate the signing algorithm.

In the following, we introduce our security definitions. Roughly speaking, receiver security says that the security of the underling cryptographic primitive is preserved. This property must hold even against malicious receivers. Moreover, our security notion for the sender holds with respect to semi-honest senders.

Induced Game-Based Malicious Receiver Security. Regarding security, ideally we would like to achieve that the receiver learns nothing but $f(x)$, which is usually modeled via standard simulation based security notions. However, it is well known that standard simulation based security notions fail in the regime of two-round secure function evaluation [29]. Thus, our goal is to achieve a weaker notion of security, which roughly says that the security of the underlying cryptographic primitive is preserved. More precisely, we consider the secure evaluation of cryptographic primitives, which are equipped with a game based security notion. In our formalization the adversary in the corresponding security experiment has black-box access to the primitive. Then, we define an induced security

notion by replacing black-box calls to the primitive in the security game with instances of the two round SFE protocol. I.e., instead of giving the adversary black access to the primitive, it acts as a malicious receiver in an SFE session with the sender. Achieving this notion and showing that the underlying security guarantees are preserved is non-trivial, because the adversary is *not* semi-honest and may not follow the protocol.

Definition 5 (Induced Game-Based Malicious Receiver Security). *Let* $\mathsf{Exp} = (\mathsf{Init}, \mathcal{O}, \mathsf{Update}, \mathsf{Test})$ *be a cryptographic security experiment for a primitive* P*. Let* $\Pi = (\mathcal{S}, \mathcal{R})$ *be a two-round SFE protocol for a function* F *of* P*. The* induced security experiment Exp' *is defined by replacing* $\mathcal{O}$ *with instances of* Π*, where the adversary is allowed to act as a malicious receiver.*

In the following, we study the implications of our security notion with respect to the security of the underlying cryptographic primitive. It is not very difficult to see, that if a protocol is perfectly correct and securely realizes our notion of induced game-based security, then it immediately implies the security of the underlying cryptographic primitive. Second, one can also show that the converse is not true, by giving a counterexample. The basic idea of the counterexample is to build a two-round SFE protocol that completely leaks the circuit and thus the entire private input of the sender. The main result of our paper is a two-round SFE protocol that preserves the underlying security guarantees.

Semi-honest Sender Security. We define security against semi-honest senders via the standard simulation based definition [24].

Definition 6 (Semi-honest Sender Security). *Let* $\Pi = (\mathcal{S}, \mathcal{R})$ *be a two-party protocol for a functionality* F*. We say that* Π *is semi-honest sender secure, if there exists a PPT simulator* Sim *such that it holds for all receiver inputs* x *and all sender inputs* f *that*

$$(x, f, \mathsf{view}(\mathcal{S}), \langle \mathcal{S}, \mathcal{R}(x) \rangle) \overset{\text{COMP.}}{\approx} (x, f, \mathsf{Sim}(\mathsf{f}), f(x))$$

3 2-Round SFE via 1-Hop Homomorphic Encryption

In this section, we present our protocol and prove that it is induced game-based malicious receiver secure (Definitions 5) and semi-honest sender secure (Definition 6).

3.1 1-Hop Homomorphic Encryption

1-hop homomorphic encryption schemes are a special kind of homomorphic encryption schemes that allow a server to compute on encrypted data. Given a ciphertext c produced by the encryption algorithm Enc, the evaluation algorithm Eval can evaluate a circuit C from $\mathcal{C}$ on c. After this no further computation on the output ciphertext is supported. We recall the definition of 1-hop homomorphic encryption schemes and the corresponding notions of security [23].

Definition 7 (1-Hop Homomorphic Encryption). *Let* $\mathcal{C} : \{0,1\}^n \to \{0,1\}^o$ *be a family of circuits. A 1-hop homomorphic encryption scheme* $\mathsf{HE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}, \mathcal{C}_1, \mathcal{C}_2)$ *for* $\mathcal{C}$ *consists of the following efficient algorithms:*

Key Generation. *The input of the key generation algorithm* $\mathsf{Kg}(1^\lambda)$ *is the security parameter* λ *and it returns an encryption key* ek *and a decryption key* dk*.*

Encryption. *The encryption algorithm* $\mathsf{Enc}(ek, m)$ *takes as input an encryption key* ek *and a message* $m \in \{0,1\}^n$ *and returns a ciphertext* $c \in \mathcal{C}_1$*.*

Evaluation. *The evaluation algorithm* $\mathsf{Eval}(ek, c, C)$ *takes as input a public encryption key* ek*, a ciphertext* c *generated by* Enc *and a circuit* $C \in \mathcal{C}$ *and returns a ciphertext* $c' \in \mathcal{C}_2$*.*

Decryption. *The decryption algorithm* $\mathsf{Dec}(dk, c)$ *takes as input a private decryption key* dk *and a ciphertext* c' *generated by* Eval *and returns a message* $y \in \{0,1\}^o$*.*

We recall that the standard notions of completeness and compactness [23]. A homomorphic encryption scheme is complete if the probability of a decryption error is 0. It is compact if the size of the output ciphertext c' of the evaluation algorithm Eval is independent of the size of the circuit C. Moreover, we recall the standard notion of IND-CPA-security for homomorphic encryption schemes: Given a public key ek for the scheme, no PPT adversary succeeds to distinguish encryptions of two adversarially chosen messages m_0 and m_1.

For our purposes we need a homomorphic encryption scheme with malicious circuit privacy. This property says that even if both maliciously formed public key and ciphertext are used, encrypted outputs only reveal the evaluation of the circuit on some well-formed input x^*. We recall the definition in the following.

Definition 8 (Malicious Circuit Privacy). *A 1-hop homomorphic encryption scheme* $\mathsf{HE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}, \mathcal{C}_1, \mathcal{C}_2)$ *for a family* $\mathcal{C}$ *of circuits is (maliciously) circuit private if there exist unbounded algorithms* $\mathsf{Sim}_{\mathsf{HE}}(ek, c, y)$*, and deterministic* $\mathsf{Ext}_{\mathsf{HE}}(ek, c)$ *such that for all* λ*, and all* ek*, all* $c \in \mathcal{C}_1$ *and all circuits* $C \in \mathcal{C}$ *it holds that*

$$\mathsf{Sim}_{\mathsf{HE}}(ek, c, C(x)) \overset{\text{STAT.}}{\approx} \mathsf{Eval}(ek, C, c),$$

where $x = \mathsf{Ext}_{\mathsf{HE}}(ek, c)$*.*

Instantiations. We consider instantiations of maliciously circuit private 1-hop homomorphic encryption. Maliciously circuit private homomorphic encryption for logarithmic depth circuits can be achieved by combining information-theoretic garbled circuits (aka randomized encodings) [2,33,38] with two-message oblivious transfer [1,30,48].

Theorem 1 [1,2,30,33,38,48]. *Under numerous number-theoretic assumptions, there exist a non-compact maliciously circuit private homomorphic encryption scheme that support circuits of logarithmic depth.*

Ostrovsky et al. [52] provide a construction that bootstraps a maliciously circuit privacy scheme that supports only evaluation of logarithmic depth circuits into a scheme that supports all circuits (i.e., it is fully homomorphic).

Theorem 2 (Theorem 1 in [52]). *Assume there exists a compact semi-honest circuit private fully homomorphic encryption scheme* FHE *with decryption circuits of logarithmic depth and perfect completeness. Assume further that there exists a (non-compact) maliciously circuit private homomorphic encryption scheme for logarithmic depth circuits. Then there exists a maliciously circuit private fully homomorphic encryption scheme with perfect completeness.*

3.2 Construction

We can now state the two message SFE protocol. If f is a cryptographic functionality that takes input s from the sender, input x from the receiver and random coins r, we augment the functionality such that both parties contribute to the random coins. I.e., both parties also input random string $r_{\mathcal{S}}$ and $r_{\mathcal{R}}$ and the random coins for the functionality is set to $r_{\mathcal{S}} \oplus r_{\mathcal{R}}$.

Construction 1. *Let* HE *be a 1-hop homomorphic encryption scheme. The interactive protocol that realizes* $\mathcal{F} : (s, r_{\mathcal{S}}, x, r_{\mathcal{R}}) \to (\bot, f(s, r_{\mathcal{S}}; x, r_{\mathcal{R}}))$ *is shown in Fig. 3.*

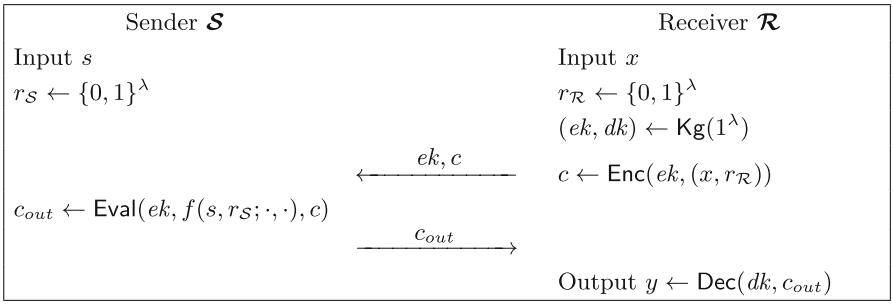

Fig. 3. Oblivious two-party protocol

The following theorem shows that security against malicious receivers with respect to our definition of induced game-based security.

Theorem 3. *Let* P *be a cryptographic primitive and* Exp *be the corresponding security experiment. If there exists an efficient oblivious black-box reduction* $\mathcal{B}$ *that reduces security of* P *to a hard problem* π*, then the protocol* Π *is secure with respect to* Exp'*. Formally, there exists an efficient reduction* $\mathcal{B}'$ *that reduces the security of* Π *to* π*.*

Proof. Assume there exists a PPT adversary $\mathcal{A}$ that has non-negligible advantage ϵ_1 in the security experiment Exp'.

Step 1. In the first step, we change the security experiment Exp' to an indistinguishable experiment Exp^*. In particular, we implement $\mathcal{A}$'s oracles differently. In Exp', the oracle gets a sender-input $(s, r_{\mathcal{S}})$ and a receiver-message (ek, c), computes $c' \leftarrow \mathsf{Eval}(ek, f(s, r_{\mathcal{S}}; \cdot, \cdot), c)$ and outputs c' to $\mathcal{A}$. In Exp^*, the oracle is implemented as follows. Given sender-input $(s, r_{\mathcal{S}})$ and a receiver-message (ek, c), the oracle first computes $(x, r_{\mathcal{R}}) \leftarrow \mathsf{Ext}_{\mathsf{HE}}(ek, c)$. Then it computes $y \leftarrow f(s, r_{\mathcal{S}}; x, r_{\mathcal{R}})$ and then $c' \leftarrow \mathsf{Sim}_{\mathsf{HE}}(ek, c, y)$ and finally outputs c' to $\mathcal{A}$

$\mathsf{OA}_1(ek, c)$	$\mathsf{OA}_2(ek, c)$
$c' \leftarrow \mathsf{Eval}(ek, f(s, r_{\mathcal{S}}; \cdot, \cdot)$	$(x, r_{\mathcal{R}}) \leftarrow \mathsf{Ext}_{\mathsf{HE}}(ek, c)$
Return c'	$y \leftarrow f(s, r_{\mathcal{S}}; x, r_{\mathcal{R}})$
	$c' \leftarrow \mathsf{Sim}_{\mathsf{HE}}(ek, c, y)$
	Return c'

We claim that $\epsilon_2 = \mathbf{Adv}_{\mathsf{Exp}^*}(\mathcal{A}) \geq \mathbf{Adv}_{\mathsf{Exp}'}(\mathcal{A}) - \mathsf{negl}(\lambda)$. We establish this via a hybrid argument. Assume that $\mathcal{A}$ makes at most $\ell = \mathsf{poly}(\lambda)$ oracle queries. Define $\ell + 1$ hybrid experiments $\mathcal{H}_0, \ldots, \mathcal{H}_\ell$. $\mathcal{H}_0$ simulates the oracle as in experiment $\mathsf{Exp}'(\mathcal{A})$, whereas $\mathcal{H}_\ell$ simulates it as in $\mathsf{Exp}^*(\mathcal{A})$. In $\mathcal{H}_i$ the first i oracle queries to $\mathcal{A}$ are answered as in $\mathsf{Exp}'(\mathcal{A})$, whereas the last $\ell - i$ oracle queries of $\mathcal{A}$ are answered as in $\mathsf{Exp}^*(\mathcal{A})$. It follows by the statistical circuit privacy of HE that the statistical distance between each $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ is at most ν for a negligible ν. Thus, by the triangle inequality the statistical distance between $\mathsf{Exp}'(\mathcal{A})$ and $\mathsf{Exp}^*(\mathcal{A})$ is at most $\ell \cdot \nu$, which is negligible. Note that the experiment Exp^* is not efficient anymore.

Step 2. The second step is a bridging step: We move both the extractor $\mathsf{Ext}_{\mathsf{HE}}$ and the simulator $\mathsf{Sim}_{\mathsf{HE}}$ into a new adversary $\mathcal{A}_2$, which internally simulates $\mathcal{A}$. The adversary $\mathcal{A}_2$ is an *unbounded* adversary against the experiment Exp with advantage ϵ_2. Adversary $\mathcal{A}_2$ works as follows. When adversary $\mathcal{A}$ sends an oracle query (ek, c), $\mathcal{A}_2$ computes $(x, r_{\mathcal{R}}) \leftarrow \mathsf{Ext}_{\mathsf{HE}}(ek, c)$ and sends x to its own oracle (in the Exp experiment). Once it receives an oracle output y, it computes $c' \leftarrow \mathsf{Sim}_{\mathsf{HE}}(ek, c, y)$ and forwards c' to $\mathcal{A}$.

Adversary $\mathcal{A}_2(\mathsf{pp}, \mathsf{st}_{\mathcal{A}})$	Oracle $\mathsf{OA}'(ek, c)$
Has access to oracle OA	$(x, r_{\mathcal{R}}) \leftarrow \mathsf{Ext}_{\mathsf{HE}}(ek, c)$
$(\mathsf{resp}, \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}^{\mathsf{OA}'(\cdot)}(\mathsf{pp}, \mathsf{st}_{\mathcal{A}})$	$y \leftarrow \mathsf{OA}(x)$
Return $(\mathsf{resp}, \mathsf{st}_{\mathcal{A}})$	$c' \leftarrow \mathsf{Sim}_{\mathsf{HE}}(ek, c, y)$
	Return c'

We claim that $\mathsf{Exp}(\mathcal{A}_2)$ is identically distributed to $\mathsf{Exp}^*(\mathcal{A})$. To see this, note that we've just regrouped the algorithms $\mathsf{Ext}_{\mathsf{HE}}$ and $\mathsf{Sim}_{\mathsf{HE}}$ into $\mathcal{A}_2$ and removed the dependency of y from $r_{\mathcal{R}}$. However, since $f(s, r_{\mathcal{S}}; x, r_{\mathcal{R}})$ computes the function $F(s, x, r_{\mathcal{S}} \oplus r_{\mathcal{R}})$, the distribution of y does not depend on $r_{\mathcal{R}}$ (as $r_{\mathcal{S}}$ is chosen uniformly at random).

Step 3. In the third step, we combine the adversary $\mathcal{A}_2$ with the oblivious black-box reduction $\mathcal{B}$, which yields an (unbounded) adversary $\mathcal{B}^{\mathcal{A}_2}$ with non-negligible advantage ϵ_3 against the hard problem π (as $\epsilon_2 = \epsilon_1 - \mathsf{negl}(\lambda)$ is non-negligible).

Note at this stage that while the reduction $\mathcal{B}$ is efficient, the π-adversary $\mathcal{B}^{\mathcal{A}_2}$ is not efficient as $\mathcal{A}_2$ is not efficient.

Step 4. The fourth step is again a bridging step: We move the extractor $\mathsf{Ext}_{\mathsf{HE}}$ and the simulator $\mathsf{Sim}_{\mathsf{HE}}$ into the oracle simulated by $\mathcal{B}$, thus obtaining a new reduction $\mathcal{B}^*$. More precisely, $\mathcal{B}^*$ simulates $\mathcal{B}$, but when $\mathcal{B}$ invokes the adversary with input $(\mathsf{pp}, \mathsf{st}_{\mathcal{A}})$ and oracle circuit OA, $\mathcal{B}^*$ constructs the following new oracle OA^*. On input (ek, c), OA^* computes $(x, r_{\mathcal{R}}) \leftarrow \mathsf{Ext}_{\mathsf{HE}}(ek, c)$, $y \leftarrow \mathsf{OA}(x)$, $c' \leftarrow \mathsf{Sim}_{\mathsf{HE}}(ek, c, y)$ and outputs c'. We claim that $\langle Ch, \mathcal{B}^{\mathcal{A}_2} \rangle$ and $\langle Ch, \mathcal{B}^{*\mathcal{A}} \rangle$ are identically distributed (where Ch is the challenger for the hard problem π). In fact, we have merely rearranged the algorithms $\mathsf{Ext}_{\mathsf{HE}}$ and $\mathsf{Sim}_{\mathsf{HE}}$ from the adversary $\mathcal{A}_2$ into the oracle OA^*. Note that now the reduction $\mathcal{B}^*$ is inefficient, whereas the adversary $\mathcal{A}$ is efficient.

Step 5. In the fifth and final step, we change the way the reduction $\mathcal{B}^*$ implements its oracles, obtaining an efficient reduction $\mathcal{B}'$. We will use the circuit privacy of HE a second time to implement the oracles efficiently. Whereas $\mathcal{B}^*$ constructs oracle circuit OA^* from oracle circuit OA provided by $\mathcal{B}$, $\mathcal{B}'$ proceeds as follows. On input (ek, c), the oracle OA' computes $c' \leftarrow \mathsf{Eval}(ek, \overline{\mathsf{OA}}(\cdot), c)$ and outputs c'. Define the circuit $\overline{\mathsf{OA}}$ to compute the function $\overline{\mathsf{OA}}(x, r) = \mathsf{OA}(x)$, i.e., it just drops its second input. Using the malicious circuit privacy of HE, we can establish that $\langle Ch, \mathcal{B}^{*\mathcal{A}} \rangle$ and $\langle Ch, \mathcal{B}'^{\mathcal{A}} \rangle$ are statistically close in the same fashion as in step 1. Finally, note that both $\mathcal{B}'$ and $\mathcal{A}$ are efficient, therefore $\mathcal{B}'^{\mathcal{A}}$ is efficient. □

The following theorem shows that our protocol is secure against semi-honest senders. Note that achieving security against malicious senders is not possible (under standard assumptions). The corresponding impossibility results is given in Sect. 5.

Theorem 4. *If* HE *is an IND-CPA secure 1-hop homomorphic encryption scheme, then* Π *is secure against semi-honest senders.*

Proof. We will first provide the simulator Sim. The main idea of Sim is to run the protocol Π between a simulated sender $\mathcal{S}$ and a simulated receiver $\mathcal{R}$, where the receivers input is 0^n. After the protocol terminates, Sim outputs the view of the sender $\mathcal{S}$.

Now assume there exists a PPT distinguisher $\mathcal{D}$ that distinguishes the distributions $(x, s, \mathsf{view}(\mathcal{S}), \langle \mathcal{S}, \mathcal{R}(x) \rangle)$ and $(x, f, \mathsf{Sim}(\mathsf{s}), f(x))$ with non-negligible advantage ϵ for some inputs s and x. We will construct an adversary $\mathcal{A}$ that breaks the IND-CPA security of HE with advantage ϵ. Given a public key ek, $\mathcal{A}$ chooses a random $r_{\mathcal{R}}$ and $r'_{\mathcal{R}}$, sets $m_0 = (x, r_{\mathcal{R}})$ and $m_1 = (0^n, r'_{\mathcal{R}})$ and sends (m_0, m_1) to the IND-CPA experiment. Let c be the challenger ciphertext. $\mathcal{A}$ chooses random coins $r_{\mathcal{S}}$, and runs $\mathcal{S}$ with input s, $r_{\mathcal{S}}$ and receiver message c. Let $\mathsf{view}(\mathcal{S})$ be the simulated view of the sender. Next, $\mathcal{A}$ computes $y = f(s, r_{\mathcal{S}}; x, r_{\mathcal{R}})$. Finally, $\mathcal{A}$ runs $\mathcal{D}$ on input $(x, s, \mathsf{view}(\mathcal{S}), y)$ and outputs whatever $\mathcal{D}$ outputs.

We claim that $\mathcal{A}$ breaks the IND-CPA security of HE with advantage ϵ. First assume the IND-CPA challenge bit is 0. In this case the IND-CPA returns to $\mathcal{A}$ an encryption of $m_0 = (x, r_{\mathcal{R}})$. Thus, the view that $\mathcal{A}$ simulates is identically distributed to $\mathcal{S}$'s view in the real experiment, but also the output y has the same distribution as in the real experiment. On the other hand, if the IND-CPA choice bit is 1, then $\mathsf{view}(\mathcal{S})$ is identically distributed to $\mathsf{Sim}(s)$ and the output $y = f(s, r_{\mathcal{S}}; x, r_{\mathcal{R}})$ is independently distributed of $\mathsf{view}(\mathcal{S})$ (as $r_{\mathcal{R}}$ is independent of $\mathsf{view}(\mathcal{S})$). Thus we conclude

$$\begin{aligned}\mathbf{Adv}_{\text{IND-CPA}}(\mathcal{A}) &= |\Pr[\text{IND-CPA}^0(\mathcal{A}) = 1] - \Pr[\Pr[\text{IND-CPA}^1(\mathcal{A}) = 1]| \\ &= |\Pr[\mathcal{D}(x, s, \mathsf{view}(\mathcal{S}), \langle \mathcal{S}(s), \mathcal{R}(x)\rangle) = 1] \\ &\quad - \Pr[\mathcal{D}(x, s, \mathsf{Sim(s)}, f(s; x)) = 1]| = \epsilon,\end{aligned}$$

which concludes the proof. □

4 Round-Optimal Oblivious Pseudorandom Functions

Our technique yields the first two message oblivious pseudorandom function in the standard model. Oblivious pseudorandom functions are in essence pseudorandom functions that are obliviously evaluated in a two-party protocol. This means that the sender $\mathcal{S}$ holds a key k of a PRF F and the receiver $\mathcal{R}$ a value x and wishes to learn $F(k, x)$. As already discussed in the introduction, OPRFs have many applications, such as private key-word search [17], or secure computation of set intersection [34]. However, despite the popularity of this primitive, no scheme in the standard model is known with only two rounds of communication. Regarding the security of OPRFs, we wish to express that the sender $\mathcal{S}$ does not learn anything about the value x, and the receiver $\mathcal{R}$ learns only the pseudorandom value $F(k, x)$. First recall the standard definition of pseudorandom functions.

Definition 9 (Pseudorandom Functions). *An efficiently computable two-argument function PRF is called pseudorandom function, if it holds for every PPT distinguisher $\mathcal{D}$ that*

$$\mathbf{Adv}(\mathcal{D}) = |\Pr[\mathcal{D}^{PRF(k,\cdot)} = 1] - \Pr[\mathcal{D}^{H(\cdot)} = 1]| \leq \mathsf{negl}(\lambda),$$

where k is a randomly chosen key F and H is a random function with the same domain and range as F.

We will now turn to the standard definition of oblivious pseudorandom functions. This notion require for an OPRF protocol Π two properties to be satisfied. First, we require that Π is a secure two-party protocol realizing a function $F(k, x)$, where k is the sender input and x is the receiver input. Second, we require that $F(k, \cdot)$ is a pseudorandom function. The first part of this definition captures the idea that the π allows the receiver to learn one function value of $F(k, \cdot)$ per invocation only. The second requirement ensures that such function values are pseudorandom.

While this definition is appealing due to its modularity, it is impossible in the two message setting, even if we only consider semi-honest senders[1]. To circumvent this impossibility, we propose a security notion which captures both intuitive requirements in a single definition. In this definition, a PPT distinguisher is given access to an oracle, which implements either an OPRF sender or an unbounded simulator Sim with access to a truly random function H. Since we are considering two-message OPRF protocols, the distinguisher's queries to its oracle are simply the first message of a malicious receiver. Since we are in the two-message setting, the simulator has a very simple structure: It extracts the receiver's queries by brute force, forwards them to the random function H, and then simulates a response by the sender using the random function's output. This definition contains a minor loophole: It does not rule out trivial simulators, i.e., it does not require the simulator to use the random function it is given access to at all. The simulator could do anything, even simulating the real protocol (which would give perfect indistinguishability between the two distributions), which would defeat the purpose of the definition. To fix this, we will give the distinguisher direct access to the random function H. In the real execution, this is mirrored by giving the distinguisher access to an oracle that implements an honest receiver interacting with the sender. Now, the distinguisher can actually cross check the answers of the simulator. This definition has some flavor of the universal composability framework [9] and Nielsen's definition of non-programmable random oracles [50]. Think of a complex scenario where multiple receivers interact with one OPRF sender (e.g. a server). We may think of the distinguisher in our definition as an environment in control of several malicious receivers over which it has full control, but it can also choose inputs and observe outputs of honest receivers. Then this definition requires that from the environments view the OPRF server *looks like* it actually implements a truly random function.

Definition 10 (Security Against Malicious Receiver for Oblivious Pseudorandom Functions). *Let $\Pi = (\mathcal{S}, \mathcal{R})$ be a two-message protocol. We say that π is a two-message oblivious pseudorandom function, if for every PPT distinguisher $\mathcal{D}$ there exists a (possibly unbounded) simulator Sim, such that*

$$\mathbf{Adv}(\mathcal{D}) = |\Pr[\mathcal{D}^{\langle \mathcal{S}(k),\cdot\rangle,\langle \mathcal{S}(k),\mathcal{R}(\cdot)\rangle} = 1] - \Pr[\mathcal{D}^{\mathsf{Sim}^{H}(\cdot),H(\cdot)} = 1]| \leq \mathsf{negl}(\lambda),$$

where k is a randomly chosen input for $\mathcal{S}$ and H is a random function (with appropriate domain and range). Here, $\langle \mathcal{S}(k), \cdot\rangle$ is a session of π where $\mathcal{D}$ can choose the first message of the receiver and receives the second message by the sender. In $\langle \mathcal{S}(k), \mathcal{R}(\cdot)\rangle$, $\mathcal{D}$ chooses the input for $\mathcal{R}$ and obtains the output of $\mathcal{R}$.

The security guarantee for the receiver is the standard simulation based security against semi-honest senders (Definition 6).

[1] The impossibility is analogous to the impossibility of simulation based two message oblivious transfer. Consider a malicious receiver that gets auxiliary input z, which the malicious receiver sends as its first message. An efficient simulator Sim for this malicious receiver must extract in input x given only z.

Remark 2. We remark several points. First, if a simulator Sim is non-trivial by construction, we can omit the second oracle of the distinguisher. Basically, the only property we need to ensure non-triviality is that if the simulator gets messages from an honest receiver, then this composed system actually implements in the random function H. Formally, this requirement can be written as $\langle \mathsf{Sim}^H, \mathcal{R}(\cdot)\rangle \equiv H(\cdot)$, i.e., if an honest receiver interacts with a simulator Sim with access to H, then this protocol implements H. If this is guaranteed, then the oracles $\langle \mathcal{S}(k), \cdot\rangle$ and $\mathsf{Sim}^H(\cdot)$ are sufficient: Given such an oracle OA (which is either of the two), the distinguisher $\mathcal{D}$ can simulate the *honest* oracle by $\langle \mathsf{OA}, \mathcal{R}(\cdot)\rangle$. In our construction the simulator Sim will be canonical: It extracts the first message, sends the extracted input to the random function H, and uses the output to simulate the senders message. This simulator is non-trivial by construction, and thus giving the distinguisher access to a single oracle will be sufficient. Moreover, while Definition 10 allows the simulator Sim to depend on the distinguisher $\mathcal{D}$, our canonic simulator will be universal in the sense that it works for any PPT distinguisher $\mathcal{D}$.

Pseudorandom Functions with Oblivious Black-Box Reductions. To apply the technique developed in Sect. 3, we require a pseudorandom function with an oblivious black-box reduction. Most constructions of PRFs in the literature do not possess such a reduction. In particular, most reductions need to *program* the distinguishers oracle adaptively depending on prior oracle inputs of the distinguisher. For example, the security reduction of the construction of Goldreich, Goldwasser and Micali [25], which reduces the security of the PRF on that of the underlying pseudorandom generator is based on a hybrid argument and needs to keep a list of the distinguisher's distinct oracle queries to be able to answer oracle queries consistently. This however contradicts our notion of obliviousness.

Fortunately, there are constructions of pseudorandom functions with oblivious black-box reductions to their underlying hard problems. One example of such a PRF is the Naor Reingold PRF [49]. While the security reduction provided in [49] is not oblivious, there is simple way of converting this reduction into an oblivious black-box reduction using q-wise independent functions (Appendix A). More generally, there is a recent line of work that aims at constructing large-domain pseudorandom functions from small-domain pseudorandom functions via oblivious black-box reductions [8,14]. The baseline of these results is that large domain PRFs can be constructed by combining several small-domain (i.e., poly-sized domain) PRFs in a suitable way. The pseudorandomness of large domain PRFs is established by replacing one of the small-domain PRFs (depending on the query bound of the adversary) with a random function in a single shot. Since the small-domain PRF has a domain of just polynomial size, the reduction can (non-adptively) query its oracle on all inputs and retrieve the entire function table. Thus, there is no need of adaptively programming the distinguishers oracle based on previous queries. In order to use the framework we developed in Sect. 3, it will be convenient to use an alternative definition of pseudorandom

functions. In Definition 9, the distinguishers goal is to distinguish the PRF from a truly random function. However, if we do not know any bound on the distinguisher's number of queries in advance, the only (known) way to simulate a random function is by evaluating the random function lazily: Every time the distinguisher queries the random function on a new input, the simulation samples a random image and adds it into a table of input and output values. If a certain input has been queried before, it's image is retrieved from the table. However, such a simulation is necessarily stateful. To overcome this, we use an equivalent definition of pseudorandom functions which takes into account that a every PPT distinguisher has a polynomial upper bound on the number of its oracle queries. Once such a bound q is known, we can simulate a random function statelessly with an efficient q-wise independent function.

Definition 11 (q-Wise Independent Function). *Let F be an efficiently computable two argument function that takes a seed s and an input x. We say that F is a q-wise independent functions, if it holds for all pairwise distinct $x_1, \ldots, x_q$ that $F(s, x_1), \ldots, F(s, x_q)$ are distributed independently and uniformly random over the choice of the seed s.*

There are various constructions of efficient q-wise independent functions, such as the classical construction of Wegman and Carter [57] which is based on random degree q polynomials in large finite fields.

Definition 12 (Pseudorandom Functions, Equivalent Definition). *An efficiently computable two-argument function PRF is called pseudorandom function, if there exists a family $\{F_q\}_q$ of functions, where F_q is q-wise independent, such that the following holds. For every $q = \mathsf{poly}(\lambda)$ and every PPT distinguisher $\mathcal{D}$ that queries its oracle at most q times it holds that*

$$\mathbf{Adv}(\mathcal{D}) = |\Pr[\mathcal{D}^{PRF(k,\cdot)} = 1] - \Pr[\mathcal{D}^{F_q(s,\cdot)} = 1]| \leq \mathsf{negl}(\lambda),$$

where k is a randomly chosen key for PRF and s is a randomly chosen seed for F_q.

Theorem 5 [8,14,49]. *Under various standard hardness assumptions (pseudorandom generators, DDH, LWE) there exist pseudorandom functions with oblivious black-box reduction to their underlying hardness assumption.*

Construction. The construction is expectably simple. We combine Construction 1 with a pseudorandom function that possesses an oblivious black-box reduction to some hard problem π, which is provided by Theorem 5. For this instantiation, we need to instantiate Construction 1 with a maliciously circuit private *fully* homomorphic encryption scheme (such as provided by Theorem 2), as there is no a priori upper bound on the size of the circuits that implement q-wise independent functions. For convenience, we write down the protocol as follows. Let PRF be a pseudorandom function and HE be a fully homomorphic encryption scheme. The OPRF protocol Π is given as follows.

Protocol Π_{OPRF}
Setup
$\mathcal{S}_0(1^\lambda)$: Choose a random key k for PRF
Query
$\mathcal{R}_1(x)$
$(ek, sk) \leftarrow \mathsf{Kg}(1^\lambda)$
$c \leftarrow \mathsf{Enc}(ek, x)$
Send (ek, c) to $\mathcal{S}$
$\mathcal{S}(k, (ek, c))$:
$c' \leftarrow \mathsf{Eval}(ek, PRF(k, \cdot), x)$
Send c' to $\mathcal{R}$
$\mathcal{R}_2(c')$:
$y \leftarrow \mathsf{Dec}(sk, c')$
Output y

We can now prove the main theorem of this section.

Theorem 6. *Let* HE *be an IND-CPA secure maliciously circuit private fully homomorphic encryption scheme with perfect completeness (as provided by Theorem 2) and PRF be a pseudorandom function with an oblivious black-box reduction to hard problem π. Then the protocol Π_{OPRF} is an OPRF protocol with security against semi-honest senders and malicious receivers.*

Proof. We begin with the proof of security against malicious receivers defining the universal simulator Sim. Let $\mathsf{Ext}_{\mathsf{HE}}$ and $\mathsf{Sim}_{\mathsf{HE}}$ be the extractor and simulator for the statistical circuit privacy of HE. Simulator Sim is given as follows.

Simulator $\mathsf{Sim}^H(ek, c)$
Has oracle access to a function H
$x \leftarrow \mathsf{Ext}_{\mathsf{HE}}(ek, c)$
$y \leftarrow H(x)$
$c' \leftarrow \mathsf{Sim}_{\mathsf{HE}}(ek, y, c)$
return c'

Now, let $\mathcal{D}$ be a PPT distinguisher that makes at most $q = \mathsf{poly}(\lambda)$ oracle queries and has non-negligible advantage ϵ against the malicious receiver security experiment of Π_{OPRF}, i.e.,

$$|\Pr[\mathcal{D}^{\langle \mathcal{S}(k),\cdot\rangle,\langle \mathcal{S}(k),\mathcal{R}(\cdot)\rangle} = 1] - \Pr[\mathcal{D}^{\mathsf{Sim}^H(\cdot),H(\cdot)} = 1]| \geq \epsilon.$$

First of all, notice that since $\mathcal{D}$ makes at most q queries to its oracles, we can efficiently (and statelessly) simulate the random function H by an efficiently computable q-wise independent function F_q, i.e., we get

$$|\Pr[\mathcal{D}^{\langle \mathcal{S}(k),\cdot\rangle,\langle \mathcal{S}(k),\mathcal{R}(\cdot)\rangle} = 1] - \Pr[\mathcal{D}^{\mathsf{Sim}^{F_q(s,\cdot)}(\cdot),F_q(s,\cdot)} = 1]| \geq \epsilon.$$

Our proof strategy will now be as follows. We will use $\mathcal{D}$ to construct a distinguisher $\mathcal{D}'$ with advantage $\epsilon' = \epsilon - \mathsf{negl}(\lambda)$ against the *induced security*

experiment for PRF under the homomorphic encryption HE (c.f. Definition 5). Recall that the pseudorandom function PRF possesses an oblivious black-box reduction $\mathcal{B}$ to some hard problem π. Thus, Theorem 3 yields an efficient reduction $\mathcal{B}'$ such that $\mathcal{B}'^{\mathcal{D}'}$ has non-negligible advantage against π, contradicting its hardness.

We will now consider the induced security experiment for PRF. Therefore, we will first define a sender algorithm $\mathcal{S}'$. Basically, $\mathcal{S}'$ homomorphically evaluates the q-wise independent function F_q.

$\mathcal{S}'(s,(ek,c))$
$c' \leftarrow \mathsf{Eval}(ek, F_q(s,\cdot), c)$
return c'

Thus, while $\mathcal{S}$ homomorphically evaluates the pseudorandom function PRF, $\mathcal{S}'$ homomorphically evaluates the q-wise independent function F_q. Thus, the induced security experiment of the experiment given in Definition 12 asks to distinguish the oracles $\langle \mathcal{S}(k),\cdot\rangle$ and $\langle \mathcal{S}'(s),\cdot\rangle$.

We will now construct a distinguisher $\mathcal{D}'$ against the induced security experiment of PRF using the distinguisher $\mathcal{D}$. $\mathcal{D}'$ is given as follows.

Distinguisher $\mathcal{D}'(1^\lambda)$ Has access to oracle OA_1 $\mathsf{out} \leftarrow \mathcal{D}^{\mathsf{OA}_1(\cdot),\mathsf{OA}_2(\cdot)}(1^\lambda)$ Return out	Oracle $\mathsf{OA}_2(x)$ $y \leftarrow \langle \mathsf{OA}_1, \mathcal{R}(x)\rangle$ Return y

We claim that

$$|\Pr[\mathcal{D}'^{\langle \mathcal{S}(k),\cdot\rangle} = 1] - \Pr[\mathcal{D}^{\langle \mathcal{S}'(s),\cdot\rangle} = 1]| \geq \epsilon - \mathsf{negl}(\lambda), \tag{1}$$

i.e., $\mathcal{D}'$ has non-negligible advantage $\epsilon - \mathsf{negl}(\lambda)$ against the induced security experiment of PRF.

We claim that if $\mathsf{OA}_1 = \langle \mathcal{S}(k),\cdot\rangle$, then the output of $\mathcal{D}'^{\langle \mathcal{S}(k),\cdot\rangle}(1^\lambda)$ is identically distributed to the output $\mathcal{D}^{\langle \mathcal{S}(k),\cdot\rangle,\langle \mathcal{S}(k),\mathcal{R}(\cdot)\rangle}(1^\lambda)$. To see this, note that the oracle OA_2 implemented by $\mathcal{D}'$ is precisely $\langle \mathcal{S}(k),\mathcal{R}(\cdot)\rangle$ in this case.

On the other hand, if $\mathsf{OA}_1 = \langle \mathcal{S}'(s),\cdot\rangle$, then we claim that the output of $\mathcal{D}'^{\langle \mathcal{S}'(s),\cdot\rangle}$ is distributed statistically close to the output of $\mathcal{D}^{\mathsf{Sim}^{F_q}(\cdot),F_q(\cdot)}(1^\lambda)$. To see this, note first that in this case the oracle OA_2 provided by $\mathcal{D}'$ to $\mathcal{D}$ can be expressed as follows.

$\mathsf{OA}_2(x)$
$(ek, sk) \leftarrow \mathsf{Kg}(1^\lambda)$
$c \leftarrow \mathsf{Enc}(ek, x)$
$c' \leftarrow \mathsf{Eval}(ek, F_q(s,\cdot), c)$
$y \leftarrow \mathsf{Dec}(sk, c')$
return y

It follows immediately from the perfect completeness of HE that OA_2 implements exactly $F_q(s,\cdot)$. It remains to show that the oracles $\langle \mathcal{S}'(s),\cdot\rangle$ and $\mathsf{Sim}^{F_q}(\cdot)$ are statistically close. However, as $\mathcal{S}'(s)$ homomorphically evaluates F_q, it follows from the malicious circuit privacy of HE that both oracles produce distributions that are statistically close, even given F_q. Thus, we can use a standard q-step hybrid argument over the queries of $\mathcal{D}$ to establish that $\mathcal{D}'^{\langle \mathcal{S}'(s),\cdot\rangle}$ and $\mathcal{D}^{\mathsf{Sim}^{F_q}(\cdot),F_q(\cdot)}(1^\lambda)$ are statistically close. Thus, (1) follows and we can apply Theorem 3 to arrive at a contradiction. Security against semi-honest senders follows directly from Theorem 4, which concludes the proof. □

5 Impossibility of Malicious Sender Security

In this section, we show that malicious receiver security (w.r.t. our notion of induced game-based security) and malicious sender security cannot be achieved simultaneously. Our impossibility result is constructive in the sense that we show that our framework covers the standard security notion of blind signatures. However, Fischlin and Schröder showed that a large class of three-move blind signature schemes cannot be proven secure under standard assumptions [16]. Since our framework falls into this class, the impossibility result follows.

Blind Signatures. Blind signatures [11] implement a carbon copy envelope allowing a signer to issue signatures for messages such that the signer's signature on the envelope is imprinted onto the message in the sealed envelope. In particular, the signer remains oblivious about the message (blindness), but at the same time no additional signatures without the help of the signer can be created (unforgeability). Constructing round-optimal blind signature schemes in the standard model has been a long standing open question. Fischlin and Schröder showed that all previously known schemes having at most three rounds of communication, cannot be proven secure under non-interactive assumptions in the standard model via black-box reductions [16]. Subsequently, several works used a technique called "complexity leveraging" to circumvent this impossibility result [19,20] and recently, Fuchsbauer, Hanser, Slamanig suggested a round optimal blind signature scheme that is secure in the generic group model [18]. In fact, it is still unknown if round optimal blind signatures, based on standard assumptions, exist in the standard model.

By applying our technique to the oblivious computation of signatures, we obtain a round optimal blind signature scheme without complexity leveraging and whose security can be based on standard cryptographic assumptions. Since our scheme belongs to the class characterized by Fischlin and Schröder it is not possible to prove blindness w.r.t. malicious adversaries.

Security Definition for Blind Signatures. We recall the unforgeability definition of blind signatures [35,53] that can be expressed within our formalization of a cryptographic experiment.

Definition 13 (Unforgeability). *An interactive signature scheme* $\mathsf{BS} = (\mathsf{KG}, \langle \mathcal{S}, \mathcal{U} \rangle, \mathsf{Vf})$ *is called* unforgeable *if for any efficient algorithm* $\mathcal{A}$ *(the malicious user) the probability that experiment* $\mathsf{Forge}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$ *evaluates to* 1 *is negligible (as a function of* λ*) where*

Experiment $\mathsf{Forge}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$
$(sk, pk) \leftarrow \mathsf{KG}(1^\lambda)$
$((m_1^*, \sigma_1^*), \ldots, (m_{k+1}^*, \sigma_{k+1}^*)) \leftarrow \mathcal{A}^{\langle \mathcal{S}(sk), \cdot \rangle^\infty}(pk)$
Return 1 *iff*
$m_i^* \neq m_j^*$ *for all* i, j *with* $i \neq j$*, and*
$\mathsf{Vf}(pk, m_i^*, \sigma_i^*) = 1$ *for all* i*, and*
$\mathcal{S}$ *has returned* ok *in at most* k *interactions.*

The corresponding definition of blindness says that it should be infeasible for a malicious signer $\mathcal{S}^*$ to decide which of two messages m_0 and m_1 has been signed first in two executions with an honest user $\mathcal{U}$. If one of these executions has returned $\perp$ then the signer is not informed about the other signature (Otherwise the signer could trivially identify one session by making the other abort.). If one restricts this definition the semi-honest adversaries, then this definition is immediately implied by Definition 6.

Construction. Our construction instantiates our general framework as defined in Construction 1 with a signature scheme $\mathsf{DS} = (\mathsf{Kg}_{\mathsf{Sig}}, \mathsf{Sig}, \mathsf{Vf})$ that has an oblivious black-box reduction to some underlying hard problem π. For this instantiation, we need maliciously circuit private homomorphic encryption for logarithmic depth circuits that can be achieved by combining information-theoretic garbled circuits (aka randomized encodings) [2,33,38] with two-message oblivious transfer [1,30,48] as provided by Theorem 1. Moreover, we need a digital signature scheme that can computed via a logarithmic depth circuit. Such a signature scheme can be obtained by using the non-apaptively secure signature scheme by Applebaum et al. [2]. However, this scheme is only non-adaptively secure, which means the adversary has to commit to all messages before learning the public-key and the signature. Using the standard transformation based on chameleon hash functions [31,40] one can convert any non-adaptively secure signature scheme into one that is adaptively secure. Here we actually deal with two reductions. One that deals with adversaries that find collisions of the chameleon hash function and one that deals with adversaries that do not find hash collisions, but still manage to forge signatures. The first reduction is easily seen to be obliviously black-box, as the reduction possesses the signing key for the signature scheme an hash collisions can be easily recovered from the adversary's output. Here the signing circuit is the same as in the real experiment. The second reduction has the following structure. If q is the query bound of the adversary, the reduction computes chameleon hashes on q random values and has them (non-adaptively) signed by the signing oracle. Each time the adversary queries its signing oracle, the reduction uses up one of the precomputed signatures of the chameleon hashes by computing a hash collision with the adversary's query and returning

the corresponding signature to the adversary. Note that since the reduction is allowed to reprogram the signing circuit after each query, we only need to hard-wire a single hash value and trapdoor at a time into the signing oracle circuit. Since chameleon hash functions can easily be obtained from the discrete logarithm assumption involving only two modular exponentiations and a multiplication [40], this transformation can also be computed by a circuit of logarithmic depth. Thus we obtain an oblivious black-box reduction to the non-adaptive unforgeability of the signature scheme where every circuit used by the reduction has a most an a priori known logarithmic depth. We obtain the following theorem.

Theorem 7. *Let* HE *be an IND-CPA secure maliciously circuit private homomorphic encryption scheme with perfect completeness for circuits of logarithmic depth and let* DS *be a signature scheme compute by a circuit of logarithmic depth and with an oblivious black-box reduction to hard problem* π*. Then the protocol* Π_{BS} *defined above is a blind signature protocol with security against semi-honest senders and malicious receivers.*

Given this theorem, we obtain our impossibility result in the following corollary.

Corollary 1 (Impossibility of Malicious Sender Security, Informal). *There exists no two-move secure evaluation protocol for cryptographic functionalities that is secure against malicious receivers and senders based on standard assumptions.*

Acknowledgement. Nico Döttling gratefully acknowledges support by the DAAD (German Academic Exchange Service) under the postdoctoral program (57243032). This work was in part supported by European Research Council Starting Grant 279447. Research supported in part from a DARPA/ARL SAFEWARE award, AFOSR Award FA9550-15-1-0274, and NSF CRII Award 1464397. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government. Nils Fleischhacker, Johannes Krupp and Dominique Schröder were supported by the German Federal Ministry of Education and Research (BMBF) through funding for the Center for IT-Security, Privacy and Accountability (CISPA – www.cispa-security.org) and the project PROMISE. Moreover, it was supported by the Initiative for Excellence of the German federal and state governments through funding for the Saarbrücken Graduate School of Computer Science and the DFG MMCI Cluster of Excellence. Part of this work was also supported by the German research foundation (DFG) through funding for the collaborative research center 1223 and by the DAAD PPP USA program (57129666). We would like to thank the anonymous reviewers of CRYPTO 2016 for their helpful comments.

A An Oblivious Black-Box Reduction for Naor-Reingold PRF

Lemma 1. *The Naor-Reingold PRF is secure under the DDH assumption and the reduction is oblivious.*

Proof. Given an adversary $\mathcal{A}$ who can distinguish the Naor-Reingold PRF with non-negligible probability $\epsilon(\lambda)$ from a truly random function making at most q queries to its oracle, consider the following oblivious reduction $\mathcal{B}$ against DDH:

$\mathcal{B}$ gets as input a DDH instance $(g, g^a, g^b, g^{\tilde{c}})$, where either $\tilde{c} = a \cdot b$ or not. We restrict the reduction to the case where $a, b, \tilde{c} \neq 0$ (otherwise it is trivial to tell whether $\tilde{c} = a \cdot b$). $\mathcal{B}$ will choose a random $j \xleftarrow{\$} \{1, \ldots, \lambda\}$ and pick a random q-wise independent function $F \xleftarrow{\$} \mathcal{F}^q$. It will then sample values $(a_{j+1}, \ldots, a_\lambda) \xleftarrow{\$} \mathbb{Z}_p$ and program the oracle OA for $\mathcal{A}$ as follows:

$\mathsf{OA}(x)$:
$\overline{x}\, x_j \ldots x_\lambda = x$, where $\overline{x}$ is the $(j-1)$-bit prefix of x
$\alpha = F(\overline{x})$
If $x_j = 0$:
Return $\left(\left(g^b\right)^{\alpha}\right)^{\prod_{k=j+1}^{\lambda} a_k^{x_k}}$
else
Return $\left(\left(g^{\tilde{c}}\right)^{\alpha}\right)^{\prod_{k=j+1}^{\lambda} a_k^{x_k}}$

The reduction $\mathcal{B}$ will invoke $\mathcal{A}^{\mathsf{OA}}$ and output 1 exactly whenever $\mathcal{A}^{\mathsf{OA}}$ does.

If $\tilde{c} = a \cdot b$, then for $j = 1$ the oracle perfectly simulates the Naor-Reingold PRF $\mathsf{PRF}_{\vec{a}}$ with key $\vec{a} = (b\alpha, a, a_2, \ldots, a_\lambda)$ (since $\overline{x}$ will be the empty string, α will be constant). Furthermore, if $\tilde{c} \neq a \cdot b$, then for $j = \lambda$ the oracle perfectly simulates a q-wise independent function f (observed as truly random by $\mathcal{A}$):

$$\begin{aligned}
\mathrm{Prob}\left[\mathcal{B}^{\mathcal{A}}(g, g^a, g^b, g^{\tilde{c}}) = 1 \,\middle|\, \tilde{c} = a \cdot b \wedge j = 1\right] &= \mathrm{Prob}\left[\mathcal{A}^{\mathsf{PRF}_{\vec{a}}}(1^\lambda) = 1\right] \\
\mathrm{Prob}\left[\mathcal{B}^{\mathcal{A}}(g, g^a, g^b, g^{\tilde{c}}) = 1 \,\middle|\, \tilde{c} \neq a \cdot b \wedge j = \lambda\right] &= \mathrm{Prob}\left[\mathcal{A}^{f}(1^\lambda) = 1\right]
\end{aligned}$$

Since $g^{\tilde{c}}$ is independent of g^b in case of $\tilde{c} \neq a \cdot b$ it holds that

$$\begin{aligned}
&\mathrm{Prob}\left[\mathcal{B}^{\mathcal{A}}(g, g^a, g^b, g^{\tilde{c}}) = 1 \,\middle|\, \tilde{c} \neq a \cdot b \wedge j = i\right] \\
=&\mathrm{Prob}\left[\mathcal{B}^{\mathcal{A}}(g, g^a, g^b, g^{\tilde{c}}) = 1 \,\middle|\, \tilde{c} = a \cdot b \wedge j = i + 1\right]
\end{aligned}$$

And therefore

$$\begin{aligned}
&\left|\mathrm{Prob}\left[\mathcal{B}^{\mathcal{A}}(g, g^a, g^b, g^{\tilde{c}}) = 1 \;\middle|\; \tilde{c} = a \cdot b\right]\right. \\
&\qquad\left. - \mathrm{Prob}\left[\mathcal{B}^{\mathcal{A}}(g, g^a, g^b, g^{\tilde{c}}) = 1 \;\middle|\; \tilde{c} \neq a \cdot b\right]\right| \\
= &\left|\frac{1}{\lambda} \cdot \sum_{i=1}^{\lambda} \mathrm{Prob}\left[\mathcal{B}^{\mathcal{A}}(g, g^a, g^b, g^{\tilde{c}}) = 1 \;\middle|\; \tilde{c} = a \cdot b \wedge j = i\right]\right. \\
&\qquad\left. - \frac{1}{\lambda} \cdot \sum_{i=1}^{\lambda} \mathrm{Prob}\left[\mathcal{B}^{\mathcal{A}}(g, g^a, g^b, g^{\tilde{c}}) = 1 \;\middle|\; \tilde{c} \neq a \cdot b \wedge j = i\right]\right| \\
= &\frac{1}{\lambda}\left|\mathrm{Prob}\left[\mathcal{B}^{\mathcal{A}}(g, g^a, g^b, g^{\tilde{c}}) = 1 \;\middle|\; \tilde{c} = a \cdot b \wedge j = 1\right]\right. \\
&\qquad\left. - \mathrm{Prob}\left[\mathcal{B}^{\mathcal{A}}(g, g^a, g^b, g^{\tilde{c}}) = 1 \;\middle|\; \tilde{c} \neq a \cdot b \wedge j = \lambda\right]\right| \\
= &\frac{1}{\lambda}\left|\mathrm{Prob}\left[\mathcal{A}^{\mathsf{PRF}_{\vec{a}}}(1^\lambda) = 1\right] - \mathrm{Prob}\left[\mathcal{A}^{f}(1^\lambda) = 1\right]\right| \geq \frac{1}{\lambda}\epsilon(\lambda)
\end{aligned}$$

Thus this reduction will break the DDH assumption with non-negligible probability. As the reduction does not see the queries $\mathcal{A}$ makes to the oracle OA, it is oblivious according to Definition 4. This concludes the proof.

References

1. Aiello, W., Ishai, Y., Reingold, O.: Priced oblivious transfer: how to sell digital goods. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, p. 119. Springer, Heidelberg (2001). 3.1, 1, 5
2. Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography in NC^0. In: 45th Annual Symposium on Foundations of Computer Science, pp. 166–175. IEEE Computer Society Press, October 2004. 3.1, 1, 5
3. Bar-Ilan, J., Beaver, D.: Non-cryptographic fault-tolerant computing in constant number of rounds of interaction. In: Proceedings of the Eighth Annual ACM Symposium on Principles of Distributed Computing, pp. 201–209. ACM (1989). 1.4
4. Beaver, D., Micali, S., Rogaway, P.: The round complexity of secure protocols (extended abstract). In: 22nd Annual ACM Symposium on Theory of Computing, pp. 503–513. ACM Press, May 1990. 1.4
5. Bellare, M., Jakobsson, M., Yung, M.: Round-optimal zero-knowledge arguments based on any one-way function. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 280–305. Springer, Heidelberg (1997). 1.4
6. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (2006). 2, 2.1
7. Bendlin, R., Damgård, I., Orlandi, C., Zakarias, S.: Semi-homomorphic encryption and multiparty computation. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 169–188. Springer, Heidelberg (2011). 1.4
8. Berman, I., Haitner, I.: From non-adaptive to adaptive pseudorandom functions. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 357–368. Springer, Heidelberg (2012). 4, 5
9. Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols. In: 42nd Annual Symposium on Foundations of Computer Science, pp. 136–145. IEEE Computer Society Press, October 2001. 4
10. Canetti, R., Kilian, J., Petrank, E., Rosen, A.: Black-box concurrent zero-knowledge requires omega (log n) rounds. In: 33rd Annual ACM Symposium on Theory of Computing, pp. 570–579. ACM Press, July 2001. 1.4
11. Chaum, D.: Blind signature system. In: Advances in Cryptology - CRYPTO 1983, p. 153. Plenum Press, New York (1983). 1.4, 5
12. Cramer, R., Damgård, I.B.: Secure distributed linear algebra in a constant number of rounds. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, p. 119. Springer, Heidelberg (2001). 1.4
13. Damgård, I., Pastro, V., Smart, N., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 643–662. Springer, Heidelberg (2012). 1.4
14. Döttling, N., Schröder, D.: Efficient pseudorandom functions via on-the-fly adaptation. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015. LNCS, vol. 9215, pp. 329–350. Springer, Heidelberg (2015). 4, 5
15. Feige, U., Shamir, A.: Zero knowledge proofs of knowledge in two rounds. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 526–544. Springer, Heidelberg (1990). 1.4

16. Fischlin, M., Schröder, D.: On the impossibility of three-move blind signature schemes. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 197–215. Springer, Heidelberg (2010). 1.1, 1.4, 5, 5
17. Freedman, M.J., Ishai, Y., Pinkas, B., Reingold, O.: Keyword search and oblivious pseudorandom functions. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 303–324. Springer, Heidelberg (2005). 1.4, 4
18. Garg, S., Rao, V., Sahai, A., Schröder, D., Unruh, D.: Round optimal blind signatures. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 630–648. Springer, Heidelberg (2011). 1.4, 5
19. Garg, S., Gupta, D.: Efficient round optimal blind signatures. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 477–495. Springer, Heidelberg (2014). 1.4, 5
20. Garg, S., Rao, V., Sahai, A., Schröder, D., Unruh, D.: Round optimal blind signatures. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 630–648. Springer, Heidelberg (2011). 1.4, 5
21. Gennaro, R., Ishai, Y., Kushilevitz, E., Rabin, T.: The round complexity of verifiable secret sharing and secure multicast. In: 33rd Annual ACM Symposium on Theory of Computing, pp. 580–589. ACM Press, July 2001. 1.4
22. Gennaro, R., Ishai, Y., Kushilevitz, E., Rabin, T.: On 2-round secure multiparty computation. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, p. 178. Springer, Heidelberg (2002). 1.4
23. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M., (ed.) 41st Annual ACM Symposium on Theory of Computing, pp. 169–178. ACM Press, May/June 2009. 1.2, 3.1, 3.1
24. Goldreich, O.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, New York (2004). 2.3
25. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions (extended abstract). In: 25th Annual Symposium on Foundations of Computer Science, pp. 464–479. IEEE Computer Society Press, October 1984. 4
26. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. J. Cryptology **9**(3), 167–190 (1996). 1.4
27. Goldreich, O., Krawczyk, H.: On the composition of zero-knowledge proof systems. SIAM J. Comput. **25**(1), 169–192 (1996). 1.4
28. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th Annual ACM Symposium on Theory of Computing, pp. 218–229. ACM Press, May 1987. 1.4
29. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. J. Cryptol. **7**(1), 1–32 (1994). 1, 1.1, 1.4, 2.3
30. Halevi, S., Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. J. Cryptol **25**(1), 158–193 (2012). 3.1, 1, 5
31. Hohenberger, S., Waters, B.: Short and stateless signatures from the RSA assumption. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 654–670. Springer, Heidelberg (2009). 5
32. Ishai, Y., Kushilevitz, E.: Randomizing polynomials: a new representation with applications to round-efficient secure computation. In: 41st Annual Symposium on Foundations of Computer Science, pp. 294–304. IEEE Computer Society Press, November 2000. 1.4
33. Ishai, Y., Paskin, A.: Evaluating branching programs on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 575–594. Springer, Heidelberg (2007). 3.1, 1, 5

34. Jarecki, S., Liu, X.: Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 577–594. Springer, Heidelberg (2009). 1.4, 4
35. Juels, A., Luby, M., Ostrovsky, R.: Security of blind digital signatures. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 150–164. Springer, Heidelberg (1997). 5
36. Katz, J., Ostrovsky, R.: Round-optimal secure two-party computation. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 335–354. Springer, Heidelberg (2004). 1.1, 1.4
37. Katz, J., Ostrovsky, R., Smith, A.: Round efficiency of multi-party computation with a dishonest majority. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 578–595. Springer, Heidelberg (2003). 1.4
38. Kilian, J.: Founding cryptography on oblivious transfer. In: 20th Annual ACM Symposium on Theory of Computing, pp. 20–31. ACM Press, May 1988. 3.1, 1, 5
39. Kilian, J., Petrank, E.: Concurrent and resettable zero-knowledge in poly-loalgorithm rounds. In: 33rd Annual ACM Symposium on Theory of Computing, pp. 560–569. ACM Press, July 2001. 1.4
40. Krawczyk, H., Rabin, T.: Chameleon signatures. In: ISOC Network and Distributed System Security Symposium - NDSS 2000. The Internet Society, February 2000. 5
41. Lindell, Y.: Parallel coin-tossing and constant-round secure two-party computation. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, p. 171. Springer, Heidelberg (2001). 1.4
42. Lindell, Y.: Bounded-concurrent secure two-party computation without setup assumptions. In: 35th Annual ACM Symposium on Theory of Computing, pp. 683–692. ACM Press, June 2003. 1.4
43. Lindell, Y.: Lower bounds for concurrent self composition. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 203–222. Springer, Heidelberg (2004). 1.4
44. Lindell, Y., Pinkas, B.: Secure two-party computation via cut-and-choose oblivious transfer. J. Cryptology **25**(4), 680–722 (2012). 1.4
45. Naor, M.: On cryptographic assumptions and challenges. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (2003). 2.2
46. Naor, M., Pinkas, B.: Oblivious transfer and polynomial evaluation. In: 31st Annual ACM Symposium on Theory of Computing, pp. 245–254. ACM Press, May 1999. 1.4
47. Naor, M., Pinkas, B.: Oblivious transfer with adaptive queries. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, p. 573. Springer, Heidelberg (1999). 1.4
48. Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: Kosaraju, S.R. (ed.) 12th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 448–457. ACM-SIAM, January 2001. 3.1, 1, 5
49. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. In: 38th Annual Symposium on Foundations of Computer Science, pp. 458–467. IEEE Computer Society Press, October 1997. 1, 1.4, 4, 5
50. Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: the non-committing encryption case. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, p. 111. Springer, Heidelberg (2002). 4
51. Nielsen, J.B., Nordholt, P.S., Orlandi, C., Burra, S.S.: A new approach to practical active-secure two-party computation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 681–700. Springer, Heidelberg (2012). 1.4
52. Ostrovsky, R., Paskin-Cherniavsky, A., Paskin-Cherniavsky, B.: Maliciously circuit-private FHE. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 536–553. Springer, Heidelberg (2014). 3.1, 2

53. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. J. Cryptol. **13**(3), 361–396 (2000). 5
54. Prabhakaran, M., Rosen, A., Sahai, A.: Concurrent zero knowledge with logarithmic round-complexity. In: 43rd Annual Symposium on Foundations of Computer Science, pp. 366–375. IEEE Computer Society Press, November 2002. 1.4
55. Reingold, O., Trevisan, L., Vadhan, S.P.: Notions of reducibility between cryptographic primitives. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 1–20. Springer, Heidelberg (2004). 2.2
56. Richardson, R., Kilian, J.: On the concurrent composition of zero-knowledge proofs. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, p. 415. Springer, Heidelberg (1999). 1.4
57. Wegman, M.N., Carter, L.: New classes and applications of hash functions. In: 20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29–31 October 1979, pp. 175–182 (1979). 4
58. Yao, A.C.C.: Protocols for secure computations (extended abstract). In: 23rd Annual Symposium on Foundations of Computer Science, pp. 160–164. IEEE Computer Society Press, November 1982. 1.4