Finding Collisions in Interactive Protocols – A Tight Lower Bound on the Round Complexity of Statistically-Hiding Commitments

Iftach Haitner* Jonathan J. Hoch* Omer Reingold*† Gil Segev*

Abstract

We study the round complexity of various cryptographic protocols. Our main result is a tight lower bound on the round complexity of any fully-black-box construction of a statistically-hiding commitment scheme from one-way permutations, and even from trapdoor permutations. This lower bound matches the round complexity of the statistically-hiding commitment scheme due to Naor, Ostrovsky, Venkatesan and Yung (CRYPTO '92). As a corollary, we derive similar tight lower bounds for several other cryptographic protocols, such as *single-server private information retrieval*, *interactive hashing*, and *oblivious transfer* that guarantees statistical security for one of the parties.

Our techniques extend the collision-finding oracle due to Simon (EUROCRYPT '98) to the setting of interactive protocols (our extension also implies an alternative proof for the main property of the original oracle). In addition, we substantially extend the reconstruction paradigm of Gennaro and Trevisan (FOCS '00). In both cases, our extensions are quite delicate and may be found useful in proving additional black-box separation results.

*Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel. Email: {iftach.haitner,yaakov.hoch,omer.reingold,gil.segev}@weizmann.ac.il.

†Research supported by grant 1300/05 from the Israel Science Foundation.

1 Introduction

Research in the foundations of cryptography is concerned with the construction of provably secure cryptographic tools. The security of such constructions relies on a growing number of computational assumptions, and in the last few decades much research has been devoted to demonstrating the feasibility of particular cryptographic tasks based on the weakest possible assumptions. For example, the existence of one-way functions has been shown to be equivalent to the existence of pseudorandom functions and permutations [22, 44], pseudorandom generators [3, 32], universal one-way hash functions and signature schemes [48, 53], different types of commitment schemes [31, 32, 45, 49], private-key encryption [21] and other primitives.

Many constructions based on minimal assumptions, however, result in only a theoretical impact due to their inefficiency, and in practice more efficient constructions based on seemingly stronger assumptions are being used. Thus, identifying tradeoffs between the *efficiency* of cryptographic constructions and the strength of the computational assumptions on which they rely is essential in order to obtain a better understanding of the relationship between cryptographic tasks and computational assumptions.

In this paper we follow this line of research, and study the tradeoffs between the *round complexity* of cryptographic protocols and the strength of their underlying computational assumptions. We provide a lower bound on the round complexity of black-box constructions of statistically-hiding and computationally-binding commitment schemes (for short, statistical commitment schemes) based on one-way permutations and on families of trapdoor permutations. Our lower bound matches known upper bounds resulting from [46]. As a corollary of our main result, we derive similar tight lower bounds for several other cryptographic protocols, such as *single-server private information retrieval*, *interactive hashing*, and *oblivious transfer* that guarantees statistical security for one of the parties.

Although in the current paper our techniques are used to derive lower bounds for a particular efficiency measure, namely that of the round complexity of cryptographic protocols, they may be found useful in proving additional black-box separation results. In the following paragraphs, we discuss the notion of statistically-hiding commitment schemes and describe the setting in which our lower bounds are proved.

Statistically-hiding commitment schemes. A commitment scheme defines a two-stage interactive protocol between a sender $\mathcal{S}$ and a receiver $\mathcal{R}$; informally, after the *commit stage*, $\mathcal{S}$ is bound to (at most) one value, which stays hidden from $\mathcal{R}$, and in the *reveal stage* $\mathcal{R}$ learns this value. The two security properties hinted at in this informal description are known as *binding* ($\mathcal{S}$ is bound to at most one value after the commit stage) and *hiding* ($\mathcal{R}$ does not learn the value to which $\mathcal{S}$ commits before the reveal stage). In a statistical commitment scheme the hiding property holds *even against all-powerful receivers* (i.e., the hiding holds information-theoretically), while the binding property is required to hold only for polynomially-bounded senders.

Statistical commitments can be used as a building block in constructions of statistical zero-knowledge arguments [5, 46] and of certain coin-tossing protocols [41]. When used within protocols in which certain commitments are never revealed, statistical commitments have the following advantage over computationally-hiding commitment schemes: in such a scenario, it should be infeasible to violate the binding property *only during the execution of the protocol*, whereas the committed values will remain hidden *forever* (i.e., regardless of how much time the receiver invests after the completion of the protocol).

Statistical commitments schemes with a constant number of rounds were shown to exist based on specific number-theoretic assumptions [4, 5] (or, more generally, based on any collection of claw-free permutations [26] with an efficiently-recognizable index set [23]), and collision-resistant hash functions [10, 48]. Protocols with higher round complexity were shown to exist based on different types of one way functions. Protocols with $O\left(\frac{n}{\log n}\right)$ rounds (where n is the input length of the underlying function) were based on one-way permutations [46] and (known-) regular one-way functions [29].[1] Finally, a protocol with a polynomial number of rounds was based on any one-way function [31, 49].

Black-box reductions. As already mentioned, we are interested in proving lower bounds on the round complexity of various cryptographic constructions. In particular, we are interested in showing that any construction of statistical commitments based on trapdoor permutations requires a fairly large number of rounds. Nevertheless, under standard assumptions such as the existence of collision-resistant hash functions, *constant-round statistical commitments do exist*. So if these assumptions hold, then the existence of trapdoor permutations implies the existence of constant-round statistical commitments in a *trivial logical sense*. Faced with similar difficulties, Impagliazzo and Rudich [34] presented a paradigm for proving impossibility results under a restricted, yet important, subclass of reductions called *black-box reductions*. Their method was extended to showing lower bounds on the *efficiency* of reductions by Kim, Simon and Tetali [37].

Intuitively, a black-box reduction of a primitive P to a primitive Q is a construction of P out of Q that ignores the internal structure of the implementation of Q and just uses it as a "subroutine" (i.e., as a black-box). In addition, in the case of fully-black-box reductions, the proof of security (showing that an adversary that breaks the implementation of P implies an adversary that breaks the implementation of Q), is also black-box (i.e., the internal structure of the adversary that breaks the implementation of P is ignored as well). For a more exact treatment of black-box reductions see Section 2.3.

1.1 Related Work

Impagliazzo and Rudich [34] showed that there are no black-box reductions of key-agrement protocols to one-way permutations and substantial additional work in this line followed (c.f. [18, 55, 57]). Kim, Simon and Tetali [37] initiated a new line of impossibility results, by providing a lower bound on the *efficiency* of black-box reductions (rather than on their feasibility). They proved a lower bound on the efficiency, in terms of the number of calls to the underlying primitive, of any black-box reduction of universal one-way hash functions to one-way permutations. This result was later improved, to match the known upper bound, by Gennaro et al. [16], which also provided tight lower bounds on the efficiency of several other black-box reductions [14, 15, 16]. Building upon the technique developed by [16], Horvitz and Katz [33] gave lower bounds on the efficiency of black-box reductions of statistically-binding commitments to one-way permutations. In all the above results the measure of efficiency under consideration is the number of calls to the underlying primitives.

With respect to the round complexity of statistical commitments, Fischlin [13] showed that every black-box reduction of statistical commitments to trapdoor permutations, has at least two rounds. His result follows Simon's oracle separation of collision-resistant hash functions from one-way permutations [57]. Recently, Wee [58] considered a restricted class of black-box reductions of statistical commitments to one-way permutations. Informally, Wee considered only constructions in which the sender first queries the one-way permutation on several independent inputs. Once the interaction with the receiver starts, the sender only access the outputs of these queries (and not the

[1] The original presentations of the above protocols have $O(n)$ rounds. By a natural extension, however, the number of rounds in these protocols can be reduced to $O\left(\frac{n}{\log n}\right)$, see [30, 38].

inputs) and does not perform any additional queries. Wee showed that every black-box reduction of the above class has $\Omega\left(\frac{n}{\log n}\right)$ communication rounds.

The question of deriving lower bounds on the round complexity of black-box reductions, was also addressed in the context of zero-knowledge protocols [7, 11, 24, 27, 36, 54]. In this context, however, the black-box access is to the, possibly cheating, verifier and not to any underlying primitive.

1.2 Our Results

We study the class of fully-black-box constructions of statistically-hiding commitment schemes from trapdoor permutations, and prove a lower bound on the round complexity of any such construction. Informally, our main theorem is as follows:

Main Theorem (Informal). *Any fully-black-box construction of a statistically-hiding commitment scheme from a family of trapdoor permutations over* $\{0,1\}^n$ *has* $\Omega\left(\frac{n}{\log n}\right)$ *communication rounds.*

In fact, we consider a more general notion of hardness for trapdoor permutations, which extends the standard polynomial hardness requirement. Informally, we say that a trapdoor permutation τ over $\{0,1\}^n$ is $s(n)$-hard if any probabilistic Turing-machine that runs in time $s(n)$ inverts τ on a uniformly chosen image with probability at most $1/s(n)$. Given this definition, we show that any fully-black-box construction of a statistically-hiding commitment scheme from a family of $s(n)$-hard trapdoor permutations over $\{0,1\}^n$ requires $\Omega\left(\frac{n}{\log s(n)}\right)$ communication rounds.

Our lower bound, for both notions of trapdoor permutations, matches the known upper bound due to [46, 30, 38]. The scheme of Naor et al. relies on one-way permutations in a fully-black-box manner, and thus we demonstrate that their scheme is essentially optimal with respect to the number of communication rounds. Moreover, our lower bound implies that trapdoor permutations are not superior to one-way permutations in this setting, whereas collision-resistant hash functions and specific number-theoretic assumptions are superior and imply schemes with a constant number of rounds.

Taking the security of the reduction into account. Note that the informal statement of our main theorem considers constructions which invoke only trapdoor permutations over n bits. We would like to extend the result to consider constructions which may invoke the trapdoor permutations over more than a single domain. However, in this case, better upper bounds are known. In particular, given security parameter 1^n it is possible to apply the scheme of Naor et al. using a one-way permutation over n^ϵ bits. This implies statistical commitments that run in $O(n^\epsilon)$ rounds. This subtle issue is not unique to our setting, and in fact arises in any study of the efficiency of cryptographic reductions (see, in particular, [16, 58]). The common approach for addressing this issue is by restricting the class of constructions (as in the informal statement of our main theorem above). In Section 6 we follow a less restrictive approach: we consider constructions which are given access to trapdoor permutations over *any* domain size, but require that the proof of security will be "somewhat security preserving". More specifically, we consider an additional parameter, which we refer to as the *security-parameter-expansion* of the construction. Informally, the proof of security in a fully-black-box construction gives a way to translate (in a black-box manner) an adversary S^* that breaks the binding of the commitment scheme into an adversary A that breaks the security of the trapdoor permutation. Such a construction is $\ell(n)$-security-parameter-expanding if whenever the machine A tries to invert a permutation over n bits, it invokes S^* on security parameters which are at most $1^{\ell(n)}$. It should be noted that any construction in which $\ell(n)$ is significantly larger than

n, may only be weakly security preserving (for a taxonomy of security preserving reductions see [43, Lecture 2]).

Our lower bound proof takes into consideration the security parameter expansion, and therefore our statements apply for the most general form of fully-black-box reductions. In particular, in case that $\ell(n) = O(n)$, our theorem implies that the required number of rounds is $\Omega\left(\frac{n}{\log n}\right)$, and in the general case (where $\ell(n)$ may be any polynomial in n), our theorem implies that the required number of rounds is $n^{\Omega(1)}$ (which as argued above is tight as well).

Implications to other cryptographic protocols. Our main result can be extended to any cryptographic protocol which implies statistically-hiding commitment schemes in a fully-black-box manner, as long as the reduction essentially preserves the number of communication rounds. Specifically, we derive similar $\Omega\left(\frac{n}{\log n}\right)$ lower bounds on the round complexity of fully-black-box constructions from trapdoor permutations of single-server private information retrieval, interactive hashing, and oblivious transfer that guarantees statistical security for one of the parties.

1.3 Overview of the Technique

For the sake of simplicity, we concentrate in this overview on the round complexity lower bound for statistical commitment which are based on one-way permutations (the lower bound for constructions based on families of trapdoor permutations follows similar ideas). We also assume without loss of generality that the sender's secret in the commitment protocol is a single uniform bit. Let us start by considering Simon's oracle [57] for ruling out any black-box reduction of a family of collision resistant hash functions to one-way permutation.

Simon's oracle. Simon's oracle $\mathsf{ColFinder}$ gets as an input a circuit C, possibly with π gates,[2] where π is a random permutation. It then outputs two random elements w_1 and w_2 such that $C(w_1) = C(w_2)$. Clearly, in the presence of $\mathsf{ColFinder}$ no family of collision resistant hash functions exists (the adversary simply queries $\mathsf{ColFinder}$ with the hash function circuit to find a collision). In order to rule out the existence, in the presence of $\mathsf{ColFinder}$, of any two-round statistical commitment scheme, Fischlin [13] used the following adversary $\mathcal{S}^*$ to break any such scheme: assume w.l.o.g. that the first message, q_1 is sent by $\mathcal{R}$ and consider the circuit C_{q_1}, naturally defined by q_1 and $\mathcal{S}$. Namely, C_{q_1} gets as an input the random coins of $\mathcal{S}$ and outputs the answer that $\mathcal{S}$ replies on receiving the message q_1 from $\mathcal{R}$. In the commit stage after receiving the message q_1, the cheating $\mathcal{S}^*$ constructs C_{q_1}, queries $\mathsf{ColFinder}(C_{q_1})$ to get w_1 and w_2, and answers as $\mathcal{S}(w_1)$ would (i.e., by $C_{q_1}(w_1)$). In the reveal stage, $\mathcal{S}^*$ uses both w_1 and w_2 to open the commitment (i.e. once using the random coins w_1 and then using w_2). Since the protocol is statistically hiding, the set of the sender's random coins that are consistent with this commit stage transcript is divided to almost equal size parts by the values of their secret bits. Therefore, with probability roughly half w_1 and w_2 will differ on the value of $\mathcal{S}$'s secret bit and the binding of the commitment will be violated.

In order to obtain the black-box impossibility results (both of [57] and of [13]), it is left to show that π is one-way in the presence of $\mathsf{ColFinder}$. Let A be a circuit trying to invert π on a random $y \in \{0,1\}^n$ using $\mathsf{ColFinder}$, and lets assume for now that A makes only a single call to $\mathsf{ColFinder}$. Intuitively, the way we could hope this query to $\mathsf{ColFinder}$ with input C could help is by "hitting" y in the following sense: we say that $\mathsf{ColFinder}$ *hits* y on input C, if the computations of $C(w_1)$ or of $C(w_2)$ query π on $\pi^{-1}(y)$. Now we note that for every input circuit C each one of w_1 and w_2

[2] In fact, $\mathsf{ColFinder}$ also accepts circuits C with $\mathsf{ColFinder}$ gates. For the sake of this discussion, we ignore this property.

(the outputs of $\mathsf{ColFinder}$ on C) is *individually* uniform. Therefore, the probability that $\mathsf{ColFinder}$ hits y on input C, may only be larger by a factor two than the probability that evaluating C on a uniform w queries π on $\pi^{-1}(y)$. In other words, A does not gain much by querying $\mathsf{ColFinder}$ (as A can evaluate C on a uniform w on its own). Formalizing the above intuition is far from easy, mainly when we consider A that queries $\mathsf{ColFinder}$ more than once. The difficulty lies in formalizing the claim that the only useful queries are the ones in which $\mathsf{ColFinder}$ hits y (after all, the reply to a query may give us some useful global information on π). We give some intuition in Section 1.3.1 for why this claim is valid, following a different approach than the original proof due to [57] (our version of the proof extends the reconstruction technique of Gennaro and Trevisan).

Finding collisions in interactive protocols. We would like to employ Simon's oracle for breaking the binding of more interactive protocols (with more than two rounds). Unfortunately, the "natural" attempts to do so seem to fail miserably. The first attempt that comes to mind might be the following: In the commit stage $\mathcal{S}^*$ follows the protocol and let $q_1, \ldots, q_k$ be the messages that $\mathcal{R}$ sent in this stage. In the reveal stage, $\mathcal{S}^*$ queries $\mathsf{ColFinder}$ to get a colliding pair (w_1, w_2) in $C_{q_1,\ldots,q_k}$ - the circuit naturally defined by the code of $\mathcal{S}$ and $q_1, \ldots, q_k$ (i.e., $C_{q_1,\ldots,q_k}$ gets as an input the random coins of $\mathcal{S}$ and outputs the messages sent by $\mathcal{S}$ when $\mathcal{R}$'s messages are $q_1, \ldots, q_k$). The problem is that it is very unlikely that the outputs of Sam on $C_{q_1,\ldots,q_k}$ will be consistent with the answers that $\mathcal{S}^*$ *already* gave in the commit stage (we did not encounter this problem when breaking two-round protocols, since $\mathcal{S}^*$ could query $\mathsf{ColFinder}$ on C_{q_1} before $\mathcal{S}^*$ sends its first and only message). Alternatively, we could change $\mathsf{ColFinder}$ such that it gets as an additional input w_1 and returns w_2 for which $C_{q_1,\ldots,q_k}(w_1) = C_{q_1,\ldots,q_k}(w_2)$ (that is, the new $\mathsf{ColFinder}$ finds second preimages rather than collisions). Indeed, this new $\mathsf{ColFinder}$ does imply the breaking of any commitment scheme, but it also implies the inversion of π.[3] We should not be too surprised that both the above attempts failed as they are both completely oblivious of the round complexity of $(\mathcal{S}, \mathcal{R})$. Since one-way permutations *do imply* statistical commitments in a black-box manner any oracle that breaks statistical commitments could also be used to break the underlying one-way permutations.[4]

For our oracle separation, we manage to extend Simon's oracle to the setting of interactive protocols. We will have to handle interaction with care so that our oracle is not too strong (so that it does not break the one-way permutations), but still strong enough to be useful. In fact, the more interactive our oracle will be the more powerful it will be, and eventually it will allow breaking the one-way permutations. Quantifying this growth in power is how we get our tight bounds on the round complexity of the reduction.

Our oracle. It will be useful for us to view Simon's oracle as performing two sampling tasks: First it samples w_1 uniformly and then it samples a second preimage w_2 such that $C(w_1) = C(w_2)$. As explained above, an oracle for sampling a second preimage allows inverting the one-way permutations. The reason the sampling done by $\mathsf{ColFinder}$ is not too damaging is that w_1 *was chosen by* $\mathsf{ColFinder}$ *after* C *is already given.* Therefore, an adversary A is very limited in setting up the second distribution from which $\mathsf{ColFinder}$ samples (i.e. the uniform distribution over the preimages of $C(w_1)$ under C). In other words, this distribution is jointly defined by A and $\mathsf{ColFinder}$ itself.

[3] Consider a circuit C, whose input is composed of a bit σ and an n-bit string w. The circuit C is defined by $C(0, w) = \pi(w)$ and $C(1, w) = w$. Thus, in order to compute $\pi^{-1}(y)$ we can simply invoke the new $\mathsf{ColFinder}$ on input C and $w_1 = (1, y)$. With probability half $\mathsf{ColFinder}$ will return $w_2 = (0, \pi^{-1}(y))$.

[4] In addition, in both these naive attempts the cheating sender $\mathcal{S}^*$ follows the commit stage honestly (as $\mathcal{S}$ would). It is not hard to come up with two-round protocol that works well for semi-honest commit stage senders (consider for instance the two-round variant of [46] where the receiver's queries are all sent in the first round).

Extending the above interpretation of $\mathsf{ColFinder}$ (and ignoring various technical aspects), our separation oracle Sam is defined as follows: Sam will be given as input a query $Q = (C_{\text{next}}, C, z)$, and will output a pair (w', z') where w' is a uniformly distributed preimage of z under the mapping defined by the circuit C, and $z' = C_{\text{next}}(w')$. Following the intuition above we impose the restriction that there was a previous query $(C, \cdot, \cdot)$ that was answered by (w, z) (note that this imposes a forest-like structure on the queries). In other words, C was announced *before w was chosen by* Sam *in answering the previous query.*[5] In addition, we only allow querying Sam up to depth $d(n)+1$ where n is the security parameter (this depth function $d(\cdot)$ will depend on the particular lower bound we will try to prove).

Sam allows breaking $d(n)$-round statistical commitments. The adversary $\mathcal{S}^*$ operates as follows: after getting the first message q_1, it constructs C_{q_1} (the circuit that computes $\mathcal{S}$'s first message) and queries Sam for a random input w_1 (i.e., it queries Sam without specifying C and z), and sends $\mathcal{R}$ the message specified by $z_1 = C_{q_1}(w_1)$. On getting the i-th receiver message q_i, the adversary $\mathcal{S}^*$ constructs $C_{q_1,\ldots,q_i}$ (the circuit that computes $\mathcal{S}$'s first i messages), queries Sam on $(C_{q_1,\ldots,q_i}, C_{q_1,\ldots,q_{i-1}}, z_{i-1})$ to get (w_i, z_i), and replies to $\mathcal{R}$ with the message specified by $z_i = C_{q_1,\ldots,q_i}(w_i)$. Finally, after completing the commit stage (when answering the last receiver message q_d) it queries Sam on $(\perp, C_{q_1,\ldots,q_d}, z_d)$ to get w_{d+1}, z_{d+1}. Both w_d and w_{d+1} are sender's random inputs that are consistent with the commit-stage transcript. Therefore, with probability roughly half they can be used to break the binding of the protocol.

Sam cannot be used to invert random permutations. To complete our impossibility result, it is left to prove that Sam cannot be used to invert the random permutation π. As in our intuition for Simon's oracle, we would like to claim that the only useful Sam-queries for an adversary A that tries to invert π on y are queries that make Sam hit y. Assume Sam is given as input a query (C_{next}, C, z), and outputs a pair (w', z'). We say that Sam hits y if evaluating $C(w')$ queries π on $\pi^{-1}(y)$. Extending the reconstruction technique of Gennaro and Trevisan, we show that A is unlikely to invert π on y if it does not make Sam hit y (see Section 1.3.1).

The most technical part of the paper is showing that a circuit A that inverts π on y while making Sam hit y can be transformed into a circuit M that inverts π without Sam hitting y. This aspect of the proof is somewhat influenced by the work of Wee [58]. Let us try to give some intuition for this claim. Assume for simplicity of notation that A only makes the following queries: $(C_1, \perp, \perp), (C_2, C_1, z_1), \ldots, (C_{d+1}, C_d, z_d)$ and it receives the corresponding replies: $(w_1, z_1), \ldots, (w_{d+1}, z_{d+1})$. We know that for some i the probability that the computation $C_i(w_{i+1})$ queries π on $\pi^{-1}(y)$ (i.e., hits y) is non-negligible (as we know that Sam is likely to hit y). On the other hand the probability that $C_1(w_2)$ hits y (which is identical to the probability that $C_1(w_1)$ hits y) is exponentially small. Therefore, unless $d = \Omega\left(\frac{n}{\log n}\right)$ we have that there exists a location i such that the probability $C_i(w_{i+1})$ hits y is larger than the probability that $C_{i-1}(w_i)$ hits y by a very large polynomial. We are also able to show (under the various restrictions on Sam) that the probability that the computation $C_i(w_i)$ hits y is unlikely to be much smaller than the probability that the computation $C_i(w_{i+1})$ hits y. Combining the above understandings we design M that inverts π on y with non-negligible probability without making Sam hit y (and this will constitute a contradiction). M simulates A but in addition, whenever A queries Sam for (C_{i+1}, C_i, z_i) and receives a reply (w_{i+1}, z_{i+1}) we let M also evaluate $C_{i+1}(w_{i+1})$. If this computation queries π on $\pi^{-1}(y)$ then M halts and outputs $\pi^{-1}(y)$. Otherwise, M continues with the simulation of A. We argue that

[5] An additional important restriction that we will not discuss here is that C_{next} is a *refinement* of the circuit C, where by refinement we mean that $C_{\text{next}}(w) = (C(w), \widetilde{C}(w))$ for some circuit $\widetilde{C}$ and for every w.

with sufficiently large probability, if the first query of A that makes Sam hit y is (C_{i+1}, C_i, z_i), then M's computation of $C_i(w_i)$ queries π on $\pi^{-1}(y)$. Therefore, M retrieves $\pi^{-1}(y)$ before making the hitting query.

1.3.1 Extending Gennaro and Trevisan's reconstruction lemma

Gennaro and Trevisan [16] presented a very elegant argument for proving that a random permutation is hard to invert also for non-uniform adversaries (previous proofs, e.g. [34], only ruled out uniform adversaries). Let A be a circuit and let π be a permutation that A inverts on a non-negligible fraction of its outputs. What Gennaro and Trevisan showed is that relative to A the permutation π has a relatively short description. Therefore, by a counting argument, there is only a tiny fraction of permutations which A inverts well. Intuitively, A saves on the description of π as it allows us to reconstruct π on (many of) the x's for which $A^{\pi}(\pi(x)) = x$. The formal proof strongly relies on a bound on the number of π gates in A: when we use A to reconstruct π on x we need all the π-queries made by $A^{\pi}(\pi(x))$ (apart perhaps of the query for $\pi(x)$ itself) to already be reconstructed.

In our setting, we would like to consider an adversary $A^{\mathsf{Sam}}(y)$ that (many times) inverts y without making Sam produce a y-hit. Recall that the oracle Sam is given as an input a circuit C with π-gates and has to produce a random inverse of some value z under the mapping defined by C. We would like to apply the argument of [16] to claim that relative to A and Sam there is a short description of π. However, we are faced with a substantial obstacle as the simulation of Sam requires making a huge amount of π queries.[6] Overcoming this obstacle requires much care both in the definition and analysis of Sam. We defer more details to Section 5.3.

1.4 Paper Organization

In Section 2, we briefly present the notations and formal definitions used in this paper and in Section 3 we describe the oracle that is used to derive our results. In Section 4, we show that this oracle can be used to break the security of statistical-hiding commitment schemes, and in Section 5 we show that every circuit which tries to invert a random permutation using this oracle (under some restrictions), fails with high probability. In Section 6, we combine the results of Sections 4 and 5, and derive our lower bound result. Finally, Section 7 discusses the implications of the result to other cryptographic protocols.

2 Preliminaries

We denote by Π_n the set of all permutations over $\{0,1\}^n$. For a finite set X, we denote by $x \leftarrow X$ the experiment of choosing an element of X according to the uniform distribution. Similarly, for a distribution $\mathcal{D}$ over a set X, we denote by $x \leftarrow \mathcal{D}$ the experiment of choosing an element of X according to the distribution $\mathcal{D}$. The statistical distance between two distributions X and Y over Ω is denoted $\mathrm{SD}(X,Y)$, and defined as

$$\mathrm{SD}(X,Y) = \frac{1}{2}\sum_{\omega\in\Omega} \left|\Pr_X[\omega] - \Pr_Y[\omega]\right| \quad .$$

The following standard fact (see, for example [56, Fact 2.6]) will be useful for us in analyzing statistically-close distributions.

[6] Consider for example C such that on input w it truncates the last bit of $\pi(w)$ and outputs the result. Finding collisions in C requires knowledge of π almost entirely.

Fact 2.1. *If X and Y are two distributions such that $\mathrm{SD}(X,Y) < \epsilon$, then with probability at least $1-2\sqrt{\epsilon}$ over $x \leftarrow X$ it holds that*

$$\left(1-\sqrt{\epsilon}\right)\cdot \Pr\left[X=x\right] < \Pr\left[Y=x\right] < \left(1+\sqrt{\epsilon}\right)\cdot \Pr\left[X=x\right] \ .$$

2.1 One-Way Permutations and Trapdoor Permutations

We briefly present the notions of one-way permutations and trapdoor (one-way) permutations which are used in this paper. For a more comprehensive discussion we refer the reader to [19].

Definition 2.2. A collection of permutations $\pi = \{\pi_n\}_{n=1}^{\infty}$, where $\pi_n \in \Pi_n$ for every n, is $s(n)$-*hard* if for every probabilistic Turing-machine A that runs in time $s(n)$, and for all sufficiently large n,

$$\Pr\left[A(1^n, y) = \pi_n^{-1}(y)\right] \leq \frac{1}{s(n)} \ ,$$

where the probability is taken uniformly over all the possible choices of $y \in \{0,1\}^n$ and over all the possible outcomes of the internal coin tosses of A.

In our setting, whenever such a collection π is given as an oracle, we denote by A^{π} a circuit or a Turing-machine A with oracle access to π. In addition, when we consider the probability of an event over the choice of π, we mean that for every integer n, a permutation π_n is chosen uniformly at random from Π_n and independently of all other permutations.

A collection of trapdoor permutations is represented as a triplet $\tau = \left(G, F, F^{-1}\right)$. Informally, G corresponds to a key generation procedure, which is queried on a string td (intended as the "trapdoor") and produces a corresponding public key pk. The procedure F is the actual permutation, which is queried on a public key pk and an input x. Finally, the procedure F^{-1} is the inverse of F: If $G(td) = pk$ and $F(pk, x) = y$, then $F^{-1}(td, y) = x$. In this paper, since we are concerned with providing a lower bound, we do not consider the most general definition of a collection of trapdoor permutations. Instead, we denote by T_n the set of all triplets $\tau_n = \left(G_n, F_n, F_n^{-1}\right)$ of the following form:

1. $G_n \in \Pi_n$.
2. $F_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is a function such that $F_n(pk, \cdot) \in \Pi_n$ for every $pk \in \{0,1\}^n$.
3. $F_n^{-1} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is a function such that $F_n^{-1}(td, y)$ returns the unique $x \in \{0,1\}^n$ for which $F_n(G_n(td), x) = y$.

Our lower bound proof is based on analyzing random instances of such collections. A uniformly distributed $\tau_n \in T_n$ can be chosen as follows: G_n is chosen uniformly at random from Π_n, and for each $pk \in \{0,1\}^n$ a permutation $F_n(pk, \cdot)$ is chosen uniformly and independently at random from Π_n. As above, we do not consider a single collection τ_n: we consider a family $\tau = \{\tau_n\}_{n=1}^{\infty}$ of collection of trapdoor permutations where $\tau_n \in T_n$ for every n. Whenever such a family τ is given as an oracle, we denote by A^{τ} a circuit or a Turing-machine A with oracle access to τ. In addition, when we consider the probability of an event over the choice of τ, we mean that for every integer n, a collection of trapdoor permutation τ_n is chosen uniformly at random from T_n and independently of all other collections.

Definition 2.3. A family of trapdoor permutations $\tau = \left\{\tau_n = \left(G_n, F_n, F_n^{-1}\right)\right\}_{n=1}^{\infty}$ is $s(n)$-*hard* if for every probabilistic Turing-machine A that runs in time $s(n)$, and for all sufficiently large n,

$$\Pr\left[A^{\tau}(1^n, G_n(td), y) = F_n^{-1}(td, y)\right] \leq \frac{1}{s(n)} \ ,$$

where the probability is taken uniformly over all the possible choices of $td \in \{0,1\}^n$ and $y \in \{0,1\}^n$, and over all the possible outcomes of the internal coin tosses of A.

Note that Definition 2.3 refers to the difficulty of inverting a random permutation $F(pk, \cdot)$ on a uniformly distributed image y, when given only $pk = G(td)$ and y. Some applications, however, require enhanced hardness conditions. For example, it may be required (cf. [20, Appendix C]) that it is hard to invert $F(pk, \cdot)$ on y even given the random coins used in the generation of y. Note that our formulation captures such hardness condition as well and therefore the impossibility results proved in this paper hold also for enhanced trapdoor permutations.[7]

2.2 Commitment Schemes

A commitment scheme is a two-stage interactive protocol between a sender and a receiver. Informally, after the first stage of the protocol, which is referred to as the *commit stage*, the sender is bound to at most one value, not yet revealed to the receiver. In the second stage, which is referred to as the *reveal stage*, the sender reveals its committed value to the receiver. In this paper, where we are interested in proving an impossibility result for commitment schemes, it will be sufficient for us to deal with bit-commitment schemes, i.e., commitment schemes in which the committed value is only one bit. More formally, a bit-commitment scheme is defined via a triplet of probabilistic polynomial-time Turing-machines $(\mathcal{S}, \mathcal{R}, \mathcal{V})$ such that:

- $\mathcal{S}$ receives as input the security parameter 1^n and a bit b. Following its interaction, it outputs some information decom (the decommitment).
- $\mathcal{R}$ receives as input the security parameter 1^n. Following its interaction, it outputs a state information com (the commitment).
- $\mathcal{V}$ (acting as the receiver in the reveal stage[8]) receives as input the security parameter 1^n, a commitment com and a decommitment decom. It outputs either a bit b' or $\perp$.

Denote by $(\mathsf{decom}|\mathsf{com}) \leftarrow \langle \mathcal{S}(1^n, b), \mathcal{R}(1^n)\rangle$ the experiment in which $\mathcal{S}$ and $\mathcal{R}$ interact (using the given inputs and uniformly chosen random coins), and then $\mathcal{S}$ outputs decom while $\mathcal{R}$ outputs com. It is required that for all n, every bit b, and every pair $(\mathsf{decom}|\mathsf{com})$ that may be output by $\langle \mathcal{S}(1^n, b), \mathcal{R}(1^n)\rangle$, it holds that $\mathcal{V}(\mathsf{com}, \mathsf{decom}) = b$.[9]

The security of a commitment scheme can be defined in two complementary ways, protecting against either an all-powerful sender or an all-powerful receiver. In this paper, we deal with commitment schemes of the latter type, which are referred to as *statistically-hiding* commitment schemes. In order to define the security properties of such schemes, we first introduce the following notation. Given a commitment scheme $(\mathcal{S}, \mathcal{R}, \mathcal{V})$ and a Turing-machine $\mathcal{R}^*$, we denote by $\mathsf{view}_{\langle \mathcal{S}(b), \mathcal{R}^*\rangle}(n)$ the distribution on the view of $\mathcal{R}^*$ when interacting with $\mathcal{S}(1^n, b)$. This view consists of $\mathcal{R}^*$'s random coins and of the sequence of messages it receives from $\mathcal{S}$. The distribution is taken over the random coins of both $\mathcal{S}$ and $\mathcal{R}$. Note that whenever no computational restrictions are assumed on $\mathcal{R}^*$, without loss of generality $\mathcal{R}^*$ is deterministic.

[7] A different enhancement, used by [28], requires the permutations' domain to be polynomially dense in $\{0,1\}^n$. Clearly, our impossibility result holds w.r.t. this enhancement as well.

[8] Note that there is no loss of generality in assuming that the reveal stage is non-interactive. This is since any such interactive stage can be replaced with a non-interactive one as follows: The sender sends its internal state to the receiver, who then simulates the sender in the interactive stage.

[9] Although we assume perfect completeness, it is not essential for our results.

Definition 2.4. A bit-commitment scheme $(\mathcal{S}, \mathcal{R}, \mathcal{V})$ is *$\rho(n)$-hiding* if for every deterministic Turing-machine $\mathcal{R}^*$ the ensembles $\{\mathsf{view}_{\langle \mathcal{S}(0), \mathcal{R}^* \rangle}(n)\}$ and $\{\mathsf{view}_{\langle \mathcal{S}(1), \mathcal{R}^* \rangle}(n)\}$ have statistical difference at most $\rho(n)$ for all sufficiently large n. Such a scheme is *statistically-hiding* if it is $\rho(n)$-hiding for some negligible function $\rho(n)$.

In the paper, we consider a weaker hiding requirement than stated in the above definition. Our lower bound holds even for commitment schemes that guarantee the security of the sender only when interacting with the honest receiver. This is indeed a weaker requirement, and therefore it only strengthens our lower bound result.

Definition 2.5. A bit-commitment scheme $(\mathcal{S}, \mathcal{R}, \mathcal{V})$ is *honest-receiver $\rho(n)$-hiding* if the ensembles $\{\mathsf{view}_{\langle \mathcal{S}(0), \mathcal{R} \rangle}(n)\}$ and $\{\mathsf{view}_{\langle \mathcal{S}(1), \mathcal{R} \rangle}(n)\}$ have statistical difference at most $\rho(n)$ for all sufficiently large n. Such a scheme is *honest-receiver statistically-hiding* if it is honest-receiver $\rho(n)$-hiding for some negligible function $\rho(n)$.

Definition 2.6. A bit-commitment scheme $(\mathcal{S}, \mathcal{R}, \mathcal{V})$ is *$\mu(n)$-binding* if for every probabilistic polynomial-time Turing-machine $\mathcal{S}^*$ it holds that

$$\Pr\left[((\mathsf{decom}, \mathsf{decom}')|\mathsf{com}) \leftarrow \langle \mathcal{S}^*(1^n), \mathcal{R}(1^n) \rangle : \begin{array}{c} \mathcal{V}(\mathsf{com}, \mathsf{decom}) = 0 \\ \mathcal{V}(\mathsf{com}, \mathsf{decom}') = 1 \end{array}\right] < \mu(n)$$

for all sufficiently large n, where the probability is taken over the random coins of both $\mathcal{S}^*$ and $\mathcal{R}$. Such a scheme is *computationally-binding* if it is $\mu(n)$-binding for some negligible function $\mu(n)$, and is *weakly-binding* if it is $(1 - 1/p(n))$-binding for some polynomial $p(n)$.

2.3 Black-Box Reductions

A reduction of a primitive P to a primitive Q is a construction of P out of Q. Such a construction consists of showing that if there exists an implementation C of Q, then there exists an implementation M_C of P. This is equivalent to showing that for every adversary that breaks M_C, there exists an adversary that breaks C. Such a reduction is *semi-black-box* if it ignores the internal structure of Q's implementation, and it is *fully-black-box* if the proof of correctness is black-box as well, i.e., the adversary for breaking Q ignores the internal structure of both Q's implementation and of the (alleged) adversary breaking P. Semi-black-box reductions are less restricted and thus more powerful than fully-black-box reductions. A taxonomy of black-box reductions was provided by Reingold, Trevisan and Vadhan [52], and the reader is referred to their paper for a more complete and formal view of these notions.

We now formally define the class of constructions considered in this paper. Our main result is concerned with the particular setting of fully-black-box constructions of weakly-binding statistically-hiding commitment schemes from trapdoor permutations. We focus here on a specific definition for these particular primitives and we refer the reader to [52] for a more general definition.

Definition 2.7. A fully-black-box construction of a weakly-binding honest-receiver statistically-hiding commitment scheme from an $s(n)$-hard family of trapdoor permutations is a quadruple of probabilistic oracle Turing-machines $(\mathcal{S}, \mathcal{R}, \mathcal{V}, A)$ for which the following hold:

1. **Correctness:** For every family τ of trapdoor permutations, $(\mathcal{S}^\tau, \mathcal{R}^\tau, \mathcal{V}^\tau)$ is an honest-receiver statistically-hiding commitment scheme.
2. **Black-box proof of binding:** For every family $\tau = \left\{\tau_n = \left(G_n, F_n, F_n^{-1}\right)\right\}_{n=1}^{\infty}$ of trapdoor permutations and for every probabilistic polynomial-time Turing-machine $\mathcal{S}^*$, if $\mathcal{S}^*$ with oracle

access to τ breaks the weak binding of $(\mathcal{S}^\tau, \mathcal{R}^\tau, \mathcal{V}^\tau)$, then

$$\Pr\left[A^{\tau,\mathcal{S}^*}(1^n, G_n(td), y) = F_n^{-1}(td, y)\right] > \frac{1}{s(n)} \ ,$$

for infinitely many values of n, where A runs in time $s(n)$, and the probability is taken uniformly over all the possible choices of $td \in \{0,1\}^n$ and $y \in \{0,1\}^n$, and over all the possible outcomes of the internal coin tosses of A.

We remark that the above correctness requirement is very strict and is not essential for our results. In fact, as will become clear later on (in Section 4), for every τ such that the protocol $(\mathcal{S}^\tau, \mathcal{R}^\tau, \mathcal{V}^\tau)$ is a statistically-hiding commitment scheme, we construct a malicious sender $\mathcal{S}^*$ which breaks the binding property of the scheme. Therefore, we could have dealt with a weaker correctness requirement as well, but stating such a weaker requirement in a meaningful way turns out to be quite subtle.

In addition, it would be useful for us to consider the following property of fully-black-box constructions: Consider a malicious sender $\mathcal{S}^*$ that breaks the binding of the commitment scheme and consider the machine A that wishes to break the security of the trapdoor permutation. Then, A receives a security parameter 1^n and invokes $\mathcal{S}^*$ in a black-box manner. Definition 2.7, however, does not restrict the range of security parameters that A is allowed to invoke $\mathcal{S}^*$ on. For example, A may invoke $\mathcal{S}^*$ on security parameter 1^{n^2}, or even on security parameter $1^{\Theta(s(n))}$, where $s(n)$ is the running time of A. The following definition will enable us to capture this property of the construction, and again, we present a specific definition for our setting.

Definition 2.8. A fully-black-box construction $(\mathcal{S}, \mathcal{R}, \mathcal{V}, A)$ is $\ell(n)$*-security-parameter-expanding*, if for every malicious sender $\mathcal{S}^*$, the machine A on security parameter 1^n invokes $\mathcal{S}^*$ on security parameters which are at most $1^{\ell(n)}$.

3 The Oracle

In this section we describe the oracle that will imply our lower bound results. Our oracle $\mathcal{O}$ is of the form $(\tau, \mathsf{Sam}^\tau)$, where τ is a family of trapdoor permutations (i.e., $\tau = \{\tau_n\}_{n=1}^\infty$, where $\tau_n \in T_n$ for every n), and Sam^τ is an oracle that, very informally, receives as input a description of a circuit C (which may contain τ-gates) and a string z, and outputs a uniformly distributed preimage of z under the mapping defined by C. As discussed in the introduction, we will impose several essential restrictions on the querying of Sam that will prevent it from assisting in inverting τ.

Description of Sam. The oracle Sam receives as input a query $Q = (C^\tau_{\text{next}}, C^\tau, z)$, and outputs a pair (w', z') where w' is a uniformly distributed preimage of z under the mapping defined by the circuit C^τ, and $z' = C^\tau_{\text{next}}(w')$. We impose the following restrictions:

1. z was the result of a previous query with C^τ as the next-query circuit (note that this imposes a forest-like structure on the queries).

2. The circuit C^τ_{next} is a *refinement* of the circuit C^τ, where by a refinement we mean that $C^\tau_{\text{next}}(w) = (C^\tau(w), \widetilde{C}^\tau(w))$ for some circuit $\widetilde{C}^\tau$ and for every w. In particular, this implies that C^τ and C^τ_{next} have the same input length. Given a query Q, we denote this input length by $m(Q)$, and when the query Q is clear from the context we will write only m.

3. Each query contains a security parameter 1^n, and Sam answers queries only up to depth $\mathsf{depth}(n)$, for some "depth restriction" function $\mathsf{depth} : \mathbb{N} \to \mathbb{N}$ which is part of the description of Sam. The security parameter is set such that a query with security parameter 1^n is allowed to contain circuits with queries to permutations on up to n bits. Note that although different queries may have different security parameters, we ask that in the same "query-tree", all queries will have the same security parameter (hence the depth of the tree is already determined by the root query).

In order to impose these restrictions, we equip Sam with a family $\mathsf{sign} = \{\mathsf{sign}_k\}_{k=1}^{\infty}$ of (random) functions $\mathsf{sign}_k : \{0,1\}^k \to \{0,1\}^{2k}$ that will be used as "signatures" for identifying legal queries as follows: in addition to outputting (w', z'), Sam will also output the value $\mathsf{sign}(1^n, C^\tau_{\text{next}}, z', dep+1)$, where dep is the depth of the query, 1^n is the security parameter of the query, and by applying the "function" sign we actually mean that we apply the function sign_k for the correct input length. Each query of the form $Q = (1^n, C^\tau_{\text{next}}, C^\tau, z, dep, sig)$ is answered by Sam if and only if C^τ_{next} is a refinement of C^τ, $dep \leq \mathsf{depth}(n)$ and $sig = \mathsf{sign}(1^n, C^\tau, z, dep)$.

Finally, we provide Sam with a family of (random) permutations $\mathcal{F} = \{f_Q\}$, where for every possible query Q a permutation f_Q is chosen uniformly at random from $\Pi_{m(Q)}$. Given a query $Q = (1^n, C^\tau_{\text{next}}, C^\tau, z, dep, sig)$, the oracle Sam uses the permutation $f_Q \in \mathcal{F}$ in order to sample w' as follows: it outputs $w' = f_Q(t)$ for the lexicographically smallest $t \in \{0,1\}^m$ such that $C^\tau(f_Q(t)) = z$. Note that whenever the permutation f_Q is chosen from Π_m uniformly at random, and independently of all other permutations in $\mathcal{F}$, then w' is indeed a uniformly distributed preimage of z. In this paper, whenever we consider the probability of an event over the choice of the family $\mathcal{F}$, we mean that for each query Q a permutation f_Q is chosen uniformly at random from $\Pi_{m(Q)}$ and independently of all other permutations. A complete and formal description of the oracle is provided in Figure 1.

On input $Q = (1^n, C^\tau_{\text{next}}, C^\tau, z, dep, sig)$, the oracle $\mathsf{Sam}^{\tau,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}$ acts as follows:

1. If $C^\tau = \perp$, then output (w', z', sig') where $w' = f_Q(0^m)$, $z' = C^\tau_{\text{next}}(w')$, and $sig' = \mathsf{sign}(1^n, C^\tau_{\text{next}}, z', 1)$.
2. Else, if C^τ_{next} is a refinement of C^τ, $dep \leq \mathsf{depth}(n)$ and $sig = \mathsf{sign}(1^n, C^\tau, z, dep)$, then
   (a) Find the lexicographically smallest $t \in \{0,1\}^m$ such that $C^\tau(f_Q(t)) = z$.
   (b) Output (w', z', sig') where $w' = f_Q(t)$, $z' = C^\tau_{\text{next}}(w')$, and $sig' = \mathsf{sign}(1^n, C^\tau_{\text{next}}, z', dep+1)$.
3. Else, output $\perp$.

Figure 1: The oracle Sam.

As mentioned above, the restrictions impose a forest-like structure on any sequence of queries: each query of the form $Q = (1^n, C^\tau_{\text{next}}, \perp, \perp, \perp, \perp)$ serves as a root of a tree. For any other "legal" query $Q = (1^n, C^\tau_{\text{next}}, C^\tau, z, dep, sig)$, there exists a previous query Q' which resulted in output z and contained C^τ as its next-query circuit. The query Q' is identified as the parent of Q in the query forest and is denoted $Q' = \mathrm{p}(Q)$. If there is more than one such Q', then we choose the first Q' according to some fixed ordering of the queries. When dealing with Turing-machines, we can identify the queries according to their chronological order.[10]

Notation 3.1. We say that a circuit A queries the oracle $\mathsf{Sam}^{\tau,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}$ up to depth d, if for every Sam-query $Q = (1^n, C^\pi_{\text{next}}, C^\pi, z, dep, sig)$ that A makes, it holds that $dep \leq d$.

[10]However, when dealing with circuits we will have to identify the queries according to a some topological order which is consistent with their forest structure. As Lemma 3.2 below indicates, such an ordering implies that for every two queries Q_i and Q_j (with a sufficiently large security parameter) such that $Q_i = \mathrm{p}(Q_j)$, it holds that $i < j$.

Simplifying the notation. In the remainder of this paper we often ignore both the depth restriction function depth and the security parameters, but we still keep in mind that the oracle Sam is defined with a restriction on its query depth.

Imposing legal queries. Recall that we equip the oracle Sam with a family sign of functions $\mathsf{sign}_k : \{0,1\}^k \to \{0,1\}^{2k}$ that we claimed can be used for identifying legal queries. As indicated by Figure 1, we say that a circuit A produces an illegal k-bit Sam-query if it queries Sam with some $Q = (1^n, C^\tau_{\text{next}}, C^\tau, z, dep, sig)$ for which $C^\tau \neq \perp$ and $sig = \mathsf{sign}_k(1^n, C^\tau, z, dep)$, but the value sig was not given to A as an answer to a previous Sam-query.[11] Since access to the family sign is only through Sam, in order to produce an illegal query, one must guess the value of $\mathsf{sign}_k(v)$ for some $v \in \{0,1\}^k$ *before* Sam queried sign_k on v. Lemma 3.2 below justifies our assumption that no "illegal" Sam-queries are made. We denote by sign_{-k} the family sign where the k-th function sign_k is left undefined (and in this case we will consider the process of choosing the function sign_k uniformly at random), and prove the following lemma:

Lemma 3.2. *For every k, τ, $\mathcal{F}$, depth, sign_{-k} and circuit A of size s, the probability over the random choice of the function $\mathsf{sign}_k : \{0,1\}^k \to \{0,1\}^{2k}$ that the circuit A with oracle access to $\mathcal{O} = \left(\tau, \mathsf{Sam}^{\tau,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}\right)$ produces an illegal k-bit Sam-query is at most $s/2^k$.*

Proof. Fix k, τ, $\mathcal{F}$, depth, sign_{-k} and a circuit A of size s. Denote by Γ the set of all functions sign_k for which A with oracle access to $\mathcal{O} = \left(\tau, \mathsf{Sam}^{\tau,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}\right)$ produces an illegal Sam-query. Then, there exists a set $\Gamma' \subset \Gamma$ of size at least $|\Gamma|/s$ and an integer $1 \leq i \leq s$, such that for all functions $\mathsf{sign}_k \in \Gamma'$ we have that that i-th Sam-query that A makes is illegal, while all the previous $i-1$ queries are legal. We claim that every function $\mathsf{sign}_k \in \Gamma'$ can be described using $k \cdot (2^{k+1} - 1)$ bits, given i, τ, $\mathcal{F}$, depth, sign_{-k} and A. More specifically, given a function $\mathsf{sign}_k \in \Gamma'$, denote by $v \in \{0,1\}^k$ the value on which A guesses the correct value of $\mathsf{sign}_k(v)$ for producing the illegal query. We can describe the function sign_k by specifying its value on the set $\{0,1\}^k \setminus \{v\}$ and by specifying v. This results in $2k \cdot (2^k - 1) + k = k \cdot (2^{k+1} - 1)$ bits. Indeed, the value $\mathsf{sign}_k(v)$ can be reconstructed by following the computation of $A^{\mathcal{O}}$, answering Sam's sign-queries in the first $i-1$ queries of A, and the i-th query will contain the value $\mathsf{sign}_k(v)$. Since we are guaranteed that the first $i-1$ queries are legal, Sam does not query sign_k on v and the simulation is successful. Therefore, the value $\mathsf{sign}_k(v)$ can be reconstructed.

This implies that the cardinality of the set Γ' is at most $2^{k\cdot(2^{k+1}-1)}$, and therefore the cardinality of Γ is at most $s \cdot 2^{k\cdot(2^{k+1}-1)}$. This means that the fraction of functions $\mathsf{sign}_k : \{0,1\}^k \to \{0,1\}^{2k}$ for which A produces an illegal query is at most

$$\frac{s \cdot 2^{k\cdot(2^{k+1}-1)}}{2^{2k\cdot 2^k}} = \frac{s}{2^k} \ .$$

■

4 Breaking Statistical Commitment Schemes With Sam

In this section we show that a random instance of the oracle Sam can be used to break the binding of any statistically-hiding commitment scheme. More specifically, for every statistically-hiding commitment scheme $(\mathcal{S}, \mathcal{R}, \mathcal{V})$ with oracle access to a family of trapdoor permutations, we construct a

[11]Note that we denoted by k not the actual bit-length of the query Q, but only the bit-length of the part of Q on which sign is applied. That is, k is the bit-length of the string $(1^n, C^\tau, z, dep)$, and in particular $k \geq n$.

malicious sender $\mathcal{S}^*$ which has oracle access to $\mathsf{Sam}^{\tau,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}$, and breaks the binding of $(\mathcal{S}^\tau, \mathcal{R}^\tau, \mathcal{V}^\tau)$ with high probability over the choices of τ, $\mathcal{F}$ and sign. The key point is that if the commitment scheme has $d(n)$ communication rounds, then $\mathcal{S}^*$ needs to query Sam only up to depth $d(n)+1$. Formally, the following theorem is proved.

Theorem 4.1. *For every $d(n)$-round honest-receiver statistically-hiding bit-commitment scheme $(\mathcal{S}, \mathcal{R}, \mathcal{V})$ with oracle access to a family of trapdoor permutations, there exist a polynomial-time malicious sender $\mathcal{S}^*$ and a negligible function $\nu(n)$, such that*

$$\Pr_{\tau,\mathcal{F},\mathsf{sign},r_{\mathcal{R}}}\left[\begin{array}{r}((\mathsf{decom},\mathsf{decom}')|\mathsf{com}) \leftarrow \left\langle \mathcal{S}^{*\ \mathsf{Sam}^{\tau,\mathcal{F},\mathsf{sign}}_{d+1}}(1^n), \mathcal{R}^\tau(1^n, r_{\mathcal{R}})\right\rangle : \\ \mathcal{V}^\tau(\mathsf{com},\mathsf{decom}) = 0, \mathcal{V}^\tau(\mathsf{com},\mathsf{decom}') = 1\end{array}\right] > 1 - \nu(n) \ ,$$

for all sufficiently large n.

Note that in the above theorem, the depth restriction function $\mathsf{depth}(n)$ of the oracle Sam is set to be the function $d(n)+1$, where $d(n)$ is the number of communication rounds in the commitment scheme $(\mathcal{S}, \mathcal{R}, \mathcal{V})$ with security parameter 1^n. This way, Sam will answer queries up to depth $d(n)+1$. In what follows, we define the notation used in this section. Then, we describe the malicious sender $\mathcal{S}^*$ and turn to prove Theorem 4.1.

Notations. Let $(\mathcal{S}, \mathcal{R}, \mathcal{V})$ be a statistically-hiding bit-commitment scheme with oracle access to a collection of trapdoor permutations. We denote by $b \in \{0,1\}$ and $r_{\mathcal{S}}, r_{\mathcal{R}} \in \{0,1\}^*$ the input bit of the sender and the random coins of the sender and the receiver, respectively. We denote by $d(n)$ the number of communication rounds in the scheme with security parameter 1^n (note that we do not restrict the scheme to access only trapdoor permutations over n-bits), and without loss of generality we assume that the receiver makes the first move. Each communication round consists of a message sent from the receiver to the sender followed by a message sent from the sender to the receiver. We denote by q_i and a_i the messages sent by the receiver and the sender in the i-th round, respectively, and denote by a_{d+1} the message sent by the sender in the reveal stage. Finally, we let $\bar{a}_i = (a_1, \ldots, a_i)$ and $\bar{q}_i = (q_1, \ldots, q_i)$. A d-round bit-commitment scheme is described in Figure 2.

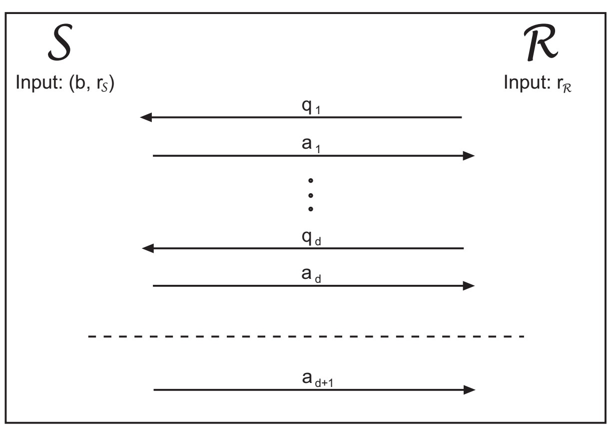

Figure 2: A d-round bit-commitment scheme.

Although the sender is a probabilistic polynomial-time Turing-machine, in order to interact with the oracle Sam we need to identify the sender with a sequence of circuits $S_1, \ldots, S_{d+1}$ as follows. In the first round, $\mathcal{S}$ sends a_1 by computing $a_1 = S_1(b, r_{\mathcal{S}}, q_1)$. Similarly, in the following rounds, $\mathcal{S}$ sends a_i by computing $a_i = S_i(b, r_{\mathcal{S}}, \bar{q}_i)$. We assume that each message a_i contains all of the sender's previous messages $a_1, \ldots a_{i-1}$ as well (i.e., in the i-th round the $\mathcal{S}$ sends actually $\bar{a}_i$), and therefore each circuit S_i is a *refinement* of S_{i-1}, as discussed in Section 3). We note that the descriptions of the circuits $S_1, \ldots, S_{d+1}$ can be computed in polynomial-time given a description of $\mathcal{S}$.

Finally, in order to simplify the notation regarding the input and output of the oracle Sam, in this section we ignore parts of the input and output of Sam: we ignore the security parameter and signature function sign, and note that Theorem 4.1 actually holds for every fixing of sign (since the malicious sender $\mathcal{S}^*$ asks only legal queries). In addition, we consider queries of the form $Q = (C^\tau_{\mathrm{next}}, C^\tau, z)$, and answers that consist only of w', i.e., an answer consists only of a uniformly distributed preimage of z under the mapping defined by C^τ.

Description of $\mathcal{S}^*$. On input 1^n, the malicious sender $\mathcal{S}^*$ with oracle access to $\mathsf{Sam}^{\tau,\mathcal{F}}_{d+1}$ interacts with the honest receiver $\mathcal{R}$ as follows.

1. In the first round, $\mathcal{S}^*$ receives $\mathcal{R}$'s message q_1, and computes the description of the circuit $C_1 = S_1(\cdot, \cdot, q_1)$ obtained from the circuit S_1 by fixing q_1 as its third input. Then, $\mathcal{S}^*$ queries $\mathsf{Sam}^{\tau,\mathcal{F}}_{d+1}$ with $(C_1, \perp, \perp)$, receives $w_1 = (b_1, r_1)$, and sends $a_1 = S_1(b_1, r_1, q_1)$ to $\mathcal{R}$.

2. For every $2 \leq i \leq d(n)$, in the i-th round $\mathcal{S}^*$ receives $\mathcal{R}$'s message q_i, and computes the description of the circuit $C_i = S_i(\cdot, \cdot, \bar{q}_i)$ obtained from S_i by fixing the vector $\bar{q}_i$ as its third input. $\mathcal{S}^*$ queries $\mathsf{Sam}^{\tau,\mathcal{F}}_{d+1}$ with $(C_i, C_{i-1}, \bar{a}_{i-1})$, and receives $w_i = (b_i, r_i)$. Then, $\mathcal{S}^*$ sends $a_i = S_i(b_i, r_i, \bar{q}_i)$ to $\mathcal{R}$.

3. In the reveal stage, $\mathcal{S}^*$ queries $\mathsf{Sam}^{\tau,\mathcal{F}}_{d+1}$ with $(\perp, C_d, \bar{a}_d)$ for n times, and receives n pairs $\left\{\left(b^{(j)}_{d+1}, r^{(j)}_{d+1}\right)\right\}_{j=1}^{n}$. If there exist $j_0, j_1 \in [n]$ such that $b^{(j_0)}_{d+1} = 0$ and $b^{(j_1)}_{d+1} = 1$, then $\mathcal{S}^*$ outputs $\mathsf{decom} = S_{d+1}\left(b^{(j_0)}_{d+1}, r^{(j_0)}_{d+1}, \bar{q}_d\right)$ and $\mathsf{decom}' = S_{d+1}\left(b^{(j_1)}_{d+1}, r^{(j_1)}_{d+1}, \bar{q}_d\right)$.

A minor technical detail in step 3 is that the first parameter in each of the n queries made in the reveal stage should be a distinct circuit instead of $\perp$. This guarantees that the answers returned by Sam in the reveal stage are independent (otherwise, Sam will return the exact same answer n times). Any fixed sequence of n distinct circuits may be used. In addition, notice that $\mathcal{S}^*$ queries Sam up to depth $d(n)+1$, as allowed by the depth restriction function $d(n)+1$.

The two main ideas underlying the proof are the following:

1. The distribution of the protocol's transcript when executed with $\mathcal{S}^*$ and an honest receiver is identical to the distribution of the protocol's transcript when both parties are honest.

2. The assumption that the commitment scheme is honest-receiver statistically-hiding implies that a random transcript can be revealed both as a commitment to $b = 0$ and as a commitment to $b = 1$, with almost equal probabilities.

More specifically, we define two distributions:

- $\mathcal{D}^*_n = \mathsf{view}_{\langle \mathcal{S}^*, \mathcal{R} \rangle}(n)$ is the distribution of the view of $\mathcal{R}$ in the commit stage when interacting with the malicious sender $\mathcal{S}^*$. This view consists of $\mathcal{R}$'s random coins and of the sequence of messages it receives from $\mathcal{S}^*$. The distribution is taken over $\mathcal{R}$'s random coins and over the uniform choice of τ and $\mathcal{F}$.

- $\mathcal{D}_n = \mathsf{view}_{\langle \mathcal{S}, \mathcal{R} \rangle}(n)$ is the distribution of the view of $\mathcal{R}$ in the commit stage when interacting with the honest sender $\mathcal{S}(1^n, b, r_{\mathcal{S}})$. This view consists of $\mathcal{R}$'s random coins and of the sequence of messages it receives from $\mathcal{S}$. The distribution is taken over the random coins of $\mathcal{R}$ and $\mathcal{S}$, and over the uniform choice of $b \in \{0, 1\}$ and τ.

Lemma 4.2. *The distributions $\mathcal{D}_n$ and $\mathcal{D}_n^*$ are identical.*

Proof. We show that the distributions $\mathcal{D}_n$ and $\mathcal{D}_n^*$ assign equal probabilities to every triplet $(r_\mathcal{R}, \bar{q}_d, \bar{a}_d)$. More specifically, we prove by induction on $1 \leq i \leq d$ that $\Pr_{\mathcal{D}_n}[r_\mathcal{R}, \bar{q}_d, \bar{a}_d] = \Pr_{\mathcal{D}_n^*}[r_\mathcal{R}, \bar{q}_d, \bar{a}_d]$.

For $i = 1$, clearly we have that $\Pr_{\mathcal{D}_n}[r_\mathcal{R}, q_1] = \Pr_{\mathcal{D}_n^*}[r_\mathcal{R}, q_1]$ since $r_\mathcal{R}$ is distributed exactly the same in the two cases, and q_1 is a deterministic function of $r_\mathcal{R}$. Therefore we only have to show that $\Pr_{\mathcal{D}_n}[a_1 | r_\mathcal{R}, q_1] = \Pr_{\mathcal{D}_n^*}[a_1 | r_\mathcal{R}, q_1]$. In the first round, the malicious sender $\mathcal{S}^*$ queries $\mathsf{Sam}_{d+1}^{\tau,\mathcal{F}}$ with $Q = (C_1, \perp, \perp)$, and receives $w_1 = (b_1, r_1)$. Note that by the description of $\mathsf{Sam}_{d+1}^{\tau,\mathcal{F}}$ and of $\mathcal{F}$, there is a random permutation f_Q which corresponds to Q, and $\mathsf{Sam}_{d+1}^{\tau,\mathcal{F}}$ outputs $(b_1, r_1) = f_Q(0^m)$, which is a uniformly distributed value. That is, $\mathcal{S}^*$ sends $a_1 = S_1(b_1, r_1, q_1)$ for a uniformly distributed pair (b_1, r_1) exactly as the honest sender $\mathcal{S}$ should do.

Assume now that the claim holds for some i, i.e., $\Pr_{\mathcal{D}_n}[r_\mathcal{R}, \bar{q}_i, \bar{a}_i] = \Pr_{\mathcal{D}_n^*}[r_\mathcal{R}, \bar{q}_i, \bar{a}_i]$. Again, we have that $\Pr_{\mathcal{D}_n}[q_{i+1} | r_\mathcal{R}, \bar{q}_i, \bar{a}_i] = \Pr_{\mathcal{D}_n^*}[q_{i+1} | r_\mathcal{R}, \bar{q}_i, \bar{a}_i]$, since in both cases q_{i+1} is a deterministic function of $r_\mathcal{R}$, $\bar{q}_i$ and $\bar{a}_i$. It remains to show that $\Pr_{\mathcal{D}_n}[a_{i+1} | r_\mathcal{R}, \bar{q}_{i+1}, \bar{a}_i] = \Pr_{\mathcal{D}_n^*}[a_{i+1} | r_\mathcal{R}, \bar{q}_{i+1}, \bar{a}_i]$. In round $i+1$, $\mathcal{S}^*$ queries $\mathsf{Sam}_{d+1}^{\tau,\mathcal{F}}$ with $Q = (C_{i+1}, C_i, \bar{a}_i)$, and receives $w_{i+1} = (b_{i+1}, r_{i+1})$. Note that by the description of $\mathsf{Sam}^{\tau,\mathcal{F}}$ and of $\mathcal{F}$, the permutation f_Q which corresponds to Q was chosen uniformly at random from Π_m and independently of all the other permutations in $\mathcal{F}$. Therefore, (b_{i+1}, r_{i+1}) is uniformly distributed among all inputs which are consistent with the protocol's transcript until this point, and therefore the distribution of the resulting a_{i+1} is exactly as if the honest sender $\mathcal{S}$ had input (b_{i+1}, r_{i+1}) to begin with. Thus, $\Pr_{\mathcal{D}_n}[a_{i+1} | r_\mathcal{R}, \bar{q}_{i+1}, \bar{a}_i] = \Pr_{\mathcal{D}_n^*}[a_{i+1} | r_\mathcal{R}, \bar{q}_{i+1}, \bar{a}_i]$, which yields the correctness of the lemma. ■

Lemma 4.2 now enables us to derive the proof of Theorem 4.1.

Proof of Theorem 4.1. In the reveal stage, the malicious sender $\mathcal{S}^*$ uses $\mathsf{Sam}_{d+1}^{\tau,\mathcal{F}}$ in order to sample uniformly and independently at random n input pairs $\left\{\left(b_{d+1}^{(j)}, r_{d+1}^{(j)}\right)\right\}_{j=1}^{n}$ from the set of all input pairs which are consistent with the transcript of the commit stage. We prove that with overwhelming probability these inputs enable $\mathcal{S}^*$ to reveal both to $b = 0$ and to $b = 1$.

Denote by $\mathcal{D}_n^0 = \mathsf{view}_{\langle \mathcal{S}(0), \mathcal{R} \rangle}(n)$ the distribution of the honest receiver's view in the commit stage when interacting with the honest sender $\mathcal{S}(1^n, 0, r_\mathcal{S})$. This view consists of its random coins and of the sequence of messages it receives from $\mathcal{S}$, and the distribution is taken over the random coins of $\mathcal{R}$ and $\mathcal{S}$ and over the choice of τ. Similarly, let $\mathcal{D}_n^1 = \mathsf{view}_{\langle \mathcal{S}(1), \mathcal{R} \rangle}(n)$.

We define a set of *good* transcripts. This set consists of all transcripts of the commit stage which enable $\mathcal{S}^*$ to reveal both to $b = 0$ and to $b = 1$ with overwhelming probability. We show that with overwhelming probability the transcript is in this set. Denote by $\rho(n)$ the hiding parameter of the commitment scheme (see Definition 2.4, and recall that the commitment scheme is honest-receiver statistically-hiding, and therefore $\rho(n)$ is a negligible function). Formally, we define

$$\mathsf{GOOD} = \left\{ \mathsf{trans} : \left(1 - \sqrt{\rho(n)}\right) \cdot \Pr_{\mathcal{D}_n^0}[\mathsf{trans}] < \Pr_{\mathcal{D}_n^1}[\mathsf{trans}] < \left(1 + \sqrt{\rho(n)}\right) \cdot \Pr_{\mathcal{D}_n^0}[\mathsf{trans}] \right\} \ .$$

Note that for every transcript trans of the commit stage and for every $j \in [n]$, it holds that

$$\frac{\Pr_{\tau,\mathcal{F},r_\mathcal{R}}\left[b_{d+1}^{(j)} = 0 \ \middle| \ \mathsf{trans}\right]}{\Pr_{\tau,\mathcal{F},r_\mathcal{R}}\left[b_{d+1}^{(j)} = 1 \ \middle| \ \mathsf{trans}\right]} = \frac{\Pr_{\tau,\mathcal{F},r_\mathcal{R}}\left[b_{d+1}^{(j)} = 0 \ \wedge \ \mathsf{trans}\right]}{\Pr_{\tau,\mathcal{F},r_\mathcal{R}}\left[b_{d+1}^{(j)} = 1 \ \wedge \ \mathsf{trans}\right]} = \frac{\Pr_{\mathcal{D}_n^0}[\mathsf{trans}]}{\Pr_{\mathcal{D}_n^1}[\mathsf{trans}]} \ ,$$

where the second equality follows from Lemma 4.2. The definition of the set GOOD implies that if $\mathsf{trans} \in \mathsf{GOOD}$, then for all sufficiently large n it holds that

$$\min\left\{\Pr_{\tau,\mathcal{F},r_{\mathcal{R}}}\left[b_{d+1}^{(j)}=0 \;\middle|\; \mathsf{trans}\right], \Pr_{\tau,\mathcal{F},r_{\mathcal{R}}}\left[b_{d+1}^{(j)}=1 \;\middle|\; \mathsf{trans}\right]\right\} > 1/3 \enspace .$$

Therefore,

$$\Pr_{\tau,\mathcal{F},r_{\mathcal{R}}}\left[\mathcal{S}^* \text{ fails} \mid \mathsf{trans} \in \mathsf{GOOD}\right] < 2 \cdot \left(\frac{2}{3}\right)^n \enspace ,$$

since a failure occurs only in the case that all n input pairs sampled in the reveal stage have $b_{d+1}^{(j)} = 0$, or that they all have $b_{d+1}^{(j)} = 1$. It remains to show that the transcript is in GOOD with overwhelming probability. Lemma 4.2 and the fact that the statistical distance between the distributions $\mathcal{D}_n^0$ and $\mathcal{D}_n^1$ is at most $\rho(n)$ imply that

$$\begin{aligned}
\Pr_{\tau,\mathcal{F},r_{\mathcal{R}}}\left[\mathsf{trans} \in \mathsf{GOOD}\right] &= \Pr_{\mathcal{D}_n}\left[\mathsf{trans} \in \mathsf{GOOD}\right] \\
&= \frac{1}{2} \cdot \left(\Pr_{\mathcal{D}_n^0}\left[\mathsf{trans} \in \mathsf{GOOD}\right] + \Pr_{\mathcal{D}_n^1}\left[\mathsf{trans} \in \mathsf{GOOD}\right]\right) \\
&\geq \frac{1}{2} \cdot \left(2 \cdot \Pr_{\mathcal{D}_n^0}\left[\mathsf{trans} \in \mathsf{GOOD}\right] - \rho(n)\right) \\
&> 1 - 2\sqrt{\rho(n)} - \frac{\rho(n)}{2} \enspace ,
\end{aligned}$$

where the last inequality follows from Fact 2.1. We conclude the proof by

$$\begin{aligned}
\Pr_{\tau,\mathcal{F},r_{\mathcal{R}}}\left[\mathcal{S}^* \text{fails}\right] &\leq \Pr_{\tau,\mathcal{F},r_{\mathcal{R}}}\left[\mathsf{trans} \notin \mathsf{GOOD}\right] + \Pr_{\tau,\mathcal{F},r_{\mathcal{R}}}\left[\mathcal{S}^* \text{ fails} \mid \mathsf{trans} \in \mathsf{GOOD}\right] \\
&\leq 2\sqrt{\rho(n)} + \frac{\rho(n)}{2} + 2 \cdot \left(\frac{2}{3}\right)^n \enspace .
\end{aligned}$$

Therefore,

$$\Pr_{\tau,\mathcal{F},r_{\mathcal{R}}}\left[\begin{array}{r}((\mathsf{decom},\mathsf{decom}')|\mathsf{com}) \leftarrow \left\langle \mathcal{S}^{*\,\mathsf{Sam}_{d+1}^{\tau,\mathcal{F}}}(1^n), \mathcal{R}^{\tau}(1^n, r_{\mathcal{R}})\right\rangle : \\ \mathcal{V}^{\tau}(\mathsf{com},\mathsf{decom}) = 0, \mathcal{V}^{\tau}(\mathsf{com},\mathsf{decom}') = 1\end{array}\right] > 1 - \nu(n) \enspace ,$$

for all sufficiently large n, where $\nu(n) = 2\sqrt{\rho(n)} + \frac{\rho(n)}{2} + 2 \cdot \left(\frac{2}{3}\right)^n$. ∎

5 Random Permutations are Hard to Invert Even With Sam

We now prove our main technical result regarding the oracle Sam. For simplicity, we first consider the task of inverting a family of permutations, and then extend the result to the task of inverting a family of trapdoor permutations. We consider the oracle Sam exactly as defined in Section 3 with the only difference that the trapdoor permutation family τ is replaced with a permutation family π.

Our goal is to upper bound the success probability of circuits having oracle access to Sam in the task of inverting a uniformly chosen permutation $\pi_n \in \Pi_n$ on a uniformly chosen image $y \in \{0,1\}^n$ (i.e., the task of retrieving the value $\pi_n^{-1}(y)$ given y and oracle access to both π and Sam). Our contribution is in relating this success probability to the maximal depth of the Sam-queries made by the circuit, and to the size of the circuit. The following theorem is proved.

Theorem 5.1. *For every circuit A of size $s(n)$ that queries* Sam *up to depth $d(n)$ such that $s(n)^{3d(n)+2} < 2^{n/8}$, for every depth restriction function* depth *and for all sufficiently large n, it holds that*

$$\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y \leftarrow \{0,1\}^n}}\left[A^{\pi,\mathsf{Sam}_{\mathsf{depth}}^{\pi,\mathcal{F},\mathsf{sign}}}(y) = \pi_n^{-1}(y)\right] \leq \frac{1}{s(n)} \ .$$

Before turning to prove Theorem 5.1, we first provide a brief overview of the structure of the proof. Consider a circuit A which is given an input $y \in \{0,1\}^n$, and its goal is to retrieve the value $\pi_n^{-1}(y)$ while having oracle access to both $\pi = \{\pi_i\}_{i=1}^{\infty}$ and Sam. The idea underlying our proof is to distinguish between two cases: one in which A obtains information on the value $\pi_n^{-1}(y)$ via one of its Sam-queries and the other in which none of A's Sam-queries provides sufficient information for retrieving $\pi_n^{-1}(y)$. More specifically, we define:

Definition 5.2. A Sam-query $(1^\ell, C_{\text{next}}^\pi, C^\pi, z, dep, sig)$ produces a *y-hit* if Sam outputs (w', z', sig') such that some π_n-gate in the computation of $C^\pi(w')$ has input $\pi_n^{-1}(y)$.

Given π, $\mathcal{F}$, sign, depth, a circuit A and a challenge image $y \in \{0,1\}^n$, we denote by SamHIT_y the event in which one of the Sam-queries made by A in the computation of $A^{\pi,\mathsf{Sam}_{\mathsf{depth}}^{\pi,\mathcal{F},\mathsf{sign}}}(y)$ produces a y-hit. From this point on, the proof proceeds in two modular parts. In the first part of the proof, we consider the case that the event SamHIT_y does not occur, and prove a "reconstruction lemma" which extends an information-theoretic argument of Gennaro and Trevisan [16]. They showed that if a circuit A manages to invert a permutation π_n on a relatively large set of images, then this permutation has a short representation given A. We generalize their argument to deal with circuits having oracle access to Sam. In this part we do not restrict at all the depth of the Sam-queries and their security parameters, and prove the following lemma.

Lemma 5.3. *For every circuit A of size at most $2^{n/7}$, for every depth restriction function* depth *and for all sufficiently large n, it holds that*

$$\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y \leftarrow \{0,1\}^n}}\left[A^{\pi,\mathsf{Sam}_{\mathsf{depth}}^{\pi,\mathcal{F},\mathsf{sign}}}(y) = \pi_n^{-1}(y) \ \ \wedge \ \ \overline{\mathsf{SamHIT}_y}\right] \leq 2^{-n/8} \ .$$

In the second part of the proof, we show that the case where the event SamHIT_y does occur can be reduced to the case where the event SamHIT_y does not occur. In this proof, both the size of the circuit and the depth of its Sam-queries play an instrumental role. Specifically, given a circuit A that tries to invert a permutation π, we construct a circuit M that succeeds almost as well as A, without M's Sam-queries producing any y-hits. By analyzing the probabilistic process of the computation $A^{\pi,\mathsf{Sam}_{\mathsf{depth}}^{\pi,\mathcal{F},\mathsf{sign}}}(y)$ we prove the following lemma.

Lemma 5.4. *For every circuit A of size $s(n)$ that queries* Sam *up to depth $d(n)$, and for every depth restriction function* depth*, if*

$$\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y \leftarrow \{0,1\}^n}}\left[A^{\pi,\mathsf{Sam}_{\mathsf{depth}}^{\pi,\mathcal{F},\mathsf{sign}}}(y) = \pi_n^{-1}(y)\right] \geq \frac{1}{s(n)}$$

for infinitely many values of n, then there exists a circuit M of size $O(s(n))$ such that

$$\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y \leftarrow \{0,1\}^n}}\left[M^{\pi,\mathsf{Sam}_{\mathsf{depth}}^{\pi,\mathcal{F},\mathsf{sign}}}(y) = \pi_n^{-1}(y) \ \ \wedge \ \ \overline{\mathsf{SamHIT}_y}\right] \geq \frac{1}{s(n)^{3d(n)+2}}$$

for infinitely many values of n.

In what follows we show that Theorem 5.1 is a straightforward corollary of Lemmata 5.3 and 5.4. In Subsection 5.1 we extend our statement to deal with trapdoor permutations, and this form of the result will be used in the lower bound proof in Section 6. Then, in Subsections 5.2 and 5.3 we prove Lemmata 5.3 and 5.4, respectively.

Proof of Theorem 5.1. Assume for a contradiction that there exist a family of circuits $A = \{A_n\}$, each of size at most $s(n)$ that queries Sam up to depth $d(n)$ such that $s(n)^{3d(n)+2} < 2^{n/8}$, and a depth restriction function depth, for which

$$\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y\leftarrow\{0,1\}^n}}\left[A_n^{\pi,\mathsf{Sam}_{\mathsf{depth}}^{\pi,\mathcal{F},\mathsf{sign}}}(y) = \pi_n^{-1}(y)\right] > \frac{1}{s(n)} \enspace ,$$

for infinitely many values of n. Lemma 5.4 implies that there exists a family $M = \{M_n\}$ of circuits, such that each M_n is of size $O(s(n)) \leq 2^{n/7}$, and

$$\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y\leftarrow\{0,1\}^n}}\left[M_n^{\pi,\mathsf{Sam}_{\mathsf{depth}}^{\pi,\mathcal{F},\mathsf{sign}}}(y) = \pi_n^{-1}(y) \;\; \wedge \;\; \overline{\mathsf{SamHIT}}_y\right] \geq \frac{1}{s(n)^{3d(n)+2}} > \frac{1}{2^{n/8}} \enspace ,$$

for infinitely many values of n, which is a contradiction to Lemma 5.3. ∎

5.1 Extension to Trapdoor Permutations

The basic idea in extending the result for trapdoor permutation is in applying Theorem 5.1 twice. Consider a collection $\tau_n = \left(G_n, F_n, F_n^{-1}\right)$ of trapdoor permutations over $\{0,1\}^n$ and a circuit A which successfully inverts a permutation $F_n(pk, \cdot)$, for some $pk = G_n(td)$, on some image y. If during A's computation the procedure F_n^{-1} is queried with td, then the circuit A can be used to invert a random permutation $\pi_n = G_n$ on pk. In addition, if the procedure F_n^{-1} is not queried with td, then essentially F_n^{-1} does not help in inverting $F_n(pk, \cdot)$ on y, and the circuit A can be used to invert a random permutation $\pi_n = F_n(pk, \cdot)$ on y. Note that an important point in this argument is that F_n^{-1} may be queried by both A and Sam. We prove the following theorem:

Theorem 5.5. *For every circuit A of size $s(n)$ that queries* Sam *up to depth $d(n)$ such that $s(n)^{3d(n)+2} < 2^{n/8}$, for every depth restriction function* depth *and for all sufficiently large n, it holds that*

$$\Pr_{\substack{td\leftarrow\{0,1\}^n,\tau,\mathcal{F} \\ y\leftarrow\{0,1\}^n,\mathsf{sign}}}\left[A^{\tau,\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}}(G_n(td), y) = F_n^{-1}(td, y)\right] \leq \frac{2}{s(n)} \enspace .$$

In order to prove Theorem 5.5 we need a slightly more general form of Theorem 5.1, in which the circuit A has oracle access to an additional (fixed) oracle AUX. Access to this oracle is given also to Sam, in order to enable Sam to sample from circuits with AUX-gates. The following statement is obtained as a straightforward refinement of notations in the proof of Theorem 5.1.

Theorem 5.6. *For every circuit A of size $s(n)$ that queries* Sam *up to depth $d(n)$ such that $s(n)^{3d(n)+2} < 2^{n/8}$, for every depth restriction function* depth*, for every oracle* AUX *and for all sufficiently large n, it holds that*

$$\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y\leftarrow\{0,1\}^n}}\left[A^{\pi,\mathsf{AUX},\mathsf{Sam}_{\mathsf{depth}}^{\pi,\mathsf{AUX},\mathcal{F},\mathsf{sign}}}(y) = \pi_n^{-1}(y)\right] \leq \frac{1}{s(n)} \enspace .$$

Proof of Theorem 5.5. Given $\tau = \left\{\left(G_i, F_i, F_i^{-1}\right)\right\}_{i=1}^{\infty}$, $\mathcal{F}$, sign, depth, $y \in \{0,1\}^n$ and a circuit A, denote by TDHIT_{td} the event in which $A^{\tau,\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}}(G_n(td), y)$ produces a Sam-query $(1^\ell, C_{\mathrm{next}}^\tau, C^\tau, z, dep, sig)$ that results in answer (w', z', sig') such that one of the F_n^{-1} gates in the computations of $C^\tau(w')$ or $C_{\mathrm{next}}^\tau(w')$ has input (td, y') for some y'. Note that without loss of generality, we can assume that A does not query τ directly, as any τ-query can be replaced by a single query to $\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}$. To simplify notation, we denote for the rest of the proof $pk = G_n(td)$, i.e., pk is the public key corresponding to the trapdoor td. For every n, it holds that

$$\begin{aligned}
&\Pr_{\substack{td\leftarrow\{0,1\}^n,\tau,\mathcal{F}\\ y\leftarrow\{0,1\}^n,\mathsf{sign}}}\left[A^{\tau,\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}}(pk,y) = F_n^{-1}(td,y)\right]\\
&\quad\le \Pr_{\substack{td\leftarrow\{0,1\}^n,\tau,\mathcal{F}\\ y\leftarrow\{0,1\}^n,\mathsf{sign}}}\left[A^{\tau,\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}}(pk,y) = F_n^{-1}(td,y) \ \wedge\ \mathsf{TDHIT}_{td}\right] \qquad (5.1)\\
&\qquad + \Pr_{\substack{td\leftarrow\{0,1\}^n,\tau,\mathcal{F}\\ y\leftarrow\{0,1\}^n,\mathsf{sign}}}\left[A^{\tau,\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}}(pk,y) = F_n^{-1}(td,y) \ \middle|\ \overline{\mathsf{TDHIT}}_{td}\right] . \qquad (5.2)
\end{aligned}$$

We show that the expressions in Equations 5.1 and 5.2 can be bounded by the probability of inverting G_n or inverting one of the $F_n(pk,\cdot)$'s respectively, using A as a subroutine.

We begin with Equation 5.1. In this case we can construct a circuit B, such that whenever TDHIT_{td} occurs, B outputs td. Therefore, if

$$\Pr_{\substack{td\leftarrow\{0,1\}^n,\tau,\mathcal{F}\\ y\leftarrow\{0,1\}^n,\mathsf{sign}}}\left[A^{\tau,\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}}(pk,y) = F_n^{-1}(td,y) \ \wedge\ \mathsf{TDHIT}_{td}\right] > \epsilon(n) ,$$

then in particular there is a fixing of $y \in \{0,1\}^n$ and of $F_n(pk,\cdot) \in \Pi_n$ for every $pk \in \{0,1\}^n$ for which this holds (i.e., we fix everything except for $G = \{G_n\}_{n=1}^{\infty}$). Therefore, we have that

$$\Pr_{\substack{G,\mathcal{F},\mathsf{sign}\\ pk\leftarrow\{0,1\}^n}}\left[B^{G,\mathsf{AUX},\mathsf{Sam}_{\mathsf{depth}}^{G,\mathsf{AUX},\mathcal{F},\mathsf{sign}}}(pk) = G_n^{-1}(pk)\right] > \epsilon(n) ,$$

where $\mathsf{AUX} = \left\{\left(F_n, F_n^{-1}\right)\right\}_{n=1}^{\infty}$. Thus, Theorem 5.6 yields that $\epsilon(n) < 1/s(n)$ for all sufficiently large n. Now we consider Equation 5.2 and assume that

$$\Pr_{\substack{td\leftarrow\{0,1\}^n,\tau,\mathcal{F}\\ y\leftarrow\{0,1\}^n,\mathsf{sign}}}\left[A^{\tau,\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}}(pk,y) = F_n^{-1}(td,y) \ \middle|\ \overline{\mathsf{TDHIT}}_{td}\right] > \epsilon(n) .$$

In this case, for every n there exist a specific $pk_n = G_n(td)$ and a fixing of G_n and $F_n(pk'_n,\cdot)$ for all $pk'_n \neq pk_n$ such that

$$\Pr_{\substack{F_{pk},\mathcal{F},\mathsf{sign}\\ y\leftarrow\{0,1\}^n}}\left[A^{F_{pk},\mathsf{AUX},\mathsf{Sam}_{\mathsf{depth}}^{F_{pk},\mathsf{AUX},\mathcal{F},\mathsf{sign}}}(pk_n,y) = F_n^{-1}(td_n,y)\right] > \epsilon(n) ,$$

where $F_{pk} = \{F_i(pk_i,\cdot)\}_{i=1}^{\infty}$, $\mathsf{AUX} = \left(G, F_{\neq pk_n}, F_{\neq pk_n}^{-1}\right)$, and $F_{\neq pk_n}$ and $F_{\neq pk_n}^{-1}$ denote that these oracles do not answer queries on pk_n or $td_n = G_n^{-1}(pk_n)$, respectively. An important remark here is that if the event TDHIT_{td} does not occur, then from the description of the oracle Sam we know that Sam does not query F_n^{-1} on td_n. Therefore it is sufficient to provide Sam with access to $F_{\neq pk_n}^{-1}$. Thus, Theorem 5.6 yields that $\epsilon(n) < 1/s(n)$ for all sufficiently large n. Hence, for all sufficiently large n,

$$\Pr_{\substack{td\leftarrow\{0,1\}^n,\tau,\mathcal{F}\\ y\leftarrow\{0,1\}^n,\mathsf{sign}}}\left[A^{\tau,\mathsf{Sam}_{\mathsf{depth}}^{\tau,\mathcal{F},\mathsf{sign}}}(G_n(td),y) = F_n^{-1}(td,y)\right] \le \frac{2}{s(n)} .$$

■

5.2 The Reconstruction Lemma

The following extends the reconstruction lemma of Gennaro and Trevisan [16]. The idea underlying the claim is the following: if a circuit A manages to invert a permutation π on some set, then given the circuit A, the permutation π can be described without specifying its value on a relatively large fraction of this set. We denote by π_{-n} a family of permutations $\pi = \{\pi_i\}_{i=1}^{\infty}$ where the permutation π_n is left undefined.

Claim 5.7. *For every π, $\mathcal{F}$, sign, depth, circuit A of size s and integer n, if*

$$\Pr_{y \leftarrow \{0,1\}^n}\left[A^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y) = \pi_n^{-1}(y) \;\; \wedge \;\; \overline{\mathsf{SamHIT}_y}\right] \geq \epsilon \;\; ,$$

then, given π_{-n}, $\mathcal{F}$, sign, depth and A, the permutation π_n can be described using $2\log\binom{2^n}{a} + \log((2^n - a)!)$ bits, where $a \geq \epsilon 2^n / \left(2s^2\right)$.

Proof. Denote by $I \subseteq \{0,1\}^n$ the set of points $y \in \{0,1\}^n$ on which A successfully inverts π_n with no y-hits. We claim that there exists a relatively large set $Y \subseteq I$, such that the value of π_n^{-1} on the set Y is determined by π_{-n}, $\mathcal{F}$, sign, depth, A, the sets Y and $X = \pi_n^{-1}(Y)$, and the value of π_n^{-1} on the set $\{0,1\}^n \setminus Y$.

We define the set Y via the following sequential process. Initially Y is empty, and we remove the lexicographically smallest element y from I and insert it into Y. Then, we follow the computation $A^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y)$, denote by $(1^{n_1}, C^\pi_{\mathrm{next},1}, C^\pi_1, z_1, dep_1, sig_1), \ldots, (1^{n_q}, C^\pi_{\mathrm{next},q}, C^\pi_q, z_q, dep_q, sig_q)$ the queries made by A to Sam, and by $(w'_1, z'_1, sig'_1), \ldots, (w'_q, z'_q, sig'_q)$ their corresponding answers. In addition, denote by $y_1, \ldots, y_t$ the outputs of all the π_n-gates in the computations of $C^\pi_1(w'_1), \ldots, C^\pi_q(w'_q)$ and the outputs of all A's direct queries to π_n. We now remove $y_1, \ldots, y_t$ from the set I (note that these are not necessarily in the set I). Then, remove the lexicographically smallest element from the remaining elements of I, insert it to Y and continue in the same manner until the set I is emptied.

Note that at each iteration one element is inserted into the set Y, and at most $s^2 + s + 1 \leq 2s^2$ elements are removed from the set I (the number q of Sam-queries made by A is at most s, and in each circuit given by A as input to Sam the number of π_n-gates is again at most s. In addition, A may directly query π_n on at most s inputs). Since the set I initially contains at least $\epsilon 2^n$ elements, then when the process terminates we have that $|Y| \geq \epsilon 2^n/(2s^2)$.

We now claim that π_n is completely determined given π_{-n}, $\mathcal{F}$, sign, depth, A, the descriptions of the sets Y and $X = \pi_n^{-1}(Y)$, and the value of π_n^{-1} on the set $\{0,1\}^n \setminus Y$. More specifically, we show that the values of π_n^{-1} on the set Y can be reconstructed. For each $y \in Y$ taken in lexicographical increasing order, we reconstruct $\pi_n^{-1}(y)$ by simulating π and $\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}$ in the computation $A^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y)$. Note that if the simulation is correct, then A will output $\pi_n^{-1}(y)$. On input $Q_i = (1^{n_i}, C^\pi_{\mathrm{next},i}, C^\pi_i, z_i, dep_i, sig_i)$ with a corresponding permutation $f_{Q_i} \in \mathcal{F}$, the simulator acts as follows.

1. If $C^\pi_i = \perp$ then output (w'_i, z'_i, sig'_i), where $w'_i = f_{Q_i}(0^m)$, $z'_i = C^\pi_{\mathrm{next},i}(w'_i)$ and $sig'_i = \mathsf{sign}(1^{n_i}, C^\pi_{\mathrm{next},i}, z'_i, 1)$. The simulation is clearly correct in this case.

2. Else, if $C^\pi_{\mathrm{next},i}$ is a refinement of C^π_i, $dep_i \leq \mathsf{depth}(n_i)$ and $sig_i = \mathsf{sign}(1^{n_i}, C^\pi_i, z_i, dep_i)$, then enumerate all $t \in \{0,1\}^m$ in lexicographically increasing order, and output $w'_i = f_{Q_i}(t)$ for the minimal t such that $C^\pi_i(f_{Q_i}(t))$ can be computed (i.e., all π_n-queries can be answered) and its resulting value is z_i. In addition, output $z'_i = C^\pi_{\mathrm{next},i}(w'_i)$ and $sig'_i = \mathsf{sign}(1^{n_i}, C^\pi_{\mathrm{next},i}, z'_i, dep_i + 1)$. We claim that the simulator indeed outputs $w'_i = f_{Q_i}(t)$ for the lexicographically smallest t such that $C^\pi_i(f(t)) = z_i$, and therefore the simulation is correct. Denote by t_0 this minimal t. It is sufficient to show that the simulator can compute $C^\pi_i(f_{Q_i}(t_0))$. Clearly, it can compute any π_i-queries for every $i \neq n$. In addition, the computation may involve four possible π_n-queries:

   - π_n-query on $x \in \{0,1\}^n \setminus X$. The value is explicitly given.
   - π_n-query on $x \in X$ for which $\pi_n(x) <_{lex} y$. The required value was already reconstructed.

- π_n-query on $x \in X$ for which $\pi_n(x) >_{lex} y$. This is impossible: otherwise, we have that both $y \in Y$ and $\pi_n(x) \in Y$, but y was inserted to Y before $\pi_n(x)$ was inserted to Y, and therefore $\pi_n(x)$ should have been removed from I, and in particular not inserted into Y.
- π_n-query on $x \in X$ for which $\pi_n(x) = y$. Impossible, otherwise the Sam-query Q_i produces a y-hit.

3. Else, output $\perp$.

We also have to show that the simulator can answer all of A's direct π-queries. Again, it can clearly answer any direct π_i-queries for every $i \neq n$. Whenever A asks for the value of π_n on some value x, the simulator acts as follows: if this value is already known, then the simulator outputs $\pi_n(x)$ to A. Otherwise, if the value is not known, we claim that it must be that $x = \pi_n^{-1}(y)$ and in this case the simulator successfully reconstructed the desired value and can halt. Indeed, there are four possible such queries:

- π_n-query on $x \in \{0,1\}^n \setminus X$. The value is explicitly given.
- π_n-query on $x \in X$ for which $\pi_n(x) <_{lex} y$. The required value was already reconstructed.
- π_n-query on $x \in X$ for which $\pi_n(x) >_{lex} y$. This is impossible (as above).
- π_n-query on $x \in X$ for which $\pi_n(x) = y$. In this case the query itself gives the desired answer $\pi_n^{-1}(y)$.

Thus, we can successfully reconstruct the values of π_n^{-1} on the set Y. Finally, note that describing the sets Y and X, and the values of π_n^{-1} on the set $\{0,1\}^n \setminus Y$ requires $2\log\binom{2^n}{|Y|} + \log((2^n - |Y|)!)$ bits. ∎

Now we are able to prove the following lemma, which is a stronger form of Lemma 5.3.

Lemma 5.8. *For every π_{-n}, $\mathcal{F}$, sign, depth, circuit A of size at most $2^{n/7}$ and for all sufficiently large n,*

$$\Pr_{\substack{\pi_n \leftarrow \Pi_n \\ y \leftarrow \{0,1\}^n}}\left[A^{\pi, \mathsf{Sam}_{\mathsf{depth}}^{\pi,\mathcal{F},\mathsf{sign}}}(y) = \pi_n^{-1}(y) \;\wedge\; \overline{\mathsf{SamHIT}_y}\right] \leq 2^{-n/8} \enspace .$$

Proof. Claim 5.7 implies that for every circuit A of size $s \leq 2^{n/7}$ and for every π_{-n}, $\mathcal{F}$, sign and depth, the fraction of permutations π_n for which

$$\Pr_{y \leftarrow \{0,1\}^n}\left[A^{\pi, \mathsf{Sam}_{\mathsf{depth}}^{\pi,\mathcal{F},\mathsf{sign}}}(y) = \pi_n^{-1}(y) \;\wedge\; \overline{\mathsf{SamHIT}_y}\right] \geq 2^{-n/7}$$

is at most

$$\frac{\binom{N}{a}^2 (N-a)!}{N!} = \frac{\binom{N}{a}}{a!} \enspace ,$$

where $N = 2^n$, and $a \geq 2^{-n/7} \cdot N/(2s^2) \geq N^{4/7}/2$. Using the inequalities $a! \geq (a/e)^a$ and $\binom{N}{a} \leq (Ne/a)^a$, the above expression is upper bounded by

$$\left(\frac{Ne^2}{a^2}\right)^a \leq \left(\frac{4e^2}{N^{1/7}}\right)^a \leq 2^{-a} \leq 2^{-N^{4/7}/2} \enspace ,$$

for sufficiently large N. Therefore,

$$\Pr_{\substack{\pi_n \leftarrow \Pi_n \\ y \leftarrow \{0,1\}^n}}\left[A^{\pi, \mathsf{Sam}_{\mathsf{depth}}^{\pi,\mathcal{F},\mathsf{sign}}}(y) = \pi_n^{-1}(y) \;\wedge\; \overline{\mathsf{SamHIT}_y}\right] \leq 2^{-N^{4/7}/2} + 2^{-n/7} \leq 2^{-n/8} \enspace .$$

∎

5.3 Avoiding y-Hits by Sam

Given a circuit A of size $s(n)$ that queries Sam up to depth $d(n)$ such that

$$\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y\leftarrow\{0,1\}^n}}\left[A^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y)=\pi_n^{-1}(y)\right]\geq\frac{1}{s(n)} \ ,$$

we would like to construct a circuit M which inverts a random permutation $\pi_n \in \Pi_n$ on a random image $y \in \{0,1\}^n$ almost as well as A does, without M's Sam-queries producing any y-hits. Recall (Definition 5.2), that we say that a Sam-query $Q = (1^\ell, C^\pi_{\mathrm{next}}, C^\pi, z, dep, sig)$ produces a y-hit if Sam outputs (w', z', sig') such that some π_n-gate in the computation of $C^\pi(w')$ has input $\pi_n^{-1}(y)$. In addition, we denoted by SamHIT_y the event in which at least one Sam-query produces a y-hit.

Description of M. On input $y \in \{0,1\}^n$, M feeds A with y as its input, and delivers all of A's queries to Sam and to π with the following exception: for each Sam-query $(1^\ell, C^\pi_{\mathrm{next}}, C^\pi, z, dep, sig)$ with answer (w', z', sig') from Sam, M computes $C^\pi_{\mathrm{next}}(w')$. If some π_n-gate in the computation of $C^\pi_{\mathrm{next}}(w')$ has input $\pi_n^{-1}(y)$, then M outputs $\pi_n^{-1}(y)$ and halts. Otherwise, it provides A with the answer (w', z', sig') to the query, which enables A to proceed with its computation. If M did not halt before the termination of A's computation, then it outputs the output of A and halts.

Proof of Lemma 5.4. The circuit M does not make any additional Sam-queries other than those made by A. Therefore, if A inverts π_n on y without producing any y-hits in its Sam-queries, then so does M. Formally, if

$$\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y\leftarrow\{0,1\}^n}}\left[A^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y)=\pi_n^{-1}(y) \ \wedge \ \overline{\mathsf{SamHIT}_y}\right]\geq\frac{1}{2s(n)} \ ,$$

then

$$\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y\leftarrow\{0,1\}^n}}\left[M^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y)=\pi_n^{-1}(y) \ \wedge \ \overline{\mathsf{SamHIT}_y}\right]\geq\frac{1}{2s(n)} \ .$$

Thus, for the rest of the proof we focus on the more interesting case, in which A does produce a y-hit in one of its Sam-queries with noticeable probability. That is, we assume that

$$\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y\leftarrow\{0,1\}^n}}\left[A^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y)=\pi_n^{-1}(y) \ \wedge \ \mathsf{SamHIT}_y\right]\geq\frac{1}{2s(n)} \ . \tag{5.3}$$

Let us assume for now (only for this paragraph) that all the Sam-queries made by A are legal (as discussed in Section 3), and suppose that $Q_i = (1^{\ell_i}, C^\pi_{\mathrm{next},i}, C^\pi_i, z_i, dep_i, sig_i)$ is a query that produces a y-hit. Recall that the restrictions on Sam impose a forest-like structure on the legal queries, and therefore there exists a parent-query $Q_{\mathrm{p}(i)} = (1^{\ell_{\mathrm{p}(i)}}, C^\pi_{\mathrm{next},\mathrm{p}(i)}, C^\pi_{\mathrm{p}(i)}, z_{\mathrm{p}(i)}, dep_{\mathrm{p}(i)}, sig_{\mathrm{p}(i)})$ of Q_i, for which it holds that $C^\pi_{\mathrm{next},\mathrm{p}(i)} = C^\pi_i$. Our main observation is the following: if Q_i results in output w'_i such that one of the π_n-gates in the computation of $C^\pi_i(w'_i)$ has input $\pi_n^{-1}(y)$, then with high probability the parent query $Q_{\mathrm{p}(i)}$ results in output $w'_{\mathrm{p}(i)}$ such that one of the π_n-gates in the computation of $C^\pi_{\mathrm{next},\mathrm{p}(i)}(w'_{\mathrm{p}(i)})$ has input $\pi_n^{-1}(y)$. Therefore, already after query $Q_{\mathrm{p}(i)}$, the circuit M will retrieve the value $\pi_n^{-1}(y)$ and halt. In particular, M will not query Sam with Q_i, and therefore no y-hits will occur.

It may be that some of the Sam-queries made by A are illegal, and then the above observation does not necessarily hold. Indeed, it is not particulary hard to guess a valid signature on a short query. However, when A tries to invert a permutation π_n over n bits, we need only consider queries with security parameter at least 1^n: any other query does not contain circuits with π_n gates (and therefore cannot produce any y-hits) and also cannot be a parent of queries with π_n gates. Formally, we denote by Legal_n the event in which all the Sam-queries that A makes which include security parameter at least 1^n are legal. Lemma 3.2 enables us to claim that if Equation 5.3 holds, then

$$\begin{aligned}
&\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y\leftarrow\{0,1\}^n}}\left[A^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y)=\pi_n^{-1}(y) \;\wedge\; \mathsf{SamHIT}_y \;\wedge\; \mathsf{Legal}_n\right] \\
&\geq \frac{1}{2s(n)}-\sum_{i\geq n}^{\infty}\frac{s(n)}{2^i} \\
&= \frac{1}{2s(n)}-\frac{s(n)}{2^{n-1}} \\
&\geq \frac{1}{4s(n)} \enspace ,
\end{aligned}$$

where the last inequality holds for every $s(n)\leq 2^{(n-3)/2}$. Therefore, we can assume for the rest of the proof that all the Sam-queries with security parameter at least 1^n are legal. Denote by $Q_1,\ldots,Q_q$ the random variables corresponding to A's queries that have security parameter at least 1^n. Then, assuming that all these queries are legal, we have that every query Q_i is either a root-query (i.e., Q_i is of the form $(1^{\ell_i},C_{\mathrm{next},i},\perp,\perp,\perp,\perp)$), or there exists a query Q_j such that $j=\mathrm{p}(i)$ as discussed in Section 3 (i.e., Q_j is the parent of Q_i).

In order to provide a clear exposition of the proof and its main ideas, we choose to focus here on a simplified case which captures the main difficulties: we assume that A queries Sam *along a single path* up to depth $d=d(n)$. That is, we assume that A's Sam-queries $Q_1,\ldots,Q_d$ satisfy $\mathrm{p}(Q_i)=Q_{i-1}$ for every $2\leq i\leq d$. In Subsection 5.3.1 we describe in detail the extension to the more general case. Under this simplifying assumption, we prove the following lemma:

Lemma 5.9. *For every π, sign and $y\in\{0,1\}^n$, if*

$$\Pr_{\mathcal{F}}\left[A^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y)=\pi_n^{-1}(y) \;\wedge\; \mathsf{SamHIT}_y \;\wedge\; \mathsf{Legal}_n\right]\geq\frac{1}{8s(n)} \enspace , \tag{5.4}$$

then

$$\Pr_{\mathcal{F}}\left[M^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y)=\pi_n^{-1}(y) \;\wedge\; \overline{\mathsf{SamHIT}_y}\right]\geq\frac{1}{s(n)^{3d(n)}} \enspace .$$

Proof of Lemma 5.9. Fix π, sign, $y\in\{0,1\}^n$, and let $s=s(n)$. We introduce the following conventions and notations:

- Without loss of generality, the circuit A does not query π directly, as any π-query can be replaced by a single Sam-query. In order to query π_m on some $t\in\{0,1\}^m$, A computes the description of a circuit C which contains a single π_m-gate with fixed input t. Then, A queries Sam on $(1^m,C,\perp,\perp,\perp,\perp)$ that returns (w',z',sig') where $z'=C(w')=\pi_m(t)$.

- For every $1\leq i\leq d$, we let $Q_i=(C_{\mathrm{next},i},C_i,z_i)$ (for simplicity, since we assume that these queries are legal, we can ignore parts of the input and output of Sam). In addition, denote by $(w'_1,z'_1),\ldots,(w'_d,z'_d)$ the random variables corresponding to Sam's answers on the queries $Q_1,\ldots,Q_d$, respectively. Our simplifying assumption implies that for every $1\leq i\leq d-1$, it holds that $C_{i+1}=C_{\mathrm{next},i}$ and $z_{i+1}=z'_i$.

- For every query Q_i, denote by D_i the distribution from which Sam samples w'_i. Specifically, D_1 is the uniform distribution over $\{0,1\}^m$, and for every $2\leq i\leq d$ the distribution D_i is the uniform distribution over the set $C_i^{-1}(z_i)$ (i.e., over the set of all preimages of z_i under the mapping defined by the circuit C_i). Note that each D_i is in fact random variable, that depends on the previous queries $Q_1, \ldots, Q_{i-1}$ and on the answer w'_{i-1} to the query Q_{i-1}.

- Given a circuit C and an input w, we say that w *produces a* (C, y)*-hit* if some π_n-gate in the computation of $C(w)$ has input $\pi_n^{-1}(y)$.

Throughout the proof, we provide intuitions by considering the interaction between A and Sam as a "game". This game has d rounds, where in the i-th round A chooses a query $Q_i = (C_{\text{next},i}, C_i, z_i)$, and the oracle Sam samples w'_i from the distribution D_i. The goal of the circuit A in this game is to come up with a query Q_i that will produce a y-hit (i.e., to cause the event SamHIT_y). Formally,

- For every $2 \leq i \leq d$, we denote by α_i the probability that query Q_i produces a y-hit, i.e.,

$$\alpha_i = \Pr_{w'_i \leftarrow D_i}\left[w'_i \text{ produces a } (C_i, y)\text{-hit}\right] \quad ,$$

  and we also let $\alpha_1 = 0$ (since $C_1 = \perp$). Note that these α_i's are random variables that depend on the queries $Q_1, \ldots, Q_{i-1}$ and on the answer w'_{i-1} to the query Q_{i-1}.

- For every $2 \leq i \leq d$, we denote by JUMP_i the event that $\alpha_i > \max\left\{64s^2\alpha_{i-1}, 1/(64s^2)^{d+1}\right\}$, and let $\mathsf{JUMP} = \bigcup_i \mathsf{JUMP}_i$.

Equation 5.4 states that A has a noticeable probability in producing a y-hit, and therefore in winning the game. Our first observation is that in this case, the event JUMP occurs with noticeable probability. If JUMP does not occur, then the α_i's are too small in order to produce a y-hit with noticeable probability.

Claim 5.10. $\Pr_{\mathcal{F}}\left[\mathsf{SamHIT}_y \;\middle|\; \overline{\mathsf{JUMP}}\right] \leq 1/(16s)$.

Proof. Assuming that the event JUMP does not occur, we prove by induction that for every $2 \leq i \leq d$ it holds that $\alpha_i \leq 1/(64s^2)^{d-i+3}$. The event $\overline{\mathsf{JUMP}}$ implies that for every $2 \leq i \leq d$ it holds that $\alpha_i \leq \max\left\{64s^2\alpha_{i-1}, 1/(64s^2)^{d+1}\right\}$, and therefore

$$\alpha_2 \leq \max\left\{64s^2\alpha_1, \frac{1}{(64s^2)^{d+1}}\right\} = \max\left\{0, \frac{1}{(64s^2)^{d+1}}\right\} = \frac{1}{(64s^2)^{d+1}} \quad .$$

Suppose now that the claim holds for α_{i-1}, then

$$\alpha_i \leq \max\left\{64s^2\alpha_{i-1}, \frac{1}{(64s^2)^{d+1}}\right\} \leq \max\left\{64s^2 \cdot \frac{1}{(64s^2)^{d-(i-1)+3}}, \frac{1}{(64s^2)^{d+1}}\right\} \leq \frac{1}{(64s^2)^{d-i+3}} \quad .$$

In particular, each α_i is at most $1/(64s^2)^3$, thus

$$\Pr_{\mathcal{F}}\left[\mathsf{SamHIT}_y \;\middle|\; \overline{\mathsf{JUMP}}\right] \leq d \cdot \frac{1}{(64s^2)^3} \leq s \cdot \frac{1}{(64s^2)^3} \leq \frac{1}{16s} \quad .$$

■

As a result of the previous claim, we can now easily derive that the event JUMP has noticeable probability.

Claim 5.11. $\Pr_{\mathcal{F}}\left[\mathsf{JUMP}\right] \geq 1/(16s)$.

Proof. On one hand, Equation 5.4 implies in particular that

$$\Pr_{\mathcal{F}}[\mathsf{SamHIT}_y] \geq \frac{1}{8s} \ .$$

However, on the other hand, Claim 5.10 implies that

$$\begin{aligned}\Pr_{\mathcal{F}}[\mathsf{SamHIT}_y] &\leq \Pr_{\mathcal{F}}[\mathsf{JUMP}] + \Pr_{\mathcal{F}}\left[\mathsf{SamHIT}_y \ \middle|\ \overline{\mathsf{JUMP}}\right] \\ &\leq \Pr_{\mathcal{F}}[\mathsf{JUMP}] + \frac{1}{16s} \ .\end{aligned}$$

Therefore,

$$\Pr_{\mathcal{F}}[\mathsf{JUMP}] \geq \frac{1}{8s} - \frac{1}{16s} \geq \frac{1}{16s} \ .$$

■

At this point, we begin considering the point of view of M in the game. For each query $Q_i = (C_{\mathrm{next},i}, C_i, z_i)$ with answer (w'_i, z'_i), the circuit M computes $C_{\mathrm{next},i}(w'_i)$. If some π_n-gate in this computation has input $\pi_n^{-1}(y)$, then M outputs $\pi_n^{-1}(y)$ and halts. We say that M wins the game, if it manages to retrieve $\pi_n^{-1}(y)$ before A produces any y-hits. Formally,

- For every $1 \leq i \leq d$, we denote by β_i the probability that M outputs $\pi_n^{-1}(y)$ and halts after query Q_i, i.e.,
$$\beta_i = \Pr_{w'_i \leftarrow D_i}\left[w'_i \text{ produces a } (C_{\mathrm{next},i}, y)\text{-hit}\right] \ .$$
  Note that these β_i's are random variables as well, that depend on $Q_1, \ldots, Q_i$.

The game can be now described as follows: in the i-th round, A chooses a query Q_i which determines β_i, and Sam samples w'_i which determines α_{i+1}. If Q_i chose a high β_i, then M has high probability in winning the game: given w'_i from Sam, it will compute $C_{\mathrm{next},i}(w'_i)$ and halt if it finds $\pi_n^{-1}(y)$. In this case, A looses the game. Therefore, A should not choose a high β_i. However, we claim that if β_i is low, then with high probability α_{i+1} will be low as well. However, if α_{i+1} is low, then A has a low probability of producing a y-hit in the next query Q_{i+1}. This means that in order for A to win the game, at some point it must "take a risk" and determine a high β_i.

Formally, the following claim shows that given the queries $Q_1, \ldots, Q_i$, the expectation of α_{i+1} over the choice of $w'_i \leftarrow D_i$ is β_i. Therefore, if β_i is low, then α_{i+1} will be low as well with high probability. Note that by the definitions of β_i and α_{i+1}, given $Q_1, \ldots, Q_i$ the probability β_i is already determined, while α_{i+1} is still a random variable that depends on the answer w'_i to Q_i.

Claim 5.12. *For every $1 \leq i \leq d-1$, given $\mathsf{hist}_i = (Q_1, \ldots, Q_i)$, it holds that $\mathrm{E}_{w'_i \leftarrow D_i}[\alpha_{i+1}] = \beta_i$.*

Proof. Given hist_i, it holds that

$$\mathrm{E}_{w'_i \leftarrow D_i}[\alpha_{i+1}] = \sum_{z'_i} \Pr_{w'_i \leftarrow D_i}\left[z'_i\right] \cdot \Pr_{w'_{i+1} \leftarrow C_{i+1}^{-1}(z'_i)}\left[w'_{i+1} \text{ produces a } (C_{i+1}, y)\text{-hit}\right] \ .$$

Note that although Q_{i+1} is not yet defined, the circuit C_{i+1} is defined by the restriction $C_{i+1} = C_{\mathrm{next},i}$. Now, since $z'_i = C_{\mathrm{next},i}(w'_i) = C_{i+1}(w'_i)$, and $C_{\mathrm{next},i}$ is a refinement of C_i, then we have that

$$\Pr_{w'_i \leftarrow D_i}\left[z'_i\right] = \frac{\left|C_{i+1}^{-1}(z'_i)\right|}{\left|C_i^{-1}(z_i)\right|} \ ,$$

and that

$$\Pr_{w'_{i+1} \leftarrow C_{i+1}^{-1}(z'_i)} \left[w'_{i+1} \text{ produces a } (C_{i+1}, y)\text{-hit} \right] = \frac{\left| \left\{ w \in C_{i+1}^{-1}(z'_i) : w \text{ produces a } (C_{i+1}, y)\text{-hit} \right\} \right|}{\left| C_{i+1}^{-1}(z'_i) \right|} \ .$$

Therefore,

$$\begin{aligned} \mathrm{E}_{w'_i \leftarrow D_i} [\alpha_{i+1}] &= \sum_{z'_i} \frac{\left| C_{i+1}^{-1}(z'_i) \right|}{\left| C_i^{-1}(z_i) \right|} \cdot \frac{\left| \left\{ w \in C_{i+1}^{-1}(z'_i) : w \text{ produces a } (C_{i+1}, y)\text{-hit} \right\} \right|}{\left| C_{i+1}^{-1}(z'_i) \right|} \\ &= \sum_{z'_i} \frac{\left| \left\{ w \in C_{i+1}^{-1}(z'_i) : w \text{ produces a } (C_{i+1}, y)\text{-hit} \right\} \right|}{\left| C_i^{-1}(z_i) \right|} \ . \end{aligned}$$

Finally, the restrictions that $C_{i+1} = C_{\mathrm{next},i}$ and that $C_{\mathrm{next},i}$ is a refinement of C_i imply that

$$C_i^{-1}(z_i) = \biguplus_{z'_i} C_{\mathrm{next},i}^{-1}(z'_i) = \biguplus_{z'_i} C_{i+1}^{-1}(z'_i) \ ,$$

where $\biguplus$ denotes the union of disjoint sets. Thus,

$$\begin{aligned} \mathrm{E}_{w'_i \leftarrow D_i} [\alpha_{i+1}] &= \frac{\left| \left\{ w \in \biguplus_{z'_i} C_{i+1}^{-1}(z'_i) : w \text{ produces a } (C_{i+1}, y)\text{-hit} \right\} \right|}{\left| C_i^{-1}(z_i) \right|} \\ &= \frac{\left| \left\{ w \in C_i^{-1}(z_i) : w \text{ produces a } (C_{\mathrm{next},i}, y)\text{-hit} \right\} \right|}{\left| C_i^{-1}(z_i) \right|} \\ &= \Pr_{w \leftarrow C_i^{-1}(z_i)} [w \text{ produces a } (C_{\mathrm{next},i}, y)\text{-hit}] \\ &= \Pr_{w'_i \leftarrow D_i} \left[w'_i \text{ produces a } (C_{\mathrm{next},i}, y)\text{-hit} \right] \\ &= \beta_i \ . \end{aligned}$$

■

Up to this point, we have reached the conclusion that in order for A to win the game, it must be that at least one of the α_{i+1}'s is high (i.e., the event JUMP_{i+1} occurs). We have also seen that the latter requires A to choose a query Q_i that determines a high β_i. We would like to claim that in this case, it holds that β_i is significantly larger than α_i. Formally,

- For every $1 \leq i \leq d$, denote by GAP_i the event that $\beta_i > \max\left\{2\alpha_i, 1/(64s^2)^{d+2}\right\}$.

The following claim captures the idea that if β_i is not significantly larger than α_i, then α_{i+1} is not significantly larger than α_i as well.

Claim 5.13. *For every $1 \leq i \leq d-1$, given $\mathsf{hist}_i = (Q_1, \ldots, Q_i)$, it holds that*

$$\Pr_{w'_i \leftarrow D_i} \left[\mathsf{JUMP}_{i+1} \;\middle|\; \overline{\mathsf{GAP}}_i \right] \leq 1/(32s^2) \ .$$

Proof. Given $\mathsf{hist}_i = (Q_1, \ldots, Q_i)$, it holds that

$$\begin{aligned} &\Pr_{w'_i \leftarrow D_i} \left[\mathsf{JUMP}_{i+1} \;\middle|\; \overline{\mathsf{GAP}}_i \right] \\ &\quad \leq \Pr_{w'_i \leftarrow D_i} \left[\alpha_{i+1} > 32s^2\beta_i \right] + \Pr_{w'_i \leftarrow D_i} \left[\mathsf{JUMP}_{i+1} \;\middle|\; \overline{\mathsf{GAP}}_i \wedge \left\{ \alpha_{i+1} \leq 32s^2\beta_i \right\} \right] \ . \end{aligned}$$

Claim 5.12 and Markov's inequality imply that

$$\Pr_{w_i' \leftarrow D_i}\left[\alpha_{i+1} > 32s^2\beta_i\right] \leq \frac{1}{32s^2} \ .$$

In addition, note that the events JUMP_{i+1} and GAP_i were defined such that

$$\Pr_{w_i' \leftarrow D_i}\left[\mathsf{JUMP}_{i+1} \ \middle|\ \overline{\mathsf{GAP}}_i \wedge \left\{\alpha_{i+1} \leq 32s^2\beta_i\right\}\right] = 0 \ .$$

Therefore,

$$\Pr_{w_i' \leftarrow D_i}\left[\mathsf{JUMP}_{i+1} \ \middle|\ \overline{\mathsf{GAP}}_i\right] \leq \frac{1}{32s^2} \ .$$

■

The following claim combines the above observations, and shows that with noticeable probability there must be an i such that JUMP_{i+1} occurs and for every such i it holds that GAP_i occurs as well. In other words, if A wins the game with noticeable probability, then there must be some i for which α_{i+1} is high, and for every such i it is the case that A chooses Q_i such that β_i is significantly larger than α_i. These high β_i's will enable M to retrieve $\pi_n^{-1}(y)$ and win the game, before A produces any y-hits.

Claim 5.14. $\Pr_{\mathcal{F}}\left[\mathsf{JUMP} \wedge \left(\bigcap_{i=1}^{d-1}\left\{\mathsf{GAP}_i \vee \overline{\mathsf{JUMP}}_{i+1}\right\}\right)\right] \geq 1/(32s)$.

Proof. Claims 5.11 and 5.13 imply that

$$\begin{aligned}
&\Pr_{\mathcal{F}}\left[\mathsf{JUMP} \wedge \left(\bigcap_{i=1}^{d-1}\left\{\mathsf{GAP}_i \vee \overline{\mathsf{JUMP}}_{i+1}\right\}\right)\right] \\
&\quad \geq \Pr_{\mathcal{F}}\left[\mathsf{JUMP}\right] - \Pr_{\mathcal{F}}\left[\bigcup_{i=1}^{d-1}\left\{\overline{\mathsf{GAP}}_i \wedge \mathsf{JUMP}_{i+1}\right\}\right] \\
&\quad \geq \Pr_{\mathcal{F}}\left[\mathsf{JUMP}\right] - \sum_{i=1}^{d-1}\Pr_{\mathcal{F}}\left[\overline{\mathsf{GAP}}_i \wedge \mathsf{JUMP}_{i+1}\right] \\
&\quad \geq \Pr_{\mathcal{F}}\left[\mathsf{JUMP}\right] - \sum_{i=1}^{d-1}\Pr_{\mathcal{F}}\left[\mathsf{JUMP}_{i+1} \ \middle|\ \overline{\mathsf{GAP}}_i\right] \\
&\quad \geq \frac{1}{16s} - s \cdot \frac{1}{32s^2} \\
&\quad = \frac{1}{32s} \ .
\end{aligned}$$

■

Denote by GOOD the event that JUMP occurs, and for every JUMP_{i+1} that occurs it holds that GAP_i occurs as well (this is the event considered in Claim 5.14). Assume now that GOOD occurs, and denote by i^* the minimal $1 \leq i \leq d-1$ for which JUMP_{i+1} occurs. The probability that the query Q_{i^*} does not produce a y-hit, but M still retrieves $\pi_n^{-1}(y)$ using the answer to this query is at least $\beta_{i^*} - \alpha_{i^*}$. Since GAP_{i^*} occurs, we know that $\beta_{i^*} - \alpha_{i^*}$ is noticeable, and therefore M has a noticeable probability in winning the game at this point. Yet, it might be the case that some previous query produces a y-hit. This event has low probability, since i^* is minimal such JUMP_{i^*+1} occurs, and therefore all the previous α_i's are not sufficiently high in order to produce a y-hit. The following claim concludes the proof of Lemma 5.9.

Claim 5.15. $\Pr_{\mathcal{F}}\left[M^{\pi,\mathsf{Sam}^{\pi,\mathcal{F}}_{\mathsf{depth}}}(y) = \pi_n^{-1}(y) \;\; \wedge \;\; \overline{\mathsf{SamHIT}}_y\right] \geq 1/\left(s^{3d}\right)$.

Proof. Given that the event GOOD occurs, denote by i^* the minimal $1 \leq i \leq d-1$ for which JUMP_{i+1} occurs (as discussed above). For this i^*, the event GAP_{i^*} occurs as well and for every query Q_i that precedes Q_{i^*} the event JUMP_{i+1} does not occur. Therefore, whenever the event GOOD occurs, we can hope that the following (independent) events will take place:

- None of the queries $Q_1, \ldots, Q_{i^*-1}$ will produce a y-hit. Since for every such query Q_i the event JUMP_i does not occur, then, exactly as in the proof of Claim 5.10, the probability of this event is at least $1 - 1/(16s)$.

- Given Q_{i^*}, Sam samples w'_{i^*} which does not produce a (C_{i^*}, y)-hit, but does produce a (C_{next,i^*}, y)-hit. In other words, the query Q_{i^*} does not produce a y-hit, but still M retrieves the value $\pi_n^{-1}(y)$. The probability of this event is at least

$$\beta_{i^*} - \alpha_{i^*} \geq \frac{\beta_{i^*}}{2} \geq \frac{1}{2\left(64s^2\right)^{d+2}} \;\; .$$

Putting these together, we obtain

$$\begin{aligned}\Pr_{\mathcal{F}}\left[M^{\pi,\mathsf{Sam}^{\pi,\mathcal{F}}_{\mathsf{depth}}}(y) = \pi^{-1}(y) \;\; \wedge \;\; \overline{\mathsf{SamHIT}}_y\right] &\geq \Pr_{\mathcal{F}}\left[\mathsf{GOOD}\right] \cdot \left(1 - \frac{1}{16s}\right) \cdot \frac{1}{2\left(64s^2\right)^{d+2}} \\ &\geq \frac{1}{32s} \cdot \left(1 - \frac{1}{16s}\right) \cdot \frac{1}{2\left(64s^2\right)^{d+2}} \\ &\geq \frac{1}{s^{3d}} \;\; .\end{aligned}$$

■

This concludes the proof of Lemma 5.9. We now turn to complete the proof of Lemma 5.4. Recall that we were left to deal with the case that

$$\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y \leftarrow \{0,1\}^n}}\left[A^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y) = \pi_n^{-1}(y) \;\; \wedge \;\; \mathsf{SamHIT}_y \;\; \wedge \;\; \mathsf{Legal}_n\right] \geq \frac{1}{4s(n)} \;\; .$$

Let

$$T = \left\{(y, \pi, \mathsf{sign}) : \Pr_{\mathcal{F}}\left[A^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y) = \pi_n^{-1}(y) \;\; \wedge \;\; \mathsf{SamHIT}_y \;\; \wedge \;\; \mathsf{Legal}_n\right] \geq \frac{1}{8s(n)}\right\} \;\; .$$

Then

$$\Pr_{y \leftarrow \{0,1\}^n, \pi, \mathsf{sign}}\left[(y, \pi, \mathsf{sign}) \in T\right] \geq 1/8s(n) \;\; ,$$

and Lemma 5.9 implies that for every $(y, \pi, \mathsf{sign}) \in T$ we have

$$\Pr_{\mathcal{F}}\left[M^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y) = \pi_n^{-1}(y) \;\; \wedge \;\; \overline{\mathsf{SamHIT}}_y\right] \geq \frac{1}{s(n)^{3d(n)}} \;\; .$$

Therefore,

$$\begin{aligned}&\Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y \leftarrow \{0,1\}^n}}\left[M^{\pi,\mathsf{Sam}^{\pi,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}}(y) = \pi_n^{-1}(y) \;\; \wedge \;\; \overline{\mathsf{SamHIT}}_y\right] \\ &\geq \Pr_{\substack{\pi,\mathsf{sign} \\ y \leftarrow \{0,1\}^n}}\left[(y, \pi, \mathsf{sign}) \in T\right] \cdot \Pr_{\substack{\pi,\mathcal{F},\mathsf{sign} \\ y \leftarrow \{0,1\}^n}}\left[M^{\pi,\mathsf{Sam}^{\pi,\mathcal{F}}_{\mathsf{depth}}}(y) = \pi_n^{-1}(y) \wedge \overline{\mathsf{SamHIT}}_y \,\middle|\, (y, \pi, \mathsf{sign}) \in T\right] \\ &\geq \frac{1}{8s(n)} \cdot \frac{1}{s(n)^{3d(n)}} \\ &\geq \frac{1}{s(n)^{3d(n)+2}} \;\; .\end{aligned}$$

■

5.3.1 Dealing with a general query structure

We describe the extension of the above proof of Lemma 5.9 to the case where A's queries with security parameter at least 1^n are not necessarily along a single path. In general, these queries may form a forest structure, as discussed above and in Section 3 (i.e., each query Q_j is either a root-query, or there exists a query Q_i such that $i = \mathrm{p}(j)$). This extension is mainly technical, and essentially does not require anything more than refining some events and notations.

Consider the random variables $Q_1, \ldots, Q_q$ corresponding to A's Sam-queries with security parameter at least 1^n. As in the above proof, we can assume that these are all legal queries, and ignore all the queries with security parameters less than 1^n – this part of the proof does not change at all. However, whereas in the simplified version of the proof, the parent query of each Q_i was Q_{i-1} (i.e., $\mathrm{p}(i) = i-1$), when considering an arbitrary forest structure of the queries, the values $\mathrm{p}(i)$ are random variables as well.

We deal with this issue by utilizing the following observation: Assume that Q_i and Q_j are two queries such that $\mathrm{p}(j) = i$. Then, the probability α_j that Q_j produces a y-hit is already determined given the query $Q_i = (C_{\mathrm{next},i}, C_i, z_i)$ and its answer (w'_i, z'_i). Indeed, the forest structure guarantees that any such Q_j is of the form $(C_{\mathrm{next},j}, C_j, z_j)$ where $C_j = C_{\mathrm{next},i}$ and $z_j = z'_i$. Therefore, by the definition of the distribution D_j we have that the probability that Q_j produces a y-hit is

$$\begin{aligned}
\alpha_j &= \Pr_{w'_j \leftarrow D_j}\left[w'_j \text{ produces a } (C_j, y)\text{-hit}\right] \\
&= \Pr_{w'_j \leftarrow C_j^{-1}(z_j)}\left[w'_j \text{ produces a } (C_j, y)\text{-hit}\right] \\
&= \Pr_{w'_j \leftarrow C_{\mathrm{next},i}^{-1}(z'_i)}\left[w'_j \text{ produces a } (C_{\mathrm{next},i}, y)\text{-hit}\right] \quad .
\end{aligned}$$

It is now clear that α_j can be defined even before the query Q_j is determined (assuming of course that $\mathrm{p}(j) = i$). We formally capture this property as follows:

- For every $1 \leq i \leq q$, we denote by γ_i the probability that a potential child of $Q_i = (C_{\mathrm{next},i}, C_i, z_i)$ produces a y-hit, i.e.,

  $$\gamma_i = \Pr_{w'_j \leftarrow C_{\mathrm{next},i}^{-1}(z'_i)}\left[w'_j \text{ produces a } (C_{\mathrm{next},i}, y)\text{-hit}\right] \quad ,$$

  Note that these γ_i's are random variables which are determined given the query Q_i and its answer (w'_i, z'_i). Then, for every two queries Q_i and Q_j such that $\mathrm{p}(j) = i$, we have that $\alpha_j = \gamma_i$.

- For every $1 \leq i \leq q$, we denote by PJUMP_i (for "$\mathsf{Potential\ JUMP}$") the event in which the probability that a potential child of Q_i hits y is significantly larger than the probability that Q_i hits y. Specifically, it is the event that $\gamma_i > \max\left\{64s^2\alpha_i, 1/(64s^2)^{d+1}\right\}$. Then, for every two queries Q_i and Q_j such that $\mathrm{p}(j) = i$, we have that the events JUMP_j and PJUMP_i are equivalent.

We now describe in detail the required technical changes to the proof of Lemma 5.9. The new proof begins as before by proving that $\Pr_{\mathcal{F}}\left[\mathsf{SamHIT}_y \;\middle|\; \overline{\mathsf{JUMP}}\right] \leq 1/(16s)$ and that $\Pr_{\mathcal{F}}\left[\mathsf{JUMP}\right] \geq 1/(16s)$. In these two proofs the new definitions are not yet relevant and the only change is the usage of the notation $\mathrm{p}(i)$ instead of $i-1$. Then, Claims 5.12 and 5.13 are replaced with showing that $\mathrm{E}_{w'_i \leftarrow D_i}\left[\gamma_i\right] = \beta_i$, and that $\Pr_{w'_i \leftarrow D_i}\left[\mathsf{PJUMP}_i \;\middle|\; \overline{\mathsf{GAP}}_i\right] \leq 1/(32s^2)$. Again, these are the exact same proofs, where γ_i replaces α_{i+1}. The final change is in Claim 5.14, which is replaced by showing that $\Pr_{\mathcal{F}}\left[\mathsf{JUMP} \wedge \left(\bigcap_j \left\{\mathsf{GAP}_{\mathrm{p}(j)} \vee \overline{\mathsf{JUMP}}_j\right\}\right)\right] \geq 1/(32s)$, as follows:

$$
\begin{aligned}
&\Pr_{\mathcal{F}}\left[\mathsf{JUMP} \wedge \left(\bigcap_j \left\{\mathsf{GAP}_{\mathrm{p}(j)} \vee \overline{\mathsf{JUMP}}_j\right\}\right)\right] \\
&\quad \geq \Pr_{\mathcal{F}}[\mathsf{JUMP}] - \Pr_{\mathcal{F}}\left[\bigcup_j \left\{\overline{\mathsf{GAP}}_{\mathrm{p}(j)} \wedge \mathsf{JUMP}_j\right\}\right] \\
&\quad \geq \Pr_{\mathcal{F}}[\mathsf{JUMP}] - \Pr_{\mathcal{F}}\left[\bigcup_i \left\{\overline{\mathsf{GAP}}_i \wedge \mathsf{PJUMP}_i\right\}\right] \\
&\quad \geq \Pr_{\mathcal{F}}[\mathsf{JUMP}] - \sum_i \Pr_{\mathcal{F}}\left[\overline{\mathsf{GAP}}_i \wedge \mathsf{PJUMP}_i\right] \\
&\quad \geq \Pr_{\mathcal{F}}[\mathsf{JUMP}] - \sum_i \Pr_{\mathcal{F}}\left[\mathsf{PJUMP}_i \;\middle|\; \overline{\mathsf{GAP}}_i\right] \\
&\quad \geq \frac{1}{16s} - s \cdot \frac{1}{32s^2} \\
&\quad = \frac{1}{32s} \enspace .
\end{aligned}
$$

6 The Round Complexity Lower Bound

In this section we combine the results presented in Sections 4 and 5 and derive our main theorem. As described in Subsection 1.2, given a fully-black-box construction of a statistically-hiding commitment scheme from a family of trapdoor permutations, we consider three parameters:

1. $d(n)$ – the number of communication rounds in the commitment scheme with security parameter 1^n.
2. $s(n)$ – the hardness of the trapdoor permutation family (see Definition 2.3).
3. $\ell(n)$ – the security parameter expansion of the construction (see Definition 2.8).

We first state our result for the more standard hardness notion of trapdoor permutations, in which we consider a family of trapdoor permutations which is $s(n)$-hard for any polynomial $s(n)$. As discussed in Subsection 1.2, we consider both constructions which are security-preserving (i.e., $\ell(n) = O(n)$), and constructions which are not necessarily security-preserving (i.e., $\ell(n)$ is any polynomial in n). We begin by formally stating our results for these two cases.

Theorem 6.1. *Any $O(n)$-security-parameter-expanding fully-black-box construction of a weakly-binding and honest-receiver statistically-hiding commitment scheme from a family of trapdoor permutations has $\Omega\left(\frac{n}{\log n}\right)$ communication rounds.*

Theorem 6.2. *Any fully-black-box construction of a weakly-binding and honest-receiver statistically-hiding commitment scheme from a family of trapdoor permutations has $n^{\Omega(1)}$ communication rounds.*

The above two theorems are in fact obtained as corollaries of a more general statement. In this statement we specifically consider the notion of $s(n)$-hard trapdoor permutations, and do not consider a particular range for the security parameter expansion $\ell(n)$. The main theorem of the paper is formally stated as follows:

Theorem 6.3 (Main Theorem). *For every $\ell(n)$-security-parameter-expanding fully-black-box construction of a $d(n)$-round weakly-binding and honest-receiver statistically-hiding commitment scheme from an $s(n)$-hard family of trapdoor permutations, it holds that $d(\ell(n)) = \Omega\left(\frac{n}{\log s(n)}\right)$.*

Before turning to the formal proof of Theorem 6.3, we first provide a very brief overview. Given an $\ell(n)$-security-parameter-expanding fully-black-box construction $(\mathcal{S}, \mathcal{R}, \mathcal{V}, A)$ of a $d(n)$-round weakly-binding and honest-receiver statistically-hiding commitment scheme from an $s(n)$-hard family of trapdoor permutations, we show that there exists a oracle $\mathcal{O} = \left(\tau, \mathsf{Sam}^{\tau,\mathcal{F},\mathsf{sign}}_{\mathsf{depth}}\right)$ relative to which the following holds: there exists a malicious sender $\mathcal{S}^*$ that breaks the binding of the scheme $(\mathcal{S}^\tau, \mathcal{R}^\tau, \mathcal{V}^\tau)$, but if $d(\ell(n)) < \frac{c \cdot n}{\log s(n)}$ for some particular constant $c > 0$, then the machine A fails to break the security of τ. A technical difficulty is that the results proved in Sections 4 and 5 hold with respect to a *distribution* of oracles and not for a *single* oracle (they hold over the random choices of τ, $\mathcal{F}$ and sign). An application of the Borel-Cantelli lemma will enable us to overcome this difficulty. In order to apply the Borel-Cantelli lemma, we need the following statement, which is an immediate corollary of Theorem 4.1 via a standard averaging argument:

Corollary 6.4. *For every $d(n)$-round honest-receiver statistically-hiding bit-commitment scheme $(\mathcal{S}, \mathcal{R}, \mathcal{V})$ with oracle access to a family of trapdoor permutations, there exist a polynomial-time malicious sender $\mathcal{S}^*$ and a negligible function $\nu(n)$, such that for all sufficiently large n with probability at least $1 - 1/n^2$ over the choices of τ, $\mathcal{F}$ and sign, it holds that*

$$\Pr_{r_{\mathcal{R}}}\left[\begin{array}{r}((\mathsf{decom}, \mathsf{decom}')|\mathsf{com}) \leftarrow \left\langle \mathcal{S}^{*\ \mathsf{Sam}^{\tau,\mathcal{F},\mathsf{sign}}_{d+1}}(1^n), \mathcal{R}^\tau(1^n, r_{\mathcal{R}})\right\rangle : \\ \mathcal{V}^\tau(\mathsf{com}, \mathsf{decom}) = 0, \mathcal{V}^\tau(\mathsf{com}, \mathsf{decom}') = 1\end{array}\right] > 1 - \nu(n) \ .$$

We now turn to formally prove Theorem 6.3.

Proof of Theorem 6.3. Let $(\mathcal{S}, \mathcal{R}, \mathcal{V}, A)$ be an $\ell(n)$-security-parameter-expanding fully-black-box construction of a $d(n)$-round weakly-binding and honest-receiver statistically-hiding commitment scheme from an $s(n)$-hard family of trapdoor permutations. From this point on, we fix the depth restriction function $\mathsf{depth} : \mathbb{N} \to \mathbb{N}$ of the oracle Sam to be the function $d(n) + 1$. For every τ, $\mathcal{F}$ and sign, we denote by $E_n^{\tau,\mathcal{F},\mathsf{sign}}$ the event in which

$$\Pr_{r_{\mathcal{R}}}\left[\begin{array}{r}((\mathsf{decom}, \mathsf{decom}')|\mathsf{com}) \leftarrow \left\langle \mathcal{S}^{*\ \mathsf{Sam}^{\tau,\mathcal{F},\mathsf{sign}}_{d+1}}(1^n), \mathcal{R}^\tau(1^n, r_{\mathcal{R}})\right\rangle : \\ \mathcal{V}^\tau(\mathsf{com}, \mathsf{decom}) = 0, \mathcal{V}^\tau(\mathsf{com}, \mathsf{decom}') = 1\end{array}\right] \leq 1 - \nu(n) \ ,$$

where $\mathcal{S}^*$ and $\nu(n)$ are given by Corollary 6.4. Then, Corollary 6.4 states that for all sufficiently large n, it holds that $\Pr_{\tau,\mathcal{F},\mathsf{sign}}\left[E_n^{\tau,\mathcal{F},\mathsf{sign}}\right] \leq 1/n^2$, and thus

$$\sum_{n=1}^{\infty} \Pr_{\tau,\mathcal{F},\mathsf{sign}}\left[E_n^{\tau,\mathcal{F},\mathsf{sign}}\right] < \infty \ .$$

Therefore, the Borel-Cantelli lemma implies that the probability over the choices of τ, $\mathcal{F}$ and sign that the event $E_n^{\tau,\mathcal{F},\mathsf{sign}}$ occurs for infinitely many n's is zero. That is, for measure 1 of the oracles $\mathsf{Sam}_{d+1}^{\tau,\mathcal{F},\mathsf{sign}}$ it holds that

$$\Pr_{r_{\mathcal{R}}}\left[\begin{array}{r}((\mathsf{decom},\mathsf{decom}')|\mathsf{com}) \leftarrow \left\langle \mathcal{S}^{*\ \mathsf{Sam}_{d+1}^{\tau,\mathcal{F},\mathsf{sign}}}(1^n), \mathcal{R}^{\tau}(1^n, r_{\mathcal{R}})\right\rangle : \\ \mathcal{V}^{\tau}(\mathsf{com},\mathsf{decom}) = 0, \mathcal{V}^{\tau}(\mathsf{com},\mathsf{decom}') = 1\end{array}\right] > 1 - \nu(n) \ ,$$

for all sufficiently large n. In other words, relative to measure 1 of the oracles $\mathsf{Sam}_{d+1}^{\tau,\mathcal{F},\mathsf{sign}}$, the malicious sender $\mathcal{S}^*$ breaks the weak binding of the commitment scheme. Thus, the fully-black-box construction guarantees that relative to these oracles, we have that

$$\Pr\left[A^{\tau,\mathcal{S}^{*\ \mathsf{Sam}_{d+1}^{\tau,\mathcal{F},\mathsf{sign}}}}(1^n, G_n(td), y) = F_n^{-1}(td, y)\right] > \frac{1}{s(n)} \ , \tag{6.1}$$

for infinitely many values of n, where A runs in time $s(n)$, and the probability is taken uniformly over all the possible choices of $td \in \{0,1\}^n$ and $y \in \{0,1\}^n$, and over all the possible outcomes of the internal coin tosses of A. Note that since Equation 6.1 holds with respect to measure 1 of the oracles $\mathsf{Sam}_{d+1}^{\tau,\mathcal{F},\mathsf{sign}}$ (i.e., with probability 1 over the choice of $\tau, \mathcal{F}$ and sign), then Equation 6.1 still holds when the probability is taken over the choices of τ, $\mathcal{F}$ and sign as well. In addition, by converting the Turing-machine A to a circuit family, and by incorporating the description of $\mathcal{S}^*$ into this family, we obtain that there exists a circuit A^* of size at most, say, $s^*(n) = (s(n))^2$ such that

$$\Pr_{\substack{td\leftarrow\{0,1\}^n,\tau,\mathcal{F}\\ y\leftarrow\{0,1\}^n,\mathsf{sign}}}\left[A^{*\ \tau,\mathsf{Sam}_{d+1}^{\tau,\mathcal{F},\mathsf{sign}}}(G_n(td), y) = F_n^{-1}(td, y)\right] > \frac{1}{s(n)} > \frac{2}{s^*(n)} \ ,$$

for infinitely many values of n.

The assumption that the construction is $\ell(n)$-security-parameter-expanding (i.e., that A when given security parameter 1^n invokes $\mathcal{S}^*$ on security parameters which are at most $1^{\ell(n)}$), guarantees that A uses $\mathcal{S}^*$ in a way such that Sam is queried up to depth at most $d(\ell(n)) + 1$. This means that also the circuit A^* queries Sam up to depth at most $d(\ell(n)) + 1$.

We conclude the proof by observing that if $s^*(n)^{3d(\ell(n))+2} < 2^{n/8}$, then the existence of the circuit A^* contradicts Theorem 5.5, and therefore $s^*(n)^{3d(\ell(n))+2} \geq 2^{n/8}$, i.e., $d(\ell(n)) = \Omega\left(\frac{n}{\log s(n)}\right)$.
■

7 Implications to Other Cryptographic Protocols

Our lower bound on the round complexity of statistical commitment schemes implies similar lower bounds for several other cryptographic protocols. Our result can be extended to any cryptographic protocol which can be used to construct a weakly-binding honest-receiver statistically-hiding commitment scheme in a fully-black-box manner. Specifically, in this section we derive new lower bounds on the round complexity of single-server private information retrieval, interactive hashing, and oblivious transfer that guarantees statistical security for one of the parties. We state the corollaries in this section for construction that are security preserving (i.e., $O(n)$-security-parameter-expanding) and note that more general statements, as in Theorem 6.3, could be easily derived as well.

7.1 Single-Server Private Information Retrieval

A single-server private information retrieval (PIR) scheme [9] is a protocol between a server and a user. The server holds a database $x \in \{0,1\}^n$, and the user holds an index $i \in [n]$ to an entry of the database. Informally, the user wishes to retrieve the i-th entry of the database, without revealing to the server the value i. A naive solution is to have the user download the entire database, however, the total communication complexity of this solution is n bits. Based on specific number-theoretic assumptions, several schemes with sublinear communication complexity were developed (see [6, 8, 17, 42, 39], and a recent survey by Ostrovsky and Skeith [50]). The only non-trivial construction based on general computational assumptions is due to Kushilevitz and Ostrovsky [40]. Assuming the existence of trapdoor permutations, they constructed an interactive protocol whose communication complexity is $n - o(n)$ bits.

Beimel, Ishai, Kushilevitz and Malkin [2] showed that any single-server PIR protocol with communication complexity of at most $n/2$ bits, can be used to construct a weakly-binding statistically-hiding commitment scheme. Their construction is both fully-black-box and preserves the number of rounds. Thus, by combining this with our result, we obtain the following corollary:

Corollary 7.1. *Any $O(n)$-security-parameter-expanding fully-black-box construction of a single-server PIR protocol for an n-bit database from a family of trapdoor permutations, in which the server communicates less than $n/2$ bits, has $\Omega\left(\frac{n}{\log n}\right)$ communication rounds.*

Corollary 7.1 yields in particular a lower bound on the communication complexity: any such construction requires the server to communicate $\Omega\left(\frac{n}{\log n}\right)$ bits.

7.2 Interactive Hashing

Interactive hashing was introduced by Naor, Ostrovsky, Venkatesan and Yung [46] and is a protocol that allows a sender $\mathcal{S}$ to commit to a value y while only revealing to the receiver $\mathcal{R}$ the value $(h, z = h(y))$, where h is a 2-to-1 hash function chosen interactively during the protocol.[12] The two security properties of interactive hashing are binding ($\mathcal{S}$ is bounded by the protocol to producing at most one value of y which is consistent with the transcript) and hiding ($\mathcal{R}$ does not obtain any information about y, except for $h(y)$). Naor et al. constructed an interactive hashing protocol from any one-way permutation with $O\left(\frac{n}{\log n}\right)$ communication rounds, and showed that it implies in a fully-black-box manner a statistical commitment scheme with the same number of rounds.[13] Wee [58] has recently showed that a restricted class of fully-black-box constructions of interactive hashing from one-way permutations has $\Omega\left(\frac{n}{\log n}\right)$ rounds. Our result extends Wee's lower bound both to include the most general form of such constructions, and to trapdoor permutations.

Corollary 7.2. *Any $O(n)$-security-parameter-expanding fully-black-box construction of an interactive hashing protocol from a family of trapdoor permutations has $\Omega\left(\frac{n}{\log n}\right)$ communication rounds.*

7.3 Oblivious Transfer

Oblivious transfer (OT), introduced by Rabin [51], is a fundamental primitive in cryptography. In particular, it was shown to imply secure multiparty computation [25, 35, 60]. OT has several equivalent formulations, and we consider the formulation of $\binom{2}{1}$-OT, defined by Even, Goldreich and Lempel [12]. $\binom{2}{1}$-OT is a protocol between two parties, a sender and a receiver. The sender's input consists of two secret bits (b_0, b_1), and the receiver's input consists of a value $i \in \{0, 1\}$. At the end

[12] Several extensions to this definition were suggested, see [30, 49].

[13] Although the original proof in [46] showed the result for $O(n)$ rounds, this was recently reduced to $O\left(\frac{n}{\log n}\right)$ rounds [30, 38].

of the protocol, the receiver should learn the bit b_i while the sender does not learn the value i. The security of the protocol guarantees that even a cheating receiver should not be able to learn the bit b_{1-i}, and a cheating sender should not be able to learn i.

Given any $\binom{2}{1}$-OT protocol that guarantees statistical security for one of the parties (sender or receiver), one can construct a weakly-binding statistically-hiding commitment scheme in a fully-black-box manner while preserving the number of rounds. For the explicit reduction from a statistically protected sender see [13], and the reduction from a statistically protected receiver follows similar lines.[14] Thus, by combining this with our result, we obtain the following corollary:

Corollary 7.3. *Any $O(n)$-security-parameter-expanding fully-black-box construction of a $\binom{2}{1}$-OT protocol that guarantees statistical security for one of the parties from a family of trapdoor permutations has $\Omega\left(\frac{n}{\log n}\right)$ communication rounds.*

We stress that there exist constructions of semi-honest receiver $\binom{2}{1}$-OT protocols, relying on specific number-theoretic assumptions, where the sender enjoys statistical security with a constant number of rounds (e.g., Aiello et al. [1] and Naor and Pinkas [47]). Hence, as for statistical commitments, we demonstrate a large gap between the round complexity of OT constructions based on general assumptions and OT constructions based on specific number-theoretic assumptions.

References

[1] W. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. In *Advances in Cryptology - EUROCRYPT '01*, pages 119–135, 2001.

[2] A. Beimel, Y. Ishai, E. Kushilevitz, and T. Malkin. One-way functions are essential for single-server private information retrieval. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 89–98, 1999.

[3] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, 1984.

[4] J. Boyar, S. A. Kurtz, and M. W. Krentel. A discrete logarithm implementation of perfect zero-knowledge blobs. *Journal of Cryptology*, 2(2):63–76, 1990.

[5] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[6] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In *Advances in Cryptology - EUROCRYPT '99*, pages 402–414, 1999.

[7] R. Canetti, J. Kilian, E. Petrank, and A. Rosen. Black-box concurrent zero-knowledge requires (almost) logarithmically many rounds. *SIAM Journal on Computing*, 32(1):1–47, 2002.

[8] Y. Chang. Single database private information retrieval with logarithmic communication. In *Proceedings of the 9th Australasian Conference on Information Security and Privacy*, pages 50–61, 2004.

[9] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Proceedings of 36th Annual Symposium on Foundations of Computer Science*, pages 41–50, 1995.

[14] Alternatively, refer to [59] for switching the roles of the sender and the receiver.

[10] I. Damgård, T. P. Pedersen, and B. Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. *Journal of Cryptology*, 10(3):163–194, 1997.

[11] C. Dwork, M. Naor, and A. Sahai. Concurrent zero-knowledge. *Journal of the ACM*, 51(6):851–898, 2004.

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

[13] M. Fischlin. On the impossibility of constructing non-interactive statistically-secret protocols from any trapdoor one-way function. In *Topics in Cryptology - The Cryptographers' Track at the RSA Conference*, pages 79–95, 2002.

[14] R. Gennaro, Y. Gertner, and J. Katz. Lower bounds on the efficiency of encryption and digital signature schemes. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 417–425, 2003.

[15] R. Gennaro, Y. Gertner, J. Katz, and L. Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM Journal on Computing*, 35(1):217–246, 2005.

[16] R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 305–313, 2000.

[17] C. Gentry and Z. Ramzan. Single-database private information retrieval with constant communication rate. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming*, pages 803–815, 2005.

[18] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 325–335, 2000.

[19] O. Goldreich. *Foundations of Cryptography – Volume 1: Basic Tools*. Cambridge University Press, 2001.

[20] O. Goldreich. *Foundations of Cryptography – Volume 2: Basic Applications*. Cambridge University Press, 2004.

[21] O. Goldreich, S. Goldwasser, and S. Micali. On the cryptographic applications of random functions. In *Advances in Cryptology - CRYPTO '84*, pages 276–288, 1984.

[22] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

[23] O. Goldreich and A. Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, 1996.

[24] O. Goldreich and H. Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.

[25] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 218–229, 1987.

[26] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[27] S. Hada and T. Tanaka. On the existence of 3-round zero-knowledge protocols. In *Advances in Cryptology - CRYPTO '98*, pages 408–423, 1998.

[28] I. Haitner. Implementing oblivious transfer using collection of dense trapdoor permutations. In *Proceedings of the 1st Theory of Cryptography Conference*, pages 394–409, 2004.

[29] I. Haitner, O. Horvitz, J. Katz, C.-Y. Koo, R. Morselli, and R. Shaltiel. Reducing complexity assumptions for statistically-hiding commitment. In *Advances in Cryptology - EUROCRYPT '05*, pages 58–77, 2005.

[30] I. Haitner and O. Reingold. A new interactive hashing theorem. In *Proceedings of the 22nd Annual IEEE Conference on Computational Complexity*, pages 319–332, 2007.

[31] I. Haitner and O. Reingold. Statistically-hiding commitment from any one-way function. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 1–10, 2007.

[32] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[33] O. Horvitz and J. Katz. Bounds on the efficiency of "black-box" commitment schemes. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming*, pages 128–139, 2005.

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

[35] J. Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 20–31, 1988.

[36] J. Kilian, C. Rackoff, and E. Petrank. Lower bounds for concurrent zero knowledge. *Combinatorica*, 25(2):217–249, 2005.

[37] J. H. Kim, D. R. Simon, and P. Tetali. Limits on the efficiency of one-way permutation-based hash functions. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 535–542, 1999.

[38] T. Koshiba and Y. Seri. Round-efficient one-way permutation based perfectly concealing bit commitment scheme. Electronic Colloquium on Computational Complexity, Report TR06-093, 2006.

[39] E. Kushilevitz and R. Ostrovsky. Replication is NOT needed: SINGLE database, computationally-private information retrieval. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 364–373, 1997.

[40] E. Kushilevitz and R. Ostrovsky. One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In *Advances in Cryptology - EUROCRYPT '00*, pages 104–121, 2000.

[41] Y. Lindell. Parallel coin-tossing and constant-round secure two-party computation. *Journal of Cryptology*, 16(3):143–184, 2003.

[42] H. Lipmaa. An oblivious transfer protocol with log-squared communication. In *Proceedings of the 8th International Conference on Information Security*, pages 314–328, 2005.

[43] M. Luby. *Pseudorandomness and Cryptographic Applications*. Princeton University Press, 1996.

[44] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2):373–386, 1988.

[45] M. Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991.

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

[47] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the 12th Annual Symposium on Discrete Algorithms*, pages 448–457, 2001.

[48] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 33–43, 1989.

[49] M.-H. Nguyen, S. J. Ong, and S. P. Vadhan. Statistical zero-knowledge arguments for NP from any one-way function. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 3–14, 2006.

[50] R. Ostrovsky and W. E. Skeith. A survey of single database PIR: Techniques and applications. Cryptology ePrint Archive, Report 2007/059, 2007.

[51] M. O. Rabin. How to exchange secret by oblivious transfer. Technical Report TR-81, Harvard University, 1981.

[52] O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of reducibility between cryptographic primitives. In *Proceedings of the 1st Theory of Cryptography Conference*, pages 1–20, 2004.

[53] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 387–394, 1990.

[54] A. Rosen. A note on constant-round zero-knowledge proofs for NP. In *Proceedings of the 1st Theory of Cryptography Conference*, pages 191–202, 2004.

[55] S. Rudich. *Limits on the provable consequences of one-way functions*. PhD thesis, EECS Department, University of California, Berkeley, 1988.

[56] A. Sahai and S. P. Vadhan. A complete problem for statistical zero knowledge. *Journal of the ACM*, 50(2):196–249, 2003.

[57] D. R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Advances in Cryptology - EUROCRYPT '98*, pages 334–345, 1998.

[58] H. Wee. One-way permutations, interactive hashing and statistically hiding commitments. In *Proceedings of the 4th Theory of Cryptography Conference*, pages 419–433, 2007.

[59] S. Wolf and J. Wullschleger. Oblivious transfer is symmetric. In *Advances in Cryptology - EUROCRYPT '06*, pages 222–232, 2006.

[60] A. C. Yao. How to generate and exchange secrets. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pages 162–167, 1986.