# On the Impossibility of Constructing Non-Interactive Statistically-Secret Protocols from any Trapdoor One-Way Function

Marc Fischlin

Johann Wolfgang Goethe-University,
Frankfurt am Main, Germany

e-mail: marc @ mi.informatik.uni-frankfurt.de
URL: http://www.mi.informatik.uni-frankfurt.de/

**Abstract.** We show that non-interactive statistically-secret bit commitment cannot be constructed from arbitrary black-box one-to-one trapdoor functions and thus from general public-key cryptosystems. Reducing the problems of non-interactive crypto-computing, rerandomizable encryption, and non-interactive statistically-sender-private oblivious transfer and low-communication private information retrieval to such commitment schemes, it follows that these primitives are neither constructible from one-to-one trapdoor functions and public-key encryption in general. Furthermore, our separation sheds some light on statistical zero-knowledge proofs. There is an oracle relative to which one-to-one trapdoor functions and one-way permutations exist, while the class of promise problems with statistical zero-knowledge proofs collapses in $\mathcal{P}$. This indicates that nontrivial problems with statistical zero-knowledge proofs require more than (trapdoor) one-wayness.

## 1 Introduction

One of the fundamental questions in cryptography deals with the relationship of cryptographic primitives: does the existence of primitive $\mathcal{A}$ imply the existence of primitive $\mathcal{B}$? As for positive results, such proofs usually give rise to an explicit construction of primitive $\mathcal{B}$ given an arbitrary instance of primitive $\mathcal{A}$. For instance, given any one-way function we can effectively specify a secure signature scheme [43, 51]. We also know that one-way functions, pseudorandom generators, pseudorandom functions, private-key encryption, signature schemes and computationally-secret bit commitment are all equivalent in this sense [29, 33, 43, 51, 40, 32]. Similarly, trapdoor permutations are sufficient for oblivious transfer and public-key encryption, and for key agreement [19, 30, 22].

Concerning separations of primitives, Impagliazzo and Rudich [34] have shown that basing key agreement (and thus trapdoor permutations and oblivious transfer) on any "black-box" one-way permutation is at least as hard as proving $\mathcal{P} \neq \mathcal{NP}$. The terminology "black-box" refers to the fact that nothing beyond the structure of a primitive is assumed except for fundamental properties guaranteed by the definition. For instance, in [34] the abstract of a one-way permutation

An extended abstract appears in: RSA Security 2002 Cryptographer's Track,
Lecture Notes in Computer Science, Vol.2271, pp.79–95, Springer-Verlag, 2002.

is a one-way permutation oracle, and the efficient evaluation algorithm of the one-way permutation corresponds to single oracle step that returns the function value. Since reductions where the starting primitive is treated as a black box are common throughout complexity-based cryptography, the result of Impagliazzo and Rudich suggests that showing the equivalence of key agreement and one-way functions is infeasible. Therefore, in a sense, secure key agreement needs more than one-wayness.

Although most known reductions obey the black-box approach, there is at least one example of a reduction which is not black-box, i.e., requires that the description of the evaluation algorithm is explicit. See [34] for a discussion. Thus, oracle-based separations do not completely rule out the possibility that reductions exist. There might still be effective constructions which are not black-box. Yet, as mentioned before, the black-box design is widely used in complexity-based cryptography.

For more impossibility results we refer the reader to [52, 58, 37, 36, 24, 25]. In particular, the work by Simon [58] separates collision-intractability and one-wayness by defining an oracle relative to which one-way permutations exist, but collision-intractable hash functions do not. Here, relying on the techniques developed in [58], we extend Simon's result. In the first step we present an oracle relative to which non-interactive statistically-secret bit commitment is impossible, yet one-way permutations exists (throughout the paper we refer to non-interactive protocols as schemes where both parties consecutively send a single message only). Relative to our oracle a very weak form of non-interactive statistically-secret bit commitment does not exist. That is, secrecy is only guaranteed with respect to honestly behaving receivers, and a commitment merely binds with very small, yet noticeable probability.

We stress that it is not known whether any kind of bit commitment yields collision-intractable hash functions in general or not. Thus, our extension from collision-intractable hash functions to commitments is not known to be implied by Simon's result directly. We remark that, conversely, collision-intractable hash functions are sufficient for non-interactive statistically-secret commitment [43, 15, 31]. Furthermore, one can construct perfectly-secret bit commitment from any one-way permutation with linear many rounds in the security parameter [41]. To best of our knowledge, nothing has been reported about improvements concerning either the assumption or the round complexity.[1] Our result provides some evidence that accomplishing non-interactive statistically-secret commitment based on one-wayness alone is impossible. In contrast, non-interactive computationally-secret commitment can be based on one-way functions [32, 40].

In addition to showing that non-interactive statistically-secret commitments are impossible in the presence of general one-way functions, in the second extension step we prove that this impossibility result transfers to the case that one adds the "power" of trapdoors to the one-way function. Such (one-to-one) trap-

[1] We always refer to the classical Turing machine model in this work. In the Quantum computing model there are indeed results that one-wayness is sufficient for constant-round statistically-secret commitments [20, 14].

door functions are a relaxed version of trapdoor permutations. We only demand that the former are one-to-one in order to support unique inversion.

Bellare et al. [6] prove that many-to-one trapdoor functions with super-polynomial preimage size can be derived from any one-way function. Trapdoor functions with polynomially bounded preimage size, among which are one-to-one trapdoor functions, yield public-key cryptosystems, though. In light of [34] they cannot be derived from one-way functions in general, and therefore, in a sense, our result is a strict extension of Simon's separation.

In summery, we broaden Simon's separation in both directions. On one side, we show that relative to an oracle non-interactive weakly-binding honest-receiver statistically-secret commitments schemes do not exist. Such commitment schemes include collision-intractable hash functions, but are not known to imply the existence of such hash functions. On the other side, the negative result holds in the presence of one-way permutation *and* of one-to-one trapdoor functions. The latter functions are presumably not derivable from general one-way permutations.

The relationship of statistical secrecy and one-wayness enables us to obtain a new result about the class $\mathcal{SZK}$ of promise problems with statistical zero-knowledge proofs. We prove that relative to a one-way permutation oracle and to a one-to-one trapdoor function oracle, respectively, the class $\mathcal{SZK}$ breaks down to $\mathcal{P}$. In contrast to our impossibility result, one-way functions suffice to lift the class $\mathcal{CZK}$ of promise problems with computational zero-knowledge proofs to $\mathcal{IP} = \mathcal{PSPACE}$ [56, 35, 8]. This gives us another, oracle-based separation of $\mathcal{CZK}$ and $\mathcal{SZK}$ in addition to the one implied by the presumably strictness of the polynomial hierarchy: $\mathcal{SZK}$ belongs to $\mathcal{AM}\cap$co-$\mathcal{AM}$ [23, 1], and likewise lies much lower in the polynomial hierarchy than $\mathcal{CZK}$ (which equals $\mathcal{PSPACE}$ under the assumption that one-way functions exist). From a cryptographer's point of view, our result says that while one-wayness is sufficient and necessary [47] for nontrivial problems in $\mathcal{CZK}$, hard problems in $\mathcal{SZK}$ seem to require more than general one-way permutations and one-to-one trapdoor functions.

Finally, we consider implications to other cryptographic protocols. By constructing non-interactive weakly-binding honest-receiver statistically-secret commitment schemes from other non-interactive statistically-secret cryptographic protocols, we conclude that such protocols cannot be derived from general black-box one-to-one trapdoor functions. Specifically, we prove that this holds for non-interactive crypto-computing, rerandomizable encryption, and non-interactive statistically-sender-private protocols for oblivious transfer and, using a result of Beimel et al. [5], for private information retrieval with low communication complexity.

The paper is organized as follows. We start with basic definitions in Section 2. Then, in Section 3 we introduce the class $\mathcal{SZK}$ of statistical zero-knowledge proofs for motivating our oracle separation constructions in Section 4. In Section 5 we then apply the separation to $\mathcal{SZK}$, and we discuss implications to other cryptographic protocols in the final part.

## 2 Definitions

We occasionally view probabilistic algorithms as deterministic ones by providing the random coins explicitly. That is, let $A$ be a deterministic algorithm taking two inputs $x$ and $r$. Then we denote by $A(x, r)$ the output of $A$ for input $x, r$ and by $A(x)$ the random variable that describes the output for fixed $x$ and uniformly chosen $r$. It will be clear from the context which part of the input is considered as the random coins. Additionally, we denote by $[A(x)]$ the support of $A(x)$, i.e., $a \in [A(x)]$ if and only if there exists $r$ with $a = A(x, r)$. When passing a function as argument to, say, an oracle, it is understood that we pass a circuit description of the function.

A function $\delta(n)$ is *negligible* if it is eventually less than any polynomial fraction, i.e., $\delta(n) < 1/p(n)$ for any positive polynomial $p(n)$ and all sufficiently large $n$'s. A function $\delta(n)$ is *noticeable* if it is not negligible; it is *overwhelming* if $1-\delta(n)$ is negligible. Two sequences $X = (X_n)_{n\in\mathbb{N}}$ and $Y = (Y_n)_{n\in\mathbb{N}}$ of random variables are *computationally indistinguishable*, $X \overset{c}{\approx} Y$, if for any probabilistic polynomial-time algorithm $D$ the advantage

$$|\text{Prob}\,[D(1^n, X_n) = 1] - \text{Prob}\,[D(1^n, Y_n) = 1]|$$

is negligible.[2] The sequences arecalled *statistically close*, $X \overset{s}{=} Y$, if the statistical difference

$$\text{StatDiff}(X_n, Y_n) = \tfrac{1}{2} \cdot \sum_{s\in[X_n]\cup[Y_n]} |\text{Prob}\,[X_n = s] - \text{Prob}\,[Y_n = s]|$$

is negligible.

### 2.1 Commitment Schemes

A commitment scheme consists of two phases. In the commitment phase the sender puts a secret bit $b$ into a box and sends the locked box to the receiver. In the decommitment phase the sender assists in opening the box, say, by transmitting the key. Then, on one hand, even a malicious sender $\mathcal{S}^*$ cannot change his mind once the box has been given to the receiver (binding property). The receiver, on the other hand, does not learn anything about the bit $b$ till the decommitment step is carried out (secrecy).

We exclusively present the definition of non-interactive honest-receiver statistically-secret bit commitment schemes. Our definition captures only a very weak binding property, namely, that there is no collision-finder that nearly always succeeds in finding ambiguous decommitments. Usually, the binding property demands that any collision-finder fails with very high probability.

**Definition 1.** *The tuple* (Gen, Com, Decom, Vf) *of probabilistic polynomial-time algorithms is a non-interactive weakly-binding honest-receiver statistically-secret bit commitment scheme if*

[2] In this paper we adopt the uniform model for distinguishers. Unless stated otherwise, all consequences remain valid for nonuniform algorithms.

- *generation: on input* $1^n$ *algorithm* Gen *outputs a description* $k_n$ *(wlog.* $k_n$ *contains* $1^n$*).*
- *meaningfulness: for every* $k_n \in [\mathrm{Gen}(1^n)]$ *and every* $c = \mathrm{Com}_{k_n}(b,r)$ *and* $d = \mathrm{Decom}_{k_n}(b,r)$ *we have* $\mathrm{Vf}_{k_n}(b,c,d) = 1$.
- *honest-receiver statistical secrecy: for every sequence* $(k_n)_{n\in\mathbb{N}}$ *with* $k_n \in [\mathrm{Gen}(1^n)]$ *we have* $\mathrm{Com}_{k_n}(0) \overset{s}{=} \mathrm{Com}_{k_n}(1)$.
- *weakly binding: for any probabilistic polynomial-time algorithm* $\mathcal{S}^*$ *the probability that* $\mathcal{S}^*$ *on input* $k_n$ *outputs* $c$ *and* $d, d'$ *such that* $\mathrm{Vf}_{k_n}(0,c,d) = \mathrm{Vf}_{k_n}(1,c,d') = 1$ *is not overwhelming (where the probability is taken over the choice of* $k_n$ *and the coin tosses of* $\mathcal{S}^*$*).*

Instead of using the notation above, we sometimes adopt the viewpoint of a protocol between a sender $\mathcal{S}$ and a receiver $\mathcal{R}$, in which the honest $\mathcal{R}$ transmits $k_n$ obtained by running $\mathrm{Gen}(1^n)$ and $\mathcal{S}$ (with input $b$) answers with a sample $\mathrm{Com}_{k_n}(b,r)$ by choosing $r$ at random. Later, in the decommitment phase, the receiver then applies the verification algorithm $\mathrm{Vf}_{k_n}$ to check the validity of the decommitment $d = \mathrm{Decom}_{k_n}(b,r)$. Note that, in order to generate the decommitment $d$, algorithm Decom gets the same random string $r$ as Com.

Wlog. assume that the input length of the commitment function $\mathrm{Com}_{k_n}$ is at least as large as the security parameter $n$. Also, let the output size of $\mathrm{Com}_{k_n}(b)$ be at most the size of the randomness portion of the input. This can always be achieved by padding the input with redundant random bits. Then the domain of $\mathrm{Com}_{k_n}$ is at least twice as large as its range.

The binding property can be restated as follows. Any adverserial sender $\mathcal{S}^*$ has success probability less than $1 - 1/p(n)$ of coming up with an ambiguous decommitment for some polynomial $p(n)$ and infinitely many $n \in \mathbb{N}$. Basically, this means that in order to refute the weak binding property, $\mathcal{S}^*$ must be able to reveal distinct openings for almost any instance.

### 2.2 Black-Box (Trapdoor) One-Way Functions

We rigidly formalize the concept of black-box (trapdoor) one-way functions as discussed in the introduction.

**Definition 2.** *A black-box one-way function is an oracle* $\mathcal{F} : \{0,1\}^* \to \{0,1\}^*$ *such that for any uniform polynomial-size circuit family* $C = (C_n)_{n\in\mathbb{N}}$ *the inversion probability*

$$\mathrm{Prob}\left[C_n^{\mathcal{F}}(\mathcal{F}(x)) \in \mathcal{F}^{-1}(\mathcal{F}(x))\right]$$

*is negligible, where the probability is taken over the random choice of* $x \in_R \{0,1\}^n$ *and the internal coin tosses of* $C_n$*. If additionally* $\mathcal{F}(\{0,1\}^n) = \{0,1\}^n$ *for all* $n \in \mathbb{N}$ *then we say that* $\mathcal{F}$ *is a black-box one-way permutation.*

We remark that $C_n$ is granted oracle access to $\mathcal{F}$. This enables $C_n$ to evaluate $\mathcal{F}$ at values of its choice. Also, we demand that the family $C$ of circuits is uniform. This is necessary to derandomize the probabilistic oracle construction as described in [57, 58].

Our definition imitates the one of a one-way function with infinite domain. Instead, one sometimes uses collections of one-way function, where each function of this collection is indexed. Yet, we omit further discussions since the notion of a collection of black-box one-way functions is implicit in the definition of trapdoor one-way functions below, and can be easily inferred. From an existential point of view both notions of one-way functions are equivalent, even in the black-box case.

**Definition 3.** *A black-box one-to-one trapdoor function is an oracle $\mathcal{T}$ with three query states* $\mathsf{generate}, \mathsf{evaluate}, \mathsf{invert}$*:*

- *generation: $\mathcal{T}(\mathsf{generate}, \omega)$ for $\omega \in \{0,1\}^n$ outputs a pair $(t, i)$. Wlog. let $1^n$ be recoverable from index $i$ and trapdoor $t$. Furthermore, assume that $i$ uniquely determines $t$ and vice versa.*
- *evaluation: given $x \in \{0,1\}^n$ and an index $i$ with $(t,i) = \mathcal{T}(\mathsf{generate}, \omega)$ for some $\omega \in \{0,1\}^n$, the oracle $\mathcal{T}(\mathsf{evaluate}, i, x)$ returns $y \in \{0,1\}^{poly(n)}$. Also, let $\mathcal{T}(\mathsf{evaluate}, i, \cdot)$ be one-to-one for any index $i$.*
- *inversion: given $y \in \{0,1\}^{poly(n)}$ and $t$, the answer $\mathcal{T}(\mathsf{invert}, t, y)$ is some $x$ such that $\mathcal{T}(\mathsf{evaluate}, i, x) = y$ if such an $x$ exists (where $i$ is the uniquely determined index to $t$), and an undefined symbol otherwise.*

*Additionally, $\mathcal{T}$ satisfies the following one-wayness property: for any uniform polynomial-size circuit family $C = (C_n)_{n \in \mathbb{N}}$ the inversion probability*

$$\text{Prob}\left[C_n^{\mathcal{T}}(i, \mathcal{T}(\mathsf{evaluate}, i, x)) = x\right]$$

*is negligible, where the probability is taken over the choice of $i$ according to $\mathcal{T}(\mathsf{generate}, \omega)$ for a random $\omega \in_R \{0,1\}^n$, over $x \in_R \{0,1\}^n$, and over the randomness of $C_n$.*

The generation step of our definition says that one must externally supply the deterministic oracle $\mathcal{T}$ with randomness $\omega$ to get a random function description $(t, i)$. For simplicity and since our construction achieves this, we presume that $\mathcal{T}$ only takes $n$ random bits to produce a random description of complexity $n$. Additionally, we demand a bijective relationship of trapdoors and indices. More generally, we could allow several matching indices $i, i'$ to a single trapdoor $t$. Again, as our construction supports this uniqueness property, we do not include this in our definition.

A difference between the notion of a black-box trapdoor function $\mathcal{T}$ and the one of a black-box one-way function of Definition 2 is that $\mathcal{T}$ combines three oracles for $\mathsf{generate}, \mathsf{evaluate}, \mathsf{invert}$. For a black-box one-way function $\mathcal{F}$ with infinite domain only evaluation is necessary, i.e., any oracle query to $\mathcal{F}$ is an evaluation request. A comparison of our definition of one-to-one trapdoor functions and trapdoor permutations follows the actual construction of a black-box trapdoor function in Section 4.2.

## 3 Statistical Zero-Knowledge

When talking about complexity classes, we always refer to classes of *promise problems.* A promise problem $\Pi = (\Pi_{\text{YES}}, \Pi_{\text{NO}})$ is a pair of disjoint sets of yes-instances $\Pi_{\text{YES}} \subseteq \{0,1\}^*$ and no-instances $\Pi_{\text{NO}} \subseteq \{0,1\}^*$. The notion of promise problems generalizes the language-based approach: an algorithm putatively deciding membership for some input $x \in \{0,1\}^*$ gets a promise that $x \in \Pi_{\text{YES}} \cup \Pi_{\text{NO}}$.

We briefly introduce the zero-knowledge-based classes we deal with. As we do not use any definitional properties beyond some basic facts about their relationships, we omit formal definitions of these classes and refer the reader to [59] for details. The class $\mathcal{NISZK}$ consists of the promise problems having a non-interactive statistical zero-knowledge proof. The class $\mathcal{SZK}$ is the class of problems having general, possibly interactive statistical zero-knowledge proofs; this is clearly a subset of the class of problems where statistical zero-knowledge holds with respect to honest verifiers, denoted by $\mathcal{HVSZK}$. By [18, 28, 27] we have $\mathcal{P} \subseteq \mathcal{BPP} \subseteq \mathcal{NISZK} \subseteq \mathcal{SZK} = \mathcal{HVSZK}$.

Sahai and Vadhan [53] introduced the $\mathcal{SZK}$-complete problem statistical difference. Using the completeness of this problem we show the collapse of $\mathcal{SZK}$. To a circuit $X : \{0,1\}^m \to \{0,1\}^n$ (more precisely, to its description) we associate a random variable over $\{0,1\}^n$ by choosing the input uniformly from $\{0,1\}^m$.

**Definition 4.** *The promise problem statistical difference,* $\texttt{SD} = (\texttt{SD}_{\text{YES}}, \texttt{SD}_{\text{NO}})$*, is defined by*

$$\texttt{SD}_{\text{YES}} = \{(X_0, X_1) \mid \text{StatDiff}(X_0, X_1) \geq 2/3\}$$
$$\texttt{SD}_{\text{NO}} = \{(X_0, X_1) \mid \text{StatDiff}(X_0, X_1) \leq 1/3\}$$

To prove that $\texttt{SD}$ is complete for $\mathcal{SZK}$, Sahai and Vadhan [53, 54] established the polarization lemma. Basically, this lemma says that one can turn an instance $(X_0, X_1)$ of $\texttt{SD}$ into a pair $(Y_0, Y_1)$ of circuits such that the distributions of $Y_0, Y_1$ are almost disjoint if $(X_0, X_1) \in \texttt{SD}_{\text{YES}}$ and nearly equal if $(X_0, X_1) \in \texttt{SD}_{\text{NO}}$. Additionally, the transformation involves an error parameter $\ell$ that determines how far and close, respectively, the derived distributions are. This parameter $\ell$ may be independent of $X_0, X_1$.

**Fact 1 (Polarization Lemma [53, 54])** *There is a polynomial-time algorithm* Polarize *that on input* $(X_0, X_1, 1^\ell)$ *outputs* $(Y_0, Y_1)$ *such that*

$$(X_0, X_1) \in \texttt{SD}_{\text{YES}} \quad \Rightarrow \quad \text{StatDiff}(Y_0, Y_1) \geq 1 - 2^{-\ell}$$
$$(X_0, X_1) \in \texttt{SD}_{\text{NO}} \quad \Rightarrow \quad \text{StatDiff}(Y_0, Y_1) \leq 2^{-\ell}$$

*Set* $\text{Polarize}(X_0, X_1) = \text{Polarize}\left(X_0, X_1, 1^{|(X_0, X_1)|}\right)$.

Intuitively, one can think of a polarized pair $(Y_0, Y_1)$ as a description of a non-interactive commitment function. The sender splits the bit $b$ into $n$ random pieces $b_1, \ldots, b_n$ such that $b = b_1 \oplus \cdots \oplus b_n$. Given $n$ random instances $(Y_{i,0}, Y_{i,1})$ —among which there will be a no-instance with high probability— the sender commits to each piece $b_i$ individually by sampling according to $Y_{i,b_i}$ and handing this sample to the receiver. If $(Y_{i,0}, Y_{i,1})$ is a (polarized) no-instance then the distribution hides $b_i$ and therefore $b$ statistically; if it corresponds to a yes-instance then the sample determines $b_i$ with very high probability (as long as the sender does not bias the sample too much).

For an ambiguous decommitment to $b' = b \oplus 1$ the sender has to flip at least one piece $b_i$. Put differently, the sender has to find a random string $r'$ such that $Y_{i,b_i \oplus 1}$ maps this string to the previously given sample $Y_{1,b_i}(r)$. But if $(Y_{i,0}, Y_{i,1})$ is a yes-instance then the distributions are almost disjoint and this is quasi impossible. On the other hand, for a no-instance this is indeed possible. Hence, an ambiguous decommitment tells us the status of (at least) one of the instances. This is basically the reason why deciding membership for SD becomes tractable relative to our oracle: ambiguous decommitments can be found easily given access to the oracle. However, we remark that for a correct membership decision on *each* instance of SD, the oracle must never err. This motivates the investigation of weakly-binding commitment schemes, where the oracle must (nearly) always return ambiguous decommiments to disprove their existence. Still, a very small error for the binding property is acceptable to ensure a perfect oracle.

## 4 Extensions of Simon's Result

In this section we apply Simon's result [58] to obtain an oracle separation of black-box one-way permutations and trapdoor one-way functions from non-interactive weakly-binding honest-receiver statistically-secret bit commitment.

### 4.1 Extension to Commitment Schemes

We briefly describe the oracle construction in [58]. One starts with a random oracle $\Pi$ which contains a random permutation $f$ and a special query state collision. Basically, the random permutation $f$ constitutes a one-way function, and the query state collision enables to find collisions in hash functions. Once proven that this random oracle is one-way, one can then derandomize the construction.

**Construction 1** *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a random permutation, i.e., a random function with the constraint $f(\{0,1\}^n) = \{0,1\}^n$ for all $n \in \mathbb{N}$. Define oracle $\Pi$ to contain a random permutation $f$, and a special query state* collision *that takes a circuit description of a many-to-one hash function $h$ and outputs a random element $x$ together with a uniformly chosen value $x'$ from $\{y \mid h(x) = h(y)\}$ (and, besides, repeats the description of the hash function and any oracle queries and answers obtained within the computation of the hash values for $x$ and $x'$).*

In the rest of the paper, we call this way of generating collisions $x, x'$ the *basic sampling procedure.*

In [58] it was shown that the collision-finding portion of $\Pi$ does not help to invert $f$ significantly. Note that the description of the hash function might also include $f$- and recursive $\mathsf{collision}$-queries and that we let $\Pi$ append these queries and answers to the output for $\mathsf{collision}$-questions, too. Using an appropriate encoding for $\mathsf{collision}$-queries, e.g., substituting values in $f$ by mapping inputs of the form $(1\cdots 1, h, \ldots, h)$ to $(1\cdots 1, h, x, x', \text{queries \& answers})$ and vice versa for a sufficient number of 1's and $h$-repetitions, the special query state $\mathsf{collision}$ can be eliminated and it can be achieved that $\Pi$ is also a permutation over $\{0,1\}^*$. See [58] for details. In the sequel, we sometimes switch between both approaches for sake of convenience.

We would like to extend the negative result to non-interactive weakly-binding honest-receiver statistically-secret bit commitment schemes. To this end, we change oracle $\Pi$ to an oracle $\Sigma$ which allows to open such commitment schemes ambiguously and to contradict the weak binding property. That is, $\Sigma$ should return valid decommitments for different bits for statistically-secret commitment schemes for any sufficiently large security parameter. For instance, this can be accomplished by letting the oracle always output a non-trivial collision (i.e., with $b \neq b'$) if the statistical difference is, say, less than some bound $B$, and by reducing one-wayness of this oracle to the one-wayness of $\Pi$ by querying $\Pi$ a sufficient number of times in order to simulate the new oracle. However, for this we have to ensure that the oracle's answers to queries with statistical difference more than $B$ are answered consistently compared to the simulation. We will overcome this problem by letting our new oracle $\Sigma$ generate random ambiguous decommitments in a way that already mimics the simulator's behavior asking several questions to $\Pi$:

**Construction 2** *Let $\Pi$ be defined as in Construction 1. Modify $\Pi$ by replacing the $\mathsf{collision}$-query state as follows: if the probability that the basic sampling procedure outputs a random collision $(b,r),(b',r')$ with $b \neq b'$ for the many-to-one function $\mathrm{Com}_{k_m}$ is at least $1/6$, then uniformly select some $(b,r)$ from the set of pairs $(c,s)$ for which $C_{c,s} = \{(c\oplus 1, s') \mid \mathrm{Com}_{k_m}(c,s) = \mathrm{Com}_{k_m}(c\oplus 1, s')\}$ is not empty, together with a uniformly chosen value $(b',r')$ from $C_{b,r}$. Else, for input length $\ell$ of $\mathrm{Com}_{k_m}$, generate $\ell$ random collisions $(b,r),(b',r')$ with the basic sampling procedure; if there is some collision with $b \neq b'$ then output the first one that appears among these samples, otherwise return the first of the $\ell$ samples (which is then of the form $(b,r),(b,r')$, of course). Furthermore, the oracle appends the description of $\mathrm{Com}_{k_m}$ and all oracle queries to compute $\mathrm{Com}_{k_m}(b,r)$ and $\mathrm{Com}_{k_m}(b',r')$. Denote this oracle by $\Sigma$.*

We claim that setting the bound to $1/6$ guarantees that oracle $\Sigma$ always returns ambiguous decommitments for statistically-secret bit commitments (for sufficiently large security parameter). To see this, let $\mathrm{Com}_{k_m}$ be a statistically-secret bit commitment function. Call an input $(b,r)$ good if the number of random strings $s$ that map to the same commitment $\mathrm{Com}_{k_m}(b,r) = \mathrm{Com}_{k_m}(b,s)$ is at most twice the number of random strings $s'$ which map to the same commitment $\mathrm{Com}_{k_m}(b,r) = \mathrm{Com}_{k_m}(b\oplus 1, s')$ for the inverse bit $b\oplus 1$. With probability at least $1/2$ a random value $(b,r)$ is good (otherwise the statistical difference of $\mathrm{Com}_{k_m}(0)$ and $\mathrm{Com}_{k_m}(1)$ would be at least $1/4$ which would contradict the negligible deviation for large security parameters).[3] In this case, for a uniformly chosen colliding input $(b', r')$ to some good $(b, r)$ it holds that $b' \neq b$ with probability at least $1/3$. Hence, for statistically-secret commitment, with probability at least $1/6$ a random collision represents valid decommitments for distinct bits. See [54] for a tighter bound depending on the actual statistical difference.

Note that oracle $\Sigma$ can still be applied to find collisions for hash functions; but $\Sigma$ even tries to come up with special collisions with distinct leftmost bit in order to find collisions for statistically-secret bit commitment schemes. Also, if we eliminate the $\mathsf{collision}$-state from $\Pi$ by encoding such queries in $\{0,1\}^*$, then $\Sigma$ inherits this property.

**Lemma 1.** *Oracle $\Sigma$ in Construction 2 is a black-box one-way permutation.*

Formally, $\Sigma$ is a *random* oracle. Hence, we would better say "Picking $\Sigma$ as in Construction 2 one obtains a black-box one-way permutation." We neglect this as we will later derandomize the construction anyway.

*Proof.* Clearly, the permutation property is not affected by the modification, even if we encode $\mathsf{collision}$-queries by bit strings. It thus suffices to prove one-wayness. Assume that there exists a (uniform) polynomial-size circuit family $D = (D_n)_{n\in\mathbb{N}}$ that takes advantage of the modification in Construction 2. That is, $D$ is able to invert $\Sigma$ with noticeable probability. Let the polynomial $q(n)$ bound the size of $D$, and let $D_n$ invert a random image under $\Sigma$ with probability at least $1/p(n)$ for a polynomial $p(n)$ and infinitely many $n \in \mathbb{N}$. Note that the total number of oracle queries, including the recursive ones in $\mathsf{collision}$-queries, is bounded above by the size $q(n)$ of $D_n$. From $D$ we construct a polynomial-size circuit family $C = (C_n)_{n\in\mathbb{N}}$ interacting with a random oracle $\Pi$ according to Construction 1.

Basically, $C_n$ gets an image $y$ as input and simulates $D_n$ on $y$. Each $f$-query of $D_n$ is answered by asking the $f$-oracle of $\Sigma$. Every time $D_n$ submits a $\mathsf{collision}$-query for some $\mathrm{Com}_{k_m}$ with input length $\ell$, then circuit $C_n$ essentially (details below) asks $\Pi$ altogether $\ell$ $\mathsf{collision}$-queries by padding the description of $\mathrm{Com}_{k_m}$ with redundant bits in each query (after all, $\Pi$ is a random *function* and always returns the same answer to the same question again; padding the commitment function description thus yields independent random collisions). Then $C_n$ selects an adequate collision and hands it to $D_n$.

We explain in detail how $C_n$ finds an appropriate collision. Assume for the moment that $C_n$ picks a collision by querying $\Pi$ as described above $\ell$ times. If the commitment function $\mathrm{Com}_{k_m}$ is above the limit $1/6$, then circuit $C_n$ would find a proper collision with probability at least $1-(5/6)^\ell$; if $\ell$ is large this is very close to 1 and almost identical to the answer of $\Sigma$. For $\mathrm{Com}_{k_m}$-functions below the bound $1/6$, circuit $C_n$ would give identically distributed answers to $\mathsf{collision}$-queries in comparison to $\Sigma$. A problem occurs if $\ell$ is too small. Then

[3] By assumption the input length of commitment functions is at least as large as the security parameter. This allows us to use the same bound $1/6$ when considering the input length as replacement for the security parameter, as done in Construction 2.

$C_n$'s output would differ noticeably from $\Sigma$'s answer for commitment function above the bound $1/6$. To overcome this, we let $C_n$ search for the right answers for commitment functions $\mathrm{Com}_{k_m}$ with small input length. Then the simulation error will still leave enough mass for $C_n$'s success probability. We use the limit $L(n) = 4\log_2(2p(n)q(n)) \geq \log_{6/5}(2p(n)q(n))$ to identify a small input length. That is,

- if the input length of $\mathrm{Com}_{k_m}$ is bounded above by $L(n) = 4\log_2(2p(n)q(n))$ then $C_n$ verifies the bound of $1/6$ by computing and counting all possible commitments for $\mathrm{Com}_{k_m}$. These are at most $32p(n)q(n)$ many values, and each can be computed in polynomial-time. Thus, the overall complexity remains polynomially bounded. If the commitment function exceeds the bound of $1/6$ then $C_n$ samples $n$ random collisions by querying $\Pi$. With probability at least $1 - (5/6)^n$ we will then find an ambiguous decommitment among these samples. If the probability that a random collision for $\mathrm{Com}_{k_m}$ yields distinct leftmost bits $b \neq b'$ is below $1/6$ then proceed as $\Sigma$ by picking $\ell$ samples and returning a corresponding collision $(b, r), (b', r')$.
- if the input length is larger than $L(n)$ then $C_n$ generates $\ell$ random collisions, and outputs the first one with $b \neq b'$, or if no such exists, simply returns the first collision.

Recall that the description of many-to-one functions in collision-queries may also include recursive collision-request. We assert that the same solution as before applies. Either the input length is "very short", or using enough samples yields a sufficiently good approximation. More formally, we can first modify the commitment function description by adding an "if-then-else"-check for recursive collision-queries. This check simply imitates $C_n$'s strategy, i.e., compares the input length to $L(n)$ and proceeds accordingly.

What is the error of this simulation? Consider a single collision-query, either one on top level or a recursive one. The only cases where $C_n$'s answer differs from $\Sigma$'s reply are if the input length is at most $L(n)$ and $C_n$ does not find an appropriate collision among the $n$ samples, or if the input length is larger than $L(n)$ and the commitment function exceeds the limit of $1/6$ but $C_n$ returns a collision with $b = b'$. The error probability of the first case is at most $(5/6)^n$ which eventually becomes less than $1/(2p(n)q(n))$. The likelihood of the latter case is at most $(5/6)^\ell \leq 1/(2p(n)q(n))$ for all $n$'s. Since $D_n$ puts at most $q(n)$ oracle queries, by the union bound the simulation therefore fails with probability at most $1/(2p(n))$ for any sufficiently large $n$. Thus, at most half of the cases of $D_n$'s success are covered by the simulation error, and $C_n$ successfully inverts $\Pi$ with probability at least $1/(2p(n))$ infinitely often. But this contradicts the result in [58]. □

Derandomizing the oracle construction by taking an appropriate oracle which works for all of the countable many uniform circuits [57, 58], we obtain a one-way permutation oracle relative to which there cannot exist non-interactive weakly-binding honest-receiver statistically-secret bit commitment schemes, even such ones that use oracle queries.

**Theorem 1.** *Relative to an oracle there exist black-box one-way functions and permutations, but no non-interactive weakly-binding honest-receiver statistically-secret bit commitment schemes.*

### 4.2 Extension to Black-Box Trapdoor Functions

The essence of our construction of a black-box trapdoor function utilizes the idea of the construction of signature schemes from one-way functions [43, 51]: the public and the private key of the signature scheme are the value of a one-way function and its preimage. Here, the index $i$ of the trapdoor function is the value of $\Sigma$ at the trapdoor $t$. Incorporating $i$ into the evaluation process by setting the function to $\Sigma(i, \cdot)$ gives the desired trapdoor one-way function. To invert some $y$ in the range of $\Sigma(i, \cdot)$ one has to provide the matching trapdoor $t$ to $i$ to the inversion oracle.

**Construction 3** *Let $\Sigma$ (over $\{0,1\}^*$) be as in Construction 2. Define $\mathcal{T}$ as follows:*

- *generation: on input $\omega \in \{0,1\}^n$ oracle $\mathcal{T}$ outputs $t = \omega$ and $i = \Sigma(\omega)$.*
- *evaluation: on input $i, x \in \{0,1\}^n$ the evaluation algorithm of $\mathcal{T}$ returns $\Sigma(i, x) \in \{0,1\}^{2n}$*
- *inversion: given $y \in \{0,1\}^{2n}$ and $t \in \{0,1\}^n$ the oracle $\mathcal{T}$ first checks that $\Sigma(t)$ equals the left half of $(i, x) = \Sigma^{-1}(y)$. If so, it outputs $x$, else some undefined symbol.*

Some remarks are in place. Apparently, our function is one-to-one but not a permutation. Hence, iteration techniques for trapdoor permutations, like feeding the output into the function again, are impossible. Nevertheless, we can apply a tree construction of logarithmic depth by iterating the function on each output half. This may replace the permutation in some settings. Similarly, it may suffice to iterate the function on, say, the left half of the result and output the right half "in clear".

Also, observe how our construction circumvents the problem of claws. A pair of claw-free functions is pair of functions with identical range, but such that finding inputs for each function that both map to the same output is infeasible. Any impossibility result about the construction of non-interactive statistically-secret commitment schemes based on any trapdoor function implies that the trapdoor functions do not yield claw-free functions. In our case, any distinct trapdoor functions $(t, i) \neq (t', i')$ have disjoint ranges (because $\Sigma(i, x) \neq \Sigma(i', x')$ for all $x, x'$ for the permutation $\Sigma$).

The proof that $\mathcal{T}$ is a trapdoor one-way function follows by reduction to the one-wayness of $\Sigma$. While generation and evaluation queries for $\mathcal{T}$ can be easily emulated given access to $\Sigma$, we have to ensure that inversion queries do not lend significant power to an adversary. Indeed, for a given index $i \in \{0,1\}^n$ the range of $\Sigma(i, \cdot)$ forms a sparse subset of $\{0,1\}^{2n}$ of size $2^n$. Therefore, any algorithm that tries to invert an image $y$ by guessing a trapdoor $t'$ and asking $\mathcal{T}$ to invert a large $y$ with respect to $t'$ almost certainly gets the undefined symbol as reply. In other words, inversion queries for large images essentially lead to reasonable answers only if the corresponding image has been computed previously by querying the evaluation oracle of $\mathcal{T}$. But then the preimage is already known and gives no additional information. For short images, a preimage can be computed efficiently by searching the domain, and thus inversion queries do not give any advantage in this case either.

**Lemma 2.** *Oracle $\mathcal{T}$ in Construction 3 is a black-box one-to-one trapdoor one-way function.*

*Proof.* It is easy to see that the oracle satisfies the structural properties of Definition 3. It remains to show that it achieves the one-way property. The proof is by reduction to the one-wayness of the oracle $\Sigma$ of Construction 2, using similar ideas as in the proof of Lemma 1. Given a uniform circuit $D = (D_n)_{n \in \mathbb{N}}$ that inverts $\mathcal{T}$ with noticeable probability, we construct $C = (C_n)_{n \in \mathbb{N}}$ that inverts $\Sigma$ with noticeable probability.

Let us first fix some notations. Circuit $D_n$ is given a random input $(i, y)$ and is supposed to return the "preimage" $x$ with $\mathcal{T}(\mathsf{evaluate}, i, x) = y$. Let $t$ be the trapdoor to $i$. We say that $D_n$ *finds the trapdoor* if $D_n$ puts an inversion query $(\mathsf{invert}, t, *)$ in which $t$ appears. The intuition is that if $D_n$ finds the trapdoor this means a total break, because any value in the range of $\Sigma(i, \cdot)$ can then be inverted. We say that $D_n$ *predicts correctly* if $D_n$ puts an inversion request $(\mathsf{invert}, t', y')$ without having obtained $y'$ as reply to an evaluation query, and such that $\Sigma(t') = i'$ for $x'$ with $\mathcal{T}(\mathsf{evaluate}, i', x') = y'$. Informally, if $D_n$ is able to predict correctly with significant success, then it might also be able to find the preimage of $y$. Let $1/p(n)$ denote a lower bound on $D_n$'s success probability (achieved for infinitely many $n$'s) and $q(n)$ an upper bound on the size of $D$, where $p(n), q(n)$ are polynomials.

Next we explain how circuit $C_n$ emulates $D_n$. Obviously, $C_n$ is able to simulate generation and evaluation request by querying oracle $\Sigma$. Additionally, $\mathsf{collision}$-questions can be answered by $C_n$'s oracle, too. We address the trapdoor inversion queries of $D_n$. Our first observation is that $\mathsf{collision}$-queries (appropriately encoded as described before) can be easily inverted, because $\Sigma(\mathsf{collision}, x)$ contains $x$ as part of the output again, and the preimage of $(\mathsf{collision}, x) = \Sigma(\Sigma(\mathsf{collision}, x))$ can be computed by applying $\Sigma$ to $(\mathsf{collision}, x)$. So we only deal with other inversion queries. The idea is that trying to invert values which have not been the response to an evaluation request are useless; either they are too short and a preimage can be computed in polynomial time by exhaustive search, or they are too large and then they will not be in the range of $\Sigma(i', \cdot)$ with significant probability and thus yield the undefined symbol as answer with sufficiently large probability (where $i'$ corresponds to the trapdoor $t'$ in the inversion request). Therefore, $D_n$ merely predicts correctly for short values or with sufficiently small probability.

To formalize the concept above, we let $C_n$ exhaustively search for preimages within a bound of $L(n) = 4 \log_2(4p(n)q(n))$ bits. More specifically, before circuit $C_n$ starts to simulate $D_n$ it first records all image/preimage pairs of $\Sigma$ up to bit size $L(n)$. This can be done in polynomial time in $n$.[4] Now $C_n$ answers $D_n$'s inversion queries as follows. Let $(\mathsf{invert}, t', y')$ denote $D_n$'s request.

- If at some point in the simulation so far, $y'$ has been the answer to a query $(\mathsf{evaluate}, i', x')$, either if $D_n$ has put this question or if it is in the previously recorded list, then output $x'$ if $\Sigma(t') = i'$, and the undefined symbol if $\Sigma(t') \neq i'$;
- else, if $y'$ has not appeared before and the length of $y'$ exceeds $L(n)$, then return the undefined symbol.

Let us discuss that this way of answering any of $D_n$ inversion request is correct except with error probability $1/(4p(n)q(n))$. If $y'$ has been returned before for a query $(\mathsf{evaluate}, i', x')$, but $\Sigma(t') \neq i'$, then $D_n$'s query is invalid (because $y'$ uniquely determines $i', x'$ and thus $t'$); if $\Sigma(t') = i'$ then we return the correct answer. We consider the case that $y'$ has never appeared before. Any query of length less than or equal to $L(n)$ is answered correctly by circuit $C_n$. Returning the undefined symbol for a query with length more than $L(n)$ is right except with the following probability: since $D_n$ has put at most $q(n)$ queries about other images/preimages of equal length at this point, the chance of predicting correctly an unknown value whose preimage contains $i'$ is at most $2^{L(n)/2}/(2^{L(n)} - q(n))$. This, in turn, is less than $2^{-L(n)/4} = 1/(4p(n)q(n))$. Summerizing, the error of the simulation of an $\mathsf{invert}$-query is bounded above by $1/(4q(n)p(n))$.

Again, the description of many-to-one functions in collision-queries may also include generate, evaluate and invert request. The former ones can be simulated with help of the oracle $\Sigma$. As for inversion queries, either the preimage has already appeared in an evaluation request, or is "very short", or the query will result in an undefined answer with sufficiently high probability. That is, we modify the hash function by wiring a list of queries made so far (inclusive the ones up to length $L(n)$) into the description; assimilating an "if-then-else"-check in the description for invert-queries by setting the answer to the undefined symbol and skipping the query if the value does not appear in the query list yields a suitable replacement of $D_n$'s commitment function. Hence, even such queries do not contribute extremly to $D_n$'s success.

By the union bound, the probability that all of the at most $q(n)$ invert-queries are simulated correctly, is at least $1 - 1/(4p(n))$. Therefore, from now on we presume that $D_n$ can be simulated with error at most $1/(4p(n))$. The next observation is that $D_n$ cannot find the trapdoor $t$ with probability more than $1/(2p(n))$ for infinitely many $n$'s. Otherwise $C_n$ could be used to invert the black-box one-way permutation $\Sigma$ with noticeable success probability. Namely, $C_n$ runs $D_n$ on $(i, y)$ for a given value $i$ and by generating $y = \Sigma(i, x)$ for a random $x \in_R \{0,1\}^n$. This simulation succeeds with probability at least $1 - 1/(4p(n))$. Hence, if $D_n$ finds the trapdoor with the asserted probability, then it also finds the trapdoor with probability $1/(4p(n))$ in a successful simulation. But then $C_n$

[4] As opposed to Lemma 1 this time we prefer to compute all small values at the beginning and not on demand. Both solutions are equivalent.

would be able to invert $\Sigma$ on $i$ with noticeable probability, contradicting the one-wayness of $\Sigma$.

Given that $D_n$ does not find the trapdoor and that the simulation works, $D_n$ must be able to invert the one-to-one one-way function $\Sigma(i,\cdot) : \{0,1\}^n \to \{0,1\}^{2n}$ with noticeable probability, or more precisely, with probability $1/(4p(n))$ infinitely often. But it is not hard to see that the impossibility result in [58] also applies to this more general case of a random one-to-one function, i.e., the probability of inverting (a sequence of) random one-to-one functions efficiently is negligible.[5] This gives the desired contradiction. □

Derandomizing Construction 3 we conclude:

**Theorem 2.** *Relative to an oracle there are black-box one-to-one trapdoor functions and black-box one-way functions and permutations, but no non-interactive weakly-binding honest-receiver statistically-secret bit commitment schemes.*

In Appendix A we briefly discuss that we can turn $\mathcal{T}$ into a single oracle that operates on bit strings. This is achieved by using suitable encodings for the query states.

## 5 Nontrivial Statistical Zero-Knowledge Requires More Than Black-Box One-Wayness

In this section we prove the collapse of $\mathcal{SZK}$ relative to an appropriate one-way permutation oracle and to a one-to-one black-box trapdoor function. It is known that hard-to-predict problems in $\mathcal{SZK}$ imply one-way functions [45]. The premise of this implication was later relaxed to $\mathcal{CZK} \neq$ average-case-$\mathcal{BPP}$ [47]. Our result presents some evidence that nontrivial problems in $\mathcal{SZK}$ actually need more than one-wayness. Furthermore, we supplement the result that $\mathcal{SZK} \neq \mathcal{BPP}$

[5] We explain on a very informal level by describing the proof in [58]. The inverter is supposed to find a preimage of $0^n$ under random permutation $f$ (since $f$ is random, $0^n$ is as good as any other image). Consider a permutation oracle $\pi$ derived from oracle $f$ by transposing $0^n$ with a randomly chosen image $y$. Denote this transposition by $\delta$. If a polynomial-size circuit finds the preimage $x = f^{-1}(0^n)$ then we can also deduce $\delta$ from $\pi(x) = y$. Put differently, the chance of finding $\delta$ bounds the probability of finding $x$ from above. If the inverter puts an $f$-query then it is very unlikely that it will receive the answer $0^n$ (this would be called a $\delta$-hit), or that the oracle returns $\pi^{-1}(0^n)$ (called $\pi$-hit; then $f(\pi^{-1}(0^n)) = y$). Hence, when submitting the first collision-query there are still exponentially many possibilities for $\delta$, i.e., the circuit is oblivious about $\delta$ and therefore cannot choose a "clever" hash function. Since the oracle returns uniformly distributed values (not independent, though) the answers will not be hits either, except with very small probability. This implies that the circuit essentially remains oblivious about $\delta$. Inductively, it follows that the inverter cannot find $\delta$ and thus $x$ with significant success. From this informal discussion, one sees that the same argument holds for one-to-one functions: the proof relies on the fact that hits (defined via the domain of the function) almost never occur.

relative to an oracle [2] by showing that $\mathcal{SZK} = \mathcal{BPP} = \mathcal{P}$ relative to a one-way permutation oracle. Note that the existence of (black-box) one-way functions implies that $\mathcal{NP} \not\subseteq \mathcal{BPP}$. This easily follows from an extension of the result that the existence of one-way functions (in a structural sense) is equivalent to $\mathcal{P} \neq \mathcal{NP}$ (see [4]).

The construction of our oracle $\Gamma$, relative to which SD is easy, is a slight modification of $\Sigma$ in Construction 3. In order to preserve the interpretation that one-wayness does not suffice for nontrivial problems in $\mathcal{SZK}$, we allow instances of SD to include query gates for the oracle $\Gamma$. We remark that this extended version of SD is complete for this relativized class of $\mathcal{SZK}$; this shows for example in the completeness proof given in [59]. Hence, if we show the tractability of SD relative to $\Gamma$ it follows that the whole relativized class of $\mathcal{SZK}$ collapses.

In the construction of $\Gamma$ we presume wlog. that the input size of circuits $Y_0$ and $Y_1$ of a polarized instance (for any complexity parameter) is at least the output length, and that both circuit have the same input size. Otherwise algorithm Polarize pads the input length with a minimal number of bits. Then we can view an input $(X_0, X_1)$ as a description of a commitment function $\mathrm{Com}_{(X_0,X_1)}(b,r) = Y_b(r)$, where $Y_0, Y_1$ are derived by applying the polarization lemma (together with the length convention).

**Construction 4** *Let $\Sigma$ be as in Construction 3. Alter $\Sigma$ to $\Gamma$ by modifying the* collision*-state as follows: $\Gamma$ only accepts pairs $(X_0, X_1)$ of circuits as arguments to* collision*-queries. Then $\Gamma$ polarizes $(X_0, X_1)$ with parameter $\ell = |(X_0, X_1)|$ to obtain $(Y_0, Y_1)$. If $(X_0, X_1)$ is a yes-instance for* SD *then return a pair $(b, r), (b, r')$ such that $Y_b(r) = Y_b(r')$ (generated by the basic sampling procedure with the restriction that the second value is chosen uniformly among the collisions with the same leftmost bit $b$); if $(X_0, X_1) \in \mathtt{SD}_{\text{NO}}$ then sample a random collision $(b, r), (b \oplus 1, r')$ accordingly. Otherwise, if $(X_0, X_1) \notin \mathtt{SD}_{\text{YES}} \cup \mathtt{SD}_{\text{NO}}$, compute $\ell$ random collisions with the basic procedure, output the first one with $b \neq b'$, if such a collision exists, otherwise return the first sample. Each time, also append $(X_0, X_1)$ and all oracle queries made to compute the circuits' outputs.*

Clearly, for *any* polarized no-instance $(Y_0, Y_1)$ with probability at least 1/6 the basic sampling procedure returns a collision $(b, r), (b', r')$ with $b \neq b'$. See the discussion in Section 4.1. Next we show that for yes-instances (with statistical difference close to 1) this rarely happens:

**Lemma 3.** *Let $(X_0, X_1) \in \mathtt{SD}_{\text{YES}}$. Then the probability that the basic sampling procedure for $(Y_0, Y_1) = \mathrm{Polarize}(X_0, X_1)$ yields a collision $(b, r), (b, r')$ is at most $2^{-\ell/2+1}$ for $\ell = |(X_0, X_1)|$.*

*Proof.* Let $\mathrm{StatDiff}(Y_0, Y_1) \geq 1 - 2^{-\ell}$. We use the following alternative characterization of the statistical difference [59]:

$$\mathrm{StatDiff}(Y_0, Y_1) = \mathrm{Prob}\left[Y_0 \in S_0\right] - \mathrm{Prob}\left[Y_1 \in S_0\right]$$

where $S_0 = \{s \in [Y_0] \mid \mathrm{Prob}\left[Y_0 = s\right] > \mathrm{Prob}\left[Y_1 = s\right]\}$. Denote by $S_0^{\text{bad}} \subseteq S_0$ the images for which there is more than a $2^{-\ell/2}$-fraction of preimages under

$Y_1$ in comparison to $Y_0$. Our aim is to show $\text{Prob}\left[Y_0 \in S_0^{\text{bad}}\right] \leq 2^{-\ell/2}$, because for each $s \in S_0 - S_0^{\text{bad}}$ the probability that a random colliding input yields a different bit $b \neq b'$ is at most $2^{-\ell/2}$. We obviously have $\text{Prob}\left[Y_1 \in S_0^{\text{bad}}\right] \geq 2^{-\ell/2} \cdot \text{Prob}\left[Y_0 \in S_0^{\text{bad}}\right]$. Therefore,

$$\begin{aligned} 1 - 2^{-\ell} &\leq \text{StatDiff}(Y_0, Y_1) \\ &\leq \text{Prob}\left[Y_0 \in S_0\right] - \text{Prob}\left[Y_1 \in S_0^{\text{bad}}\right] \\ &\leq 1 - 2^{-\ell/2} \cdot \text{Prob}\left[Y_0 \in S_0^{\text{bad}}\right] \end{aligned}$$

This implies that $\text{Prob}\left[Y_0 \in S_0^{\text{bad}}\right] \leq 2^{-\ell/2}$. By a symmetrical argument, the same bound holds for $Y_1$. Hence, given that the sample $(b, r)$ does not fall into a bad part of the support, which happens with probability at least $1 - 2^{-\ell/2}$, we find a collision with distinct bits $b \neq b'$ with probability at least $1 - 2^{\ell/2}$. □

We omit a formal proof that we can reduce a circuit $D$ inverting $\Gamma$ to a circuit $C$ finding preimages for $\Pi$. The argument is almost identical to the one of Lemma 1, taking into account that the basic sampling procedure almost never yields collisions for the same bit $b$ for yes-instances according to the previous lemma. Namely, circuit $C_n$ computes correct answers up to length $L(n) = 4\log_2(2p(n)q(n))$ where $p(n)$ bounds $D$'s success probability and $q(n)$ bounds the size of $D$. For larger inputs $C_n$ only deviates from the answer of $\Gamma$ if it outputs a collision $(b, r), (b, r')$ for a no-instance, or returns $(b, r), (b \oplus 1, r')$ for a yes-instance. The former mismatch only occurs with probability at most $2^{-L(n)/4}$ for any query, and the latter one happens with probability at most $\ell \cdot 2^{-\ell/2+1} \leq 2^{-\ell/4} \leq 2^{-L(n)/4}$ for sufficiently large $n$ by the previous lemma. Thus, the simulation fails with probability $1/(2p(n))$ at most. Additionally, replacing $\Sigma$ by $\Gamma$ in Construction 3 of $\mathcal{T}$, we obtain a one-to-one trapdoor function granting access to $\Gamma$.

**Theorem 3.** *There exists an oracle relative to which $\mathcal{P} = \mathcal{BPP} = \mathcal{NISZK} = \mathcal{SZK} = \mathcal{HVSZK}$ but relative to which one-to-one trapdoor functions and one-way permutations exist.*

*Proof.* Obviously, relative to our (derandomized) oracles $\Gamma$ and $\mathcal{T}$ we have $\mathcal{SZK} \subseteq \mathcal{P} \subseteq \mathcal{BPP}$, because if we simply query the oracle about the input instance $(X_0, X_1)$ and output 1 (respectively, 0) if and only if we are given a collision $(b, r), (b, r')$ (respectively, $(b, r), (b \oplus 1, r')$), then we correctly decide membership for `SD` in polynomial time. Furthermore, the proofs [28, 27] that $\mathcal{NISZK} \subseteq \mathcal{HVSZK} \subseteq \mathcal{SZK}$ relativize, and together with $\mathcal{BPP} \subseteq \mathcal{NISZK}$ the assertion follows. □

## 6 Implications to Other Cryptographic Protocols

We show that various problems imply non-interactive weakly-binding honest-receiver statistically-secret commitment schemes. It follows that our oracle separation holds in these cases as well. Specifically, we discuss non-interactive crypto-computing, rerandomizable encryption schemes, and non-interactive versions of private information retrieval and oblivious transfer.

### 6.1 Non-Interactive Crypto-Computing

In [55] the problem of non-interactive crypto-computing has been partially solved. There are two parties, $A$ possessing an $\ell$-bit string $x$, and $B$ having a circuit $C$ with $\ell$ input bits and a single output bit. The security parameter $n$ as well as the input size $\ell = \text{poly}(n)$ are known by both parties. The task is now to present a protocol such that

- the protocol is non-interactive, i.e., $A$ sends a single message to $B$ who answers with a single message in a way that $A$ can extract the circuit's output,
- server-privacy: $A$ learns nothing more about $C$ than the output for $x$,
- client-privacy: $B$ learns nothing about $x$.

Sander et al. [55] present a protocol for circuits with logarithmic depth based on any rerandomizable encryption scheme (see Section 6.2), although $B$'s reply also gives away the depth of the circuit. Revealing this information does not affect our result since our circuits consist of a single or-gate, and this fact is publicly known anyway. Recently, Cachin et al. [10] presented a solution which works for any polynomial-size circuit and is non-interactive with respect to honest clients; their scheme relies on a specific algebraic assumption, namely, the decisional Diffie-Hellman assumption [9], and achieves computational server-privacy.

We think of the messages sent by both parties as generated via "encryption functions" $\text{Enc}_A$ and $\text{Enc}_B$, i.e., $A$ encrypts $x$ and $B$ computes another ciphertext from this encryption. Furthermore, the recovering algorithm of $A$ is denoted by $\text{Dec}_A$. This notation is in accordance with the protocol in [55] which is based on computing with encrypted data.

**Definition 5.** *A non-interactive honest-client statistically-server-private crypto-computer for a class $\mathcal{C}$ of circuits is a tuple* $(\text{Enc}_A, \text{Enc}_B, \text{Dec}_A)$ *of probabilistic polynomial-time algorithms such that*

- *meaningfulness: for any circuit $C : \{0,1\}^\ell \to \{0,1\}$ in $\mathcal{C}$, any $x \in \{0,1\}^\ell$ we have* $\text{Dec}_A\left(r, \text{Enc}_B(1^n, C, \text{Enc}_A(1^n, x, r))\right) = C(x)$. *That is,* $\text{Dec}_A$ *correctly decodes the circuit's output with the help of randomness $r$ and $B$'s answer* $\text{Enc}_B(1^n, C, \text{Enc}_A(x, r))$ *to $A$'s random encryption* $\text{Enc}_A(1^n, x, r)$ *of $x$.*
- *client-privacy: for any sequences $x = (x_n)_{n\in\mathbb{N}}$, $x' = (x'_n)_{n\in\mathbb{N}}$ of polynomial length with $|x_n| = |x'_n|$ for all $n \in \mathbb{N}$, the variables* $\text{Enc}_A(1^n, x_n)$ *and* $\text{Enc}_A(1^n, x'_n)$ *are computationally indistinguishable.*
- *honest-client statistical sender-privacy: there is a probabilistic polynomial-time simulator $S$ such that $S(1^n, C_n(x_n))$ and* $\text{Enc}_B(1^n, C_n, \text{Enc}_A(1^n, x_n))$ *are statistically close for any sequence of polynomial-size circuits $(C_n)_{n\in\mathbb{N}}$ with $C_n : \{0,1\}^{\ell_n} \to \{0,1\}$ in $\mathcal{C}$ and any sequence $(x_n)_{n\in\mathbb{N}}$ with $x_n \in \{0,1\}^{\ell_n}$; the probability of the variable* $\text{Enc}_B(1^n, C_n, \text{Enc}_A(1^n, x_n))$ *is taken over the randomness of* $\text{Enc}_A$ *and* $\text{Enc}_B$.

Let us explain how we reduce the problem of non-interactive crypto-computing to non-interactive statistically-secret bit commitment. Basically, we use the idea presented by Crépeau [13] of transforming a so-called oblivious transfer protocol (Section 6.3) into a bit commitment scheme. Assume that we have a non-interactive crypto-computing protocol between $A$ and $B$ for or-gates. The sender $\mathcal{S}$ will play the role of $B$ in the commitment protocol, and the honest receiver $\mathcal{R}$ acts on behalf of the honest party $A$. For security parameter $n$ the sender breaks the bit $b$ into $n$ random pieces $b_1, \dots, b_n$, i.e., each bit is selected uniformly with the constraint that the exclusive-or equals $b$, and constructs $n$ or-gates $\mathsf{OR}_{b_i} : \{0,1\} \to \{0,1\}$. Such an or-gate computes the function $a_i \mapsto a_i \vee b_i$. Now $\mathcal{R}$ randomly selects $a_1, \dots, a_n \in \{0,1\}$, samples $e_i$ according to $\mathrm{Enc}_A(1^n, a_i)$ and hands the encodings to $\mathcal{S}$. For each $i$ the sender returns $\mathrm{Enc}_B(1^n, \mathsf{OR}_{b_i}, e_i)$.

Informally, if $\mathcal{R}$ sends an ecryption of a bit $a_i = 0$ then he obtains the value $b_i$ from $\mathcal{S}$'s answer, and if $a_i = 1$ the or-operation of $b_i$ and $a_i$ equals 1, i.e., nothing about $b_i$ is revealed.[6] For random $a_i$'s, approximately half of the bits $b_i$ are therefore opened wheras the other half remains uncompromised. By the indistinguishability of the encryptions, the sender $\mathcal{S}^*$ does not know on which instances he is checked. But this knowledge is necessary to cheat by opening an uncompromised instance ambiguously.

**Lemma 4.** *If $a_i = 1$ then the distributions of $\mathrm{Enc}_B(1^n, \mathsf{OR}_{b_i}, e_i)$ for $b_i = 0$ and $b_i = 1$ are statistically close for all $e_i \in [\mathrm{Enc}_A(1^n, a_i)]$.*

*Proof.* By the server-privacy of the non-interactive crypto-computing protocol there exists a simulator whose output is almost identically distributed to $\mathrm{Enc}_B(1^n, \mathsf{OR}_{b_i}, e_i)$. Moreover, this simulator only gets the constant $\mathsf{OR}_{b_i}(a_i) = 1$ (and $n$ in unary) as input. Hence, the distributions of $\mathrm{Enc}_B(1^n, \mathsf{OR}_{b_i}, e_i)$ for both possibilities of $b_i$ are statistically close. □

It is easy to see that with exponentially small error probability $2^{-n}$ there exists at least one instance with $a_i = 1$; recall that we deal with honest receivers who choose the $a_i$'s really at random. Therefore, this scheme hides $b$ statistically against honest receivers.

**Lemma 5.** *Under the assumption that the non-interactive crypto-computer provides client-privacy, the commitment scheme binds weakly.*

*Proof.* Assume that the claim does not hold and that there is a probabilistic polynomial-time algorithm $\mathcal{S}^*$ that decommits with different $b$'s with probability $1 - \delta(n)$ for a negligible function $\delta(n)$. In particular, a distinct opening requires that for some $j$ algorithm $\mathcal{S}^*$ outputs different values for $b_j$. Then $\mathcal{S}^*$ yields a successful distinguisher concerning the client-privacy as follows.

We are given a random instance $e \in [\mathrm{Enc}_A(a)]$ for an unknown $a \in_R \{0,1\}$. We choose $j \in_R \{1, \dots, n\}$ and set $e_j = e$. Then we generate additional $n-1$ samples $e_i \in [\mathrm{Enc}_A(1^n, a_i)]$ for $i \neq j$ by selecting the $a_i$'s at random, and send these $n$ instances to $\mathcal{S}^*$. The sender $\mathcal{S}^*$ returns his commitment $\mathrm{Enc}_B(1^n, \mathsf{OR}_{b_i}, e_i)$

[6] Instead of using or-gates with wired constants $b_i \in \{0,1\}$, one may use circuits $a_i \vee a_i$ (for $b_i = 0$) and $a_i \vee \neg a_i$ (for $b_i = 1$).

for $i = 1, \ldots, n$. If $\mathcal{S}^*$ opens the value for $b_j$ differently for an ambiguous decommitment, then we output the guess "$a = 1$". Else we output "$a = 1$" with probability $\frac{1}{2} - \frac{1}{4n}$ and "$a = 0$" otherwise.

Roughly speaking, if $\mathcal{S}^*$ opens $b_j$ differently, then it is very likely that $a = 1$, because $\mathcal{S}^*$ almost never errs. Assume that indeed $a = 1$. Given that $\mathcal{S}^*$ decommits successfully for distinct values, $\mathcal{S}^*$ picks instance $j$ for revealing different values for $b_j$ with probability $\epsilon(n) \geq 1/n$. Hence, our output is right with probability at least

$$\begin{aligned}
\big(1-\delta(n)\big)\epsilon(n) &+ \big(1-\delta(n)\big)\big(1-\epsilon(n)\big)\big(\tfrac{1}{2}-\tfrac{1}{4n}\big) \\
&\geq \epsilon(n) - \delta(n) + \big(1-\delta(n)-\epsilon(n)\big)\big(\tfrac{1}{2}-\tfrac{1}{4n}\big) \\
&\geq \tfrac{1}{2} + \tfrac{\epsilon(n)}{2} - \tfrac{1}{4n} - \tfrac{3\delta(n)}{2} \\
&\geq \tfrac{1}{2} + \tfrac{1}{8n}
\end{aligned}$$

for all large $n$'s. Now let $a = 0$. Then we predict correctly with probability at least

$$\big(1-\delta(n)\big)\big(\tfrac{1}{2}+\tfrac{1}{4n}\big) \geq \tfrac{1}{2} + \tfrac{1}{4n} - \tfrac{3\delta(n)}{4} \geq \tfrac{1}{2} + \tfrac{1}{8n}$$

for sufficiently large $n$'s. Altogether, a successful adversary $\mathcal{S}^*$ would contradict the client-privacy of the crypto-computing protocol. □

Note that the lemma also shows that even a very weak form of client-privacy cannot hold, namely, that the server cannot distinguish inputs infinitely often.

**Corollary 1.** *There is an oracle relative to which black-box one-to-one trapdoor functions and black-box one-way permutations exist, but relative to which non-interactive honest-client statistically-server-private crypto-computing for* $\mathcal{C} = \{\mathsf{OR}_0, \mathsf{OR}_1\}$ *is impossible.*

### 6.2 Rerandomizable Encryption

Our result together with the non-interactive statistical server-private protocol in [55] shows that rerandomizable encryption schemes cannot be constructed from black-box trapdoor one-way functions in general. Here we present a more direct method to establish this. But before, let us recall the definition of rerandomizable encryption systems. We also discuss the issue of homomorphic encryption at the end of the section.

We give a succinct definition of a polynomially-secure public-key encryption scheme. We refer the reader to [30] for a more formal definition. A public-key bit encryption scheme is a tuple $(\mathrm{KGen}, \mathrm{Enc}, \mathrm{Dec})$ of probabilistic polynomial-time algorithms such that KGen on input $1^n$ outputs a random pair $(\mathrm{sk}, \mathrm{pk})$ of secret and public key, $\mathrm{Enc}(\mathrm{pk}, b)$ generates a random encryption $c_b$ of bit $b$ under pk, and $\mathrm{Dec}(\mathrm{sk}, c_b)$ decrypts $b$. We require that the encryption scheme is *polynomially secure*: no efficient algorithm should be able to distinguish between an encryption of 0 and one of 1.

Roughly speaking, a rerandomizable encryption scheme is a system that allows to renew the distribution of an encrypted bit from the public data alone.

That is, there is an efficient probabilistic algorithm $\Phi$ that takes as input an encryption of a bit $b$ and the public key and outputs an encryption which is identically distributed to a "fresh" encryption of $b$ under that public key.

**Definition 6.** *Let $\mathcal{E} = (\mathrm{KGen}, \mathrm{Enc}, \mathrm{Dec})$ be a polynomially-secure bit encryption scheme. We say that $\mathcal{E}$ is rerandomizable if there is a probabilistic polynomial-time algorithm $\Phi$ such that $\Phi(\mathrm{pk}, c_b)$ is identically distributed to* $\mathrm{Enc}(\mathrm{pk}, b)$ *for all* $\mathrm{pk} \in [\mathrm{KGen}(1^n)]$*, all $b \in \{0,1\}$ and any $c_b \in [\mathrm{Enc}(\mathrm{pk}, b)]$.*

Examples of such polynomially-secure, rerandomizable encryption protocols are the probabilistic encryption scheme of Goldwasser and Micali [30] on the quadratic residuosity problem, the ElGamal encryption scheme [21] based on the decisional Diffie-Hellman assumption [9], the Okamoto-Uchiyama scheme [44] and Pailler's variant [48]. An obvious extension would be to consider rerandomizing algorithms $\Phi$ whose output is statistically close to the output of the encryption process. We omit this for ease of notations, but remark that our results below carry over to this case.

Next we construct a non-interactive honest-client statistically-server-private crypto-computing protocol for or-gates $\mathsf{OR}_{b_i}$ as described in the previous section: upon receiving a public key pk and an encryption $c_{a_i}$ of a bit $a_i$ from the honest client, the server computes $\mathrm{Enc}(\mathrm{pk}, 1)$ if $b_i = 1$ and $\Phi(\mathrm{pk}, c_{a_i})$ otherwise. Put differently, if $b_i = 1$ then the server returns a "fresh" encryption of 1, and for $b_i = 0$ it rerandomizes the encryption of $a_i$. If and only if $a_i = 1$ then this latter encryption is identically distributed to an encryption for the case $b_i = 1$. It follows that this protocol computes or-gates $\mathsf{OR}_{b_i}$ non-interactively and achieves statistical server-privacy with respect to honest clients. Since the indistinguishability of encryptions guarantees the weak binding property, this constitutes a commitment scheme as in Definition 1.

**Corollary 2.** *There is an oracle relative to which black-box one-to-one trapdoor functions and black-box one-way permutations exist, but relative to which polynomially-secure rerandomizable public-key encryption is impossible.*

What about homomorphic encryption schemes? All the aforementioned rerandomizable encryption systems are also homomorphic: given encryptions of a messages $m$ and $m'$ one can derive an encryption of $m \odot m'$ without knowing the secret key. Here, $m, m'$ belong to some group with operation $\odot$. For example, the Goldwasser-Micali scheme [30] is homomorphic over $(\{0,1\}, \oplus)$, and Pailler's system [48] is homomorphic over $(\mathbb{Z}_N, +)$.

It is easy to devise a polynomial-secure homomorphic encryption scheme from any secure system. Simply concatenate encryptions of $m, m'$, and extend the decryption process to take pairs of encryptions by decipher each component and letting the decryption algorithm output $m \odot m'$. Though this somehow artifical procedure caricatures the idea of homomorphic schemes, it yet shows that our result does not apply to homomorphic encryption in general. The examples above, nonetheless, suggest that reasonable homomorphic schemes are also rerandomizable.

### 6.3 Oblivious Transfer and Private Information Retrieval

In a one-out-of-two oblivious transfer ($\frac{1}{2}$OT) protocol [22] between a sender $\mathcal{S}$ and a receiver $\mathcal{R}$, a randomly chosen one of two bits $b_0, b_1$ of $\mathcal{S}$ is transferred to $\mathcal{R}$ such that the sender does not learn which bit has been transferred, and such that the receiver does not learn anything about the other bit (but knows which of the bits he has received). The protocol is non-interactive if both the receiver and the sender send a single message. It is honest-receiver statistically-sender-private if the view of the honest receiver is statistically close for both possibilities of the unrevealed bit. Moreover, for any efficient malicious sender $\mathcal{S}^*$ it is computationally infeasibe to decide with noticeable advantage which bit has been transferred.

There are two fundamantal security notions for sender-privacy. One says that, conditioning on the received bit, the protocol distribution for any choices of the unrevealed bit should be statistically close. A stronger requirement is in the spirit of secure multi-party computations. This definition roughly demands that an adversarial receiver does not learn more from a protocol execution with the sender than in an ideal scenario where a trusted third party confidentially gets the bits from the sender and the choice from the receiver and hands the corresponding bit to the receiver. We refer to [11, 26] for general definitions of secure two-party computations and oblivious transfer. In this paper here, we only refer to the weak definition, strengthening our impossibility results.

Based on the idea in [13] it is straightforward to derive a non-interactive weakly-binding honest-receiver statistically-secret bit commitment scheme from an $\frac{1}{2}$OT protocol. Specifically, the sender chooses $b_0, b_1$ at random such that $b = b_0 \oplus b_1$ and invokes in an execution of the $\frac{1}{2}$OT protocol to transfer at random $b_0$ or $b_1$. Since $\mathcal{S}^*$ cannot decide with significant advantage which bit has been transferred, and because the receiver does not learn anything about the other bit in a statistical sense, this is a commitment scheme according to Definition 1. Therefore:

**Corollary 3.** *There is an oracle relative to which black-box one-to-one trapdoor functions and black-box one-way permutations exist, but relative to which non-interactive statistically-sender-private one-out-of-two oblivious transfer is impossible.*

We remark that non-interactive OT, as considered here, should not be confused with non-interactive OT as defined in, say, [46]. The latter paper refers to strictly non-interactive schemes, i.e., where the protocol consists of one party sending a single message only. In this model OT cannot be accomplished at all [46].

Clearly, the conclusion of Corollary 3 also holds for other variants of OT. In (the statistically-sender-private version of) Rabin's OT protocol [49] the sender possesses a single bit $b$ and the receiver learns $b$ with probability 1/2 and nothing in a statistical sense about $b$ with probability 1/2. On the other hand, $\mathcal{S}$ cannot distinguish both cases significantly. Splitting $b$ into $n$ pieces and invoking in independent executions of the non-interactive OT protocol for each piece obviously yields an appropriate commitment scheme. In a *chosen* one-out-of-two OT protocol, denoted by $\binom{2}{1}$OT, instead of determining the bit at random, the receiver decides which bit he would like to obtain; flipping a fair coin to decide reduces $\binom{2}{1}$OT to $\frac{1}{2}$OT for honest receivers.

For examples of non-interactive $\binom{2}{1}$OT protocols (under various assumptions and relying on public-key infrastructure) see [7, 16]. Recently, Naor and Pinkas [42] and Aiello et al. [3] devised statistically-sender-private $\binom{2}{1}$OT protocols which requires both parties to send a single message only and without any setup assumptions. Their protocols are based on the decisional Diffie-Hellman assumption. Hence, such oblivious transfers may be impossible using general public-key cryptosystems but they are constructible from specific intractability assumptions.

Closely related to OT schemes are private information retrieval (PIR) protocols [12]. In a (single database) PIR protocol a user reads the $i$-th bit from a database with $n$ bits, such that the database server does not learn $i$. Additionally, the communication complexity must not exceed $n$ bits, in order to improve the trivial solution of sending the whole database to the user. Nothing is guaranteed about the privacy of the data base, i.e., a malicious or even the honest user might be able to deduce more than a single bit from the server's answer. This is in contrast to OT protocols. We assume that the user fails to reconstruct the desired bit with negligible probability only (over the choice of his coin tosses).

Kushilevitz and Ostrovsky show that single database PIR is possible non-interactively based on the quadratic residuosity assumption [38], and using any trapdoor permutation with linear many rounds [39]. Beimel et al. [5] discuss which primitives single database PIR protocols imply. That is, they show that one-way functions are necessary, and if the protocol requires only half of the bits of the trivial solution of $n$ bits, then there are statistically-secret bit commitment schemes. In particular, the latter construction preserves the round-complexity. Hence, calling PIR protocols *low-communication* protocols if they merely need $n/2$ bits communication complexity for databases of size $n$, we derive:

**Corollary 4.** *There is an oracle relative to which black-box one-to-one trapdoor functions and black-box one-way permutations exist, but relative to which non-interactive low-communication single-database private information retrieval is impossible.*

Improving [5], Di Crescenzo et al. [17] show that single database PIR protocols (with communication strictly less than $n$) yield oblivious transfer schemes. The derived OT protocols, however, do not provide statistical sender-privacy and take at least four rounds. Therefore, this result does not fit in here.

## Acknowledgements

We are grateful to Dan Simon for helpful discussions about his paper. We would also like to thank all anonymous reviewers for their comments.

## References

1. W.Aiello, J.Håstad: Statistical Zero-Knowledge Languages can be Recognized in Two Rounds, *Journal of Computer and System Science, Vol. 42, pp. 327–345*, 1991.
2. W.Aiello, J.Håstad: Relativized Perfect Zero-Knowledge is not BPP, *Information and Computation, Vol. 93, pp. 223–240*, 1991.
3. W.Aiello, Y.Ishai, O.Reingold: Priced Oblivious Transfer: How to Sell Digital Goods, *Eurocrypt 2001, Lecture Notes in Computer Science, Vol. 2045, Springer-Verlag*, 2001.
4. J.Balcázar, J.Díaz, J.Gabarró: Structural Complexity I (Second Edition), *Springer-Verlag*, 1995.
5. A.Beimel, Y.Ishai, E.Kushilevitz, T.Malkin: One-Way Functions are Essential for Single-Server Private Information Retrieval, *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing (STOC), pp. 89–98*, 1999.
6. M.Bellare, S.Halevi, A.Sahai, S.Vadhan: Many-To-One Trapdoor Functions and Their Relation to Public-Key Cryptosystems, *Crypto '98, Lecture Notes in Computer Science, Vol. 1462, Springer-Verlag, pp. 283–298*, 1998.
7. M.Bellare, S.Micali: Non-Interactive Oblivious Transfer and Applications, *Crypto '89, Lecture Notes in Computer Science, Vol. 435, Springer-Verlag, pp. 547–559*, 1990.
8. M.Ben-Or, O.Goldreich, S.Goldwasser, J.Håstad, J.Killian, S.Micali, P.Rogaway: Everything Provable is Provable in Zero-Knowledge, *Crypto '88, Lecture Notes in Computer Science, Vol. 403, Springer-Verlag, pp. 37–56*, 1990.
9. D.Boneh: The Decision Diffie-Hellman Problem, *Third Algorithmic Number Theory Symposium, Lecture Notes in Computer Science, Vol. 1423, Springer-Verlag, pp. 48–63*, 1998.
10. C.Cachin, J.Camenisch, J.Kilian, J.Müller: One-Round Secure Computation and Secure Autonomous Mobile Agents, *ICALP 2000, Lecture Notes in Computer Science, Springer-Verlag*, 2000.
11. R.Canetti: Security and Composition of Multiparty Cryptographic Protocols, *Journal of Cryptology, Vol. 13, No. 1, Springer-Verlag, pp. 143–202*, 2000.
12. B.Chor, O.Goldreich, E.Kushilevitz, M.Sudan: Private Information Retrieval, *Journal of ACM, vol. 45, pp. 965–981*, 1998.
13. C.Crépeau: Equivalence Between two Flavours of Oblivious Transfer, *Crypto '87, Lecture Notes in Computer Science, Vol. 293, Springer-Verlag, pp. 350–354*, 1988.
14. C.Crépeau, F.Légaré, L.Savail: How to Convert a Flavor of Quantum Bit Commitment, *Eurocrypt 2001, Lecture Notes in Computer Science, Vol. 2045, Springer-Verlag*, 2001.
15. I.Damgård, T.Pedersen, B.Pfitzmann: On the Existence of Statistically Hiding Bit Commitment Schemes and Fail-Stop Signatures, *Crypto '93, Lecture Notes in Computer Science, Vol. 773, Springer-Verlag, pp. 250–255*, 1993.
16. A.De Santis, G.Di Crescenzo, G.Persiano: Public-Key Cryptography and Zero-Knowledge Arguments, *Information and Computation, Vol. 121, No. 1, pp. 23–40*, 1995.
17. G.Di Crescenzo, T.Malkin, R.Ostrovsky: Single Database Private Information Retrieval Implies Oblivious Transfer, *Eurocrypt '00, Lecture Notes in Computer Science, Vol. 1807, Springer-Verlag*, 2000.

18. G.Di Crescenzo, T.Okamoto, M.Yung: Keeping the SZK-Verifier Honest Unconditionally, *Crypto '97, Lecture Notes in Computer Science, Vol. 1294, Springer-Verlag, pp. 31–45*, 1997.
19. W.Diffie, M.Hellman: New Directions in Cryptography, *IEEE Transaction on Information Theory, Vol. 22, pp. 644–654*, 1976.
20. P.Dumais, D.Mayers, L.Salvail: Perfectly Concealing Quantum Bit Commitment from Any One-Way Permutation, *Eurocrypt 2000, Lecture Notes in Computer Science, Vol. 1807, Springer-Verlag, pp. 300–315*, 2000.
21. T.ElGamal: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, *IEEE Transaction on Information Theory, Vol. 31, pp. 469–472*, 1985.
22. S.Even, O.Goldreich, A.Lempel: A Randomized Protocol for Signing Contracts, *Communication of the ACM, vol. 28, pp. 637–647*, 1985.
23. L.Fortnow: The Complexity of Perfect Zero-Knowledge, *Proceedings of the 19th Annual ACM Symposium on the Theory of Computing (STOC), pp. 204–209*, 1987.
24. R.Gennaro, L.Trevisan: Lower Bounds on the Efficiency of Generic Cryptographic Constructions, *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science (FOCS)*, 2000.
25. Y.Gertner, S.Kannan, T.Malkin, O.Reingold, M.Viswanathan: The Relationship Between Public Key Encryption and Oblivious Transfer, *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science (FOCS)*, 2000.
26. O.Goldreich: Secure Multi-Party Computation, *(working draft, version 1.2), available at* `http://www.wisdom.weizmann.ac.il/home/oded/public_html/pp.html`, March 2000.
27. O.Goldreich, A.Sahai, S.Vadhan: Can Statistical Zero-Knowledge be made Non-Interactive? or On the Relationship of SZK and NISZK, *Crypto '99, Lecture Notes in Computer Science, Springer-Verlag*, 1999.
28. O.Goldreich, A.Sahai, S.Vadhan: Honest-Verifier Statistical Zero-Knowledge Equals General Statistical Zero-Knowledge, *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC), ACM Press, pp. 399–408*, 1998.
29. S.Goldwasser, O.Goldreich, S.Micali: How to Construct Random Functions, *Journal of ACM, vol. 33, pp. 792–807*, 1986.
30. S.Goldwasser, S.Micali: Probabilistic Encryption, *Journal of Computer and System Science, Vol. 28, pp. 270–299*, 1984.
31. S.Halevi, S.Micali: Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing, *Crypto '96, Lecture Notes in Computer Science, Vol. 1109, Springer-Verlag, pp. 201–215*, 1996.
32. J.Håstad, R.Impagliazzo, L.Levin, M.Luby: A Pseudorandom Generator from any One-way Function, *SIAM Journal on Computing, vol. 28(4), pp. 1364–1396*, 1999.
33. R.Impagliazzo, M.Luby: One-Way Functions are Essential for Complexity Based Cryptography, *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 230–235*, 1989.
34. R.Impagliazzo, S.Rudich: Limits on the Provable Consequences of One-Way Permutations, *Proceedings of the 21st Annual ACM Symposium on the Theory of Computing (STOC), pp. 44–61*, 1989.
35. R.Impagliazzo, M.Yung: Direct Minimum-Knowledge Computations, *Crypto '87, Lecture Notes in Computer Science, Vol. 293, Springer-Verlag, pp. 40–51*, 1987.

36. J.Kahn, M.Saks, C.Smyth: A Dual Version of Reimer's Inequality and a Proof of Rudich's Conjecture, *Proceedings of 15th IEEE Conference on Computational Complexity*, 2000.
37. J.Kim, D.Simon, P.Tetali: Limits on the Efficiency of One-Way Permutation-Based Hash Functions, *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS)*, 1999.
38. E.Kushilevitz, R.Ostrovsky: Replication is not Needed: Single Database, Computationally-Private Information Retrieval, *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 364–373*, 1997.
39. E.Kushilevitz, R.Ostrovsky: One-Way Trapdoor Permutations are Sufficient for Single-Server Private Information Retrieval, *Eurocrypt '00, Lecture Notes in Computer Science, Vol. 1807, Springer-Verlag*, 2000.
40. M.Naor: Bit Commitment Using Pseudo-Randomness, *Journal of Cryptology, vol. 4, pp. 151–158*, 1991.
41. M.Naor, R.Ostrovsky, R.Venkatesan, M.Yung: Perfect Zero-Knowledge Arguments for NP Using Any One-Way Permutation, *Journal of Cryptology, vol. 11, pp. 87–108*, 1998.
42. M.Naor, B.Pinkas: Efficient Oblivious Transfer Protocols, *Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2001.
43. M.Naor, M.Yung: Universal One-Way Hash Functions and Their Cryptographic Applications, *Proceedings of the 21st Annual ACM Symposium on the Theory of Computing (STOC), pp. 33–43*, 1989.
44. T.Okamoto, S.Uchiyama: A New Public-Key Cryptosystem as Secure as Factoring, *Eurocrypt '98, Lecture Notes in Computer Science, Vol. 1403, Springer-Verlag, pp. 308–318*, 1998.
45. R.Ostrovsky: One-Way Functions, Hard on Average Problems, and Statistical Zero-Knowledge Proofs, *IEEE Conference on Structure in Complexity Theory, pp. 133–138*, 1991.
46. R.Ostrovsky, R.Venkatesan, M.Yung: Fair Games Against an All-Powerful Adversary, *AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 13, pp. 155–169*, 1993.
47. R.Ostrovsky, A.Wigderson: One-Way Functions are Essential for Non-Trivial Zero-Knowledge, *Proceedings of the Second Israel Symposium on Theory of Computing and Systems*, 1993.
48. P.Pailler: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, *Eurocrypt '99, Lecture Notes in Computer Science, Vol. 1592, Springer-Verlag, pp. 223–238*, 1999.
49. M.Rabin: How to Exchange Secrets by Oblivious Transfer, *Technical Report TR-81, Harvard*, 1981.
50. R.Rivest, A.Shamir, L.Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Communication of the ACM, vol. 21(2), pp. 120–126*, 1978.
51. J.Rompel: One-Way Functions are Necessary and Sufficient for Secure Signatures, *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing (STOC), pp. 387–394*, 1990.
52. S.Rudich: The Use of Interaction in Public Cryptosystems, *Crypto '91, Lecture Notes in Computer Science, Vol. 576, Springer-Verlag, pp. 242–251*, 1992.
53. A.Sahai, S.Vadhan: A Complete Promise Problem for Statistical Zero-Knowledge, *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 448–457*, 1997.

54. A.Sahai, S.Vadhan: Manipulating Statistical Difference, *AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 43, pp. 251–270*, 1999.
55. T.Sander, A.Young, M.Yung: Non-Interactive Crypto-Computing for $NC^1$, *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS)*, 1999.
56. A.Shamir: IP=PSPACE, *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science (FOCS)*, 1990.
57. D.Simon: On the Power of Quantum Computation, *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 124–134*, 1994.
58. D.Simon: Finding Collisions on a One-Way Street: Can Secure Hash Functions be Based on General Assumptions?, *Eurocrypt '98, Lecture Notes in Computer Science, Vol. 1403, Springer-Verlag, pp. 334–345*, 1998.
59. S.Vadhan: A Study of Statistical Zero-Knowledge Proofs, *Ph.D. thesis, MIT, available at* `http://theory.lcs.mit.edu/~salil/`, September 1999.

## A One-to-One Trapdoor Function Oracle Over Bit Strings

Similar to oracles $\Pi$ and $\Sigma$ we can turn $\mathcal{T}$ into a single oracle over $\{0,1\}^*$ by encoding the query states $\mathsf{generate}, \mathsf{evaluate}, \mathsf{invert}$ in $\{0,1\}^*$ appropriately. To do so, we interpret $(\mathsf{generate}, \omega)$ and $(\mathsf{invert}, t, y)$ as strings $(1010\cdots 10, \omega, \ldots, \omega)$ and $(0101\cdots 01, t, y, \ldots, t, y)$, respectively, for a sufficiently large number of 10- and 01-repetitions (say, half of the input length), and prepend the encodings to the output again. We also let those images map to the corresponding preimages to preserve the permutation property. Evaluation queries $(\mathsf{evaluate}, z)$ are simply encoded as $z$. Moreover, we change the evaluation portion of $\mathcal{T}$ to $\Sigma(\cdot, i)$, that is, we swap index and argument. The reason is that if we now get a request for inverting $(\mathsf{generate}, v)$, then the preimage under $\Sigma(\cdot, w)$ equals $\mathsf{generate} = 10\cdots 10$ (where the right half $w$ can be computed from $v$ via $\Sigma$; if $w$ does not match $v$ then the preimage is of course the undefined symbol). Analogously for $(\mathsf{invert}, v)$-queries. Hence, such inversion queries can be easily computed. On the other hand, most computations for random $x$ still result in $f$-evaluations, and therefore $\mathcal{T}$ constitutes a trapdoor one-way function.