Finding Collisions in Interactive Protocols — Tight Lower Bounds on the Round and Communication Complexities of Statistically Hiding Commitments*

Iftach Haitner[†‡] Jonathan J. Hoch[§‡] Omer Reingold[¶] Gil Segev[‖‡]

January 15, 2015

Abstract

We study the round and communication complexities of various cryptographic protocols. We give tight lower bounds on the round and communication complexities of any fully black-box reduction of a statistically hiding commitment scheme from one-way permutations, and from trapdoor permutations. 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 and the reconstruction paradigm of Gennaro and Trevisan (FOCS '00).

Keywords: statistically hiding commitments; private information retrieval; one-way functions; black-box impossibility results.

*Preliminary versions of this work appeared as [36, 37].

[†]School of Computer Science, Tel Aviv University. E-mail: iftachh@cs.tau.ac.il. Research supported by ISF grant 1076/11, the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11), US-Israel BSF grant 2010196.

[‡]Part of this research was conducted while at the Weizmann Institute of Science.

[§]ImageSat Israel. E-mail: hoch@imagesatisrael.com.

[¶]Stanford University and Weizmann Institute of Science. E-mail: omer.reingold@gmail.com. Research partially supported by the DARPA PROCEED program

[‖]School of Computer Science and Engineering, Hebrew University of Jerusalem, Jerusalem 91904, Israel. Email: segev@cs.huji.ac.il. Research supported by the European Union's Seventh Framework Programme (FP7) via a Marie Curie Career Integration Grant, by the Israel Science Foundation (Grant No. 483/13), and by the Israeli Centers of Research Excellence (I-CORE) Program (Center No. 4/11).

Contents

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 [26, 53], pseudorandom generators [3, 41], universal one-way hash functions and signature schemes [56, 65], different types of commitment schemes [38, 39, 41, 54], private-key encryption [25] 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* and *communication* complexities of cryptographic protocols on one hand, and the strength of their underlying computational assumptions on the other. We provide lower bounds on the round and communication complexities of black-box reduction of statistically hiding and computationally binding commitment schemes (for short, statistically hiding commitments) from one-way permutations and from families of trapdoor permutations. Our lower bound matches known upper bounds resulting from [57]. 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.

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 commitments. A commitment scheme defines a two-stage interactive protocol between a sender S and a receiver R; informally, after the *commit stage*, S is bound to (at most) one value, which stays hidden from R, and in the *reveal stage* R learns this value. The two security properties hinted at in this informal description are known as *binding* (S is bound to at most one value after the commit stage) and *hiding* (R does not learn the value to which S commits before the reveal stage). In a *statistically hiding* 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.

Statistically hiding commitments can be used as a building block in constructions of statistical zero-knowledge arguments [6, 57] and of certain coin-tossing protocols [50]. When used within protocols in which certain commitments are never revealed, statistically hiding 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).

Statistically hiding commitments with a constant number of rounds were shown to exist based on specific number-theoretic assumptions [4, 6] (or, more generally, based on any collection of claw-

free permutations [28] with an efficiently recognizable index set [23]), and collision-resistant hash functions [11, 56]. Protocols with higher round complexity were shown to exist based on different types of one way functions. (The communication complexity of the aforementioned protocols varies according to the specific hardness assumption assumed). Protocols with $\Theta(n/\log n)$ rounds and $\Theta(n)$ communication complexity (where n is the input length of the underlying function) were based on one-way permutations [57] and (known-) regular one-way functions [35].[1] Finally, protocols with a polynomial number of rounds (and thus, polynomial communication complexity) were based on any one-way function [38, 39].[2]

Black-box reductions. As mentioned above, the focus of this paper is proving lower bounds on the round and communication complexity of various cryptographic constructions. In particular, showing that any construction of statistically hiding commitments based on trapdoor permutations requires a fairly large number of rounds. However, under standard assumptions (e.g., the existence of collision-resistant hash functions), *constant-round statistically hiding commitments do exist*. So if these assumptions hold, the existence of trapdoor permutations implies the existence of constant-round statistically hiding commitments in a *trivial logical sense*. Faced with similar difficulties, Impagliazzo and Rudich [43] 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 [46].

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 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 2.7.

1.1 Our Results

We study the class of fully black-box reductions of statistically hiding commitment schemes from families of trapdoor permutations, and prove lower bounds on the round and communication complexities of such constructions. Our lower bounds hold also for *enhanced* families of trapdoor permutations: one can efficiently sample a uniformly distributed public key and an element in the permutation's domain, so that inverting the element is hard, even when the random coins used for the above sampling are given as an auxiliary input. Therefore, the bounds stated below imply similar bounds for reduction from one-way permutations[3]. Informally, the round complexity lower bound is as follows:

[1] The original presentations of the above protocols have $\Theta(n)$ rounds. By a natural extension, however, the number of rounds in these protocols can be reduced to $\Theta(n/\log n)$, see [33, 47].

[2] When provided with a non-uniform advice, the round complexity of [39] reduces to $\Theta(n/\log n)$.

[3] In general, a black-box impossibility result that is proved w.r.t trapdoor permutations does not necessarily hold w.r.t. one-way permutation as the additional functionality (of having a trapdoor) may also be used by an attacker. Our result, however, holds even w.r.t. *enhanced* trapdoor permutations, and these can be used to simulate a one-way permutation by obliviously sampling a public key (the assumed obliviousness of the sampling algorithm enables to transform any attack against such a one-way permutation into an attack against the underlying enhanced trapdoor permutation).

Theorem 1 (The round complexity lower bound, informal)**.** *Any fully black-box reduction of a statistically hiding commitment scheme from a family of trapdoor permutations over* $\{0,1\}^n$ *has* $\Omega(n/\log n)$ *communication rounds.*[4]

The above lower bound matches the upper bound due to [34, 47] (the scheme of [57] has $\Theta(n)$ rounds), who give a fully black-box construction of an $n/(c \cdot \log n)$-round statistically hiding commitment scheme from one-way permutations over $\{0,1\}^n$, for any $c > 0$.[5] In addition, we note that our result and its underlying proof technique, in particular rule out fully black-box reductions of collision-resistant hash functions from one-way function. This provides an alternative and somewhat "cleaner" proof than that given by Simon [71] (although our proof applies to fully black-box reductions and Simon's proof applies even to semi black-box ones).

The separation oracle introduced for proving Theorem 1, yields the following lower bound on the communication complexity of statistically hiding commitments:

Theorem 2 (The communication complexity lower bound, informal)**.** *In any fully black-box reduction of a statistically hiding commitment scheme from family of trapdoor permutations over* $\{0,1\}^n$*, the sender communicates* $\Omega(n)$ *bits.*[6]

The above lower bound matches (up to a constant factor) the upper bound due to [57, 34, 47], who give a fully black-box reduction from a statistically hiding commitment scheme from a family of trapdoor permutations over $\{0,1\}^n$, where the sender sends $n-1$ bits. We remark, however, that the above bound says nothing about the number of bits sent by the *receiver*, a number which in the case of [57, 34, 47] is $\Theta(n^2)$, and thus dominates the overall communication complexity of the protocol. We also note that the above bound does not grow when the number of committed bits grows, and as such it only matches the bound of [57, 34, 47] when the number of bits committed is constant (when committing to k bits, the number of bits sent by the sender in [57, 34, 47] is $\Theta(nk)$ but this can be easily reduced to $\Theta(nk/\log n)$).

1.1.1 Any Hardness Reductions

We also consider a more general notion of hardness for trapdoor permutations that extends the standard polynomial hardness requirement; a trapdoor permutation τ over $\{0,1\}^n$ is s-hard, if any probabilistic algorithm running in time $s(n)$ inverts τ on a uniformly chosen image in $\{0,1\}^n$ with probability at most $1/s(n)$. We show that any fully black-box reduction of a statistically hiding commitment scheme from a family of s-hard trapdoor permutations requires $\Omega(n/\log s(n))$ communication rounds. This bound matches the any hardness reduction given in [34]. Interestingly, the communication complexity lower bound does not change when considering stronger trapdoor permutations.

1.1.2 Taking the Security of the Reduction into Account

The informal statements above consider constructions that invoke only trapdoor permutations over n bits. We would like to extend the result to consider constructions which may invoke the trapdoor

[4] The result holds even if the hiding is only guaranteed to hold against *honest receivers* — receivers that follow the prescribed protocol.

[5] Their proof of security reduction runs in time $\text{poly}(n) \cdot 2^{c \log n}$, and thus efficient only for constant c.

[6] The result holds even if the hiding is only guaranteed to hold against *honest receivers*, and the binding is only guaranteed to hold against *honest senders* — senders that follow the prescribed protocol in the commit stage.

permutations over more than a single domain. In this case, however, better upper bounds are known. In particular, given security parameter 1^n it is possible to apply the scheme of [57] using a one-way permutation over n^ϵ bits. This implies statistically hiding commitments of $\Theta(n^\epsilon)$ rounds, where the sender communicates $\Theta(n^\epsilon)$ bits. 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, 73]). The common approach for addressing this issue is by restricting the class of constructions (as in the informal statement of our main theorem above). We follow a less restrictive approach and consider constructions that are given access to trapdoor permutations over *any* domain size. 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 reduction gives a way to translate (in a black-box manner) an adversary $\widetilde{\mathsf{S}}$ that breaks the binding of the commitment scheme into an adversary A that breaks the security of the trapdoor permutation. Such a reduction is $\ell(n)$-security-parameter expanding, if whenever the machine A tries to invert a permutation over n bits, it invokes $\widetilde{\mathsf{S}}$ on security parameters which are at most $1^{\ell(n)}$. It should be noted that any reduction in which $\ell(n)$ is significantly larger than n, may only be weakly security preserving (for a taxonomy of security preserving reductions see [52, 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 theorems imply that the required number of rounds is $\Omega\left(n/\log n\right)$ and the number of bits send by the sender is $\Omega(n)$. In the general case (where $\ell(n)$ may be any polynomial in n), our theorems imply that the required number of rounds and the number of bits send by the sender is $n^{\Omega(1)}$ (which as argued above is tight as well).

1.1.3 Additional Implications

Our main results described above 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 round complexity or the communication complexity of the underlying protocol. Specifically, we derive similar lower bounds on the round complexity and communication complexity of fully black-box reductions from trapdoor permutations of single-server private information retrieval, interactive hashing, and oblivious transfer that guarantees statistical security for one of the parties. To obtain the above bounds we use known reductions from the listed primitives to statistically hiding commitment schemes. The only exception is the lower bound on the communication complexity on single-server private information retrieval. In this case, the parameters of known reduction (due to Beimel et al. [2]) fail too short to yield the desired lower bound, and we had to come up with a new reduction (given in Appendix A).

1.2 Related Work and Follow-Up Work

Impagliazzo and Rudich [43] showed that there are no black-box reductions of key-agrement protocols to one-way permutations and substantial additional work in this line followed (cf., [20, 69, 71]). Kim, Simon, and Tetali [46] initiated a new line of impossibility results, 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. Gennaro and Trevisan [16] has improved [46] to match the known upper bound, and their technique has yielded tight

lower bounds on the efficiency of several other black-box reductions [17, 18, 16, 42]. 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 statistically hiding commitments, Fischlin [15] showed that every black-box reduction of statistically hiding 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 [71]. Wee [73] considered a *restricted* class of black-box reductions of statistically hiding commitments to one-way permutations; informally, [73] considered 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 inputs) and does not perform any additional queries. Wee [73] showed that every black-box reduction of the above class has $\Omega\left(n/\log n\right)$ communication rounds. From the technical point of view, our techniques are inspired by those of Fischlin [15] and Wee [73] by significantly refining and generalizing the approach that an oracle-aided attacker can re-sample its view of the protocol (we refer the reader to Section 1.3 for more details on our approach).

The question of deriving lower bounds on the round complexity of black-box reductions, was also addressed in the context of zero-knowledge protocols [8, 13, 24, 30, 45, 66], to name a few. In this context, however, the black-box access is to the, possibly cheating, verifier and not to any underlying primitive.

Extensions in the spirit of the one we present here to the Gennaro and Trevisan [16] "reconstruction lemma", where used in several works, e.g., [32, 61, 62, 12]. In addition, the separation oracle "Sam" we present here (see Section 1.3), was found to be useful in other separation results [40, 29, 60, 67, 68, 5].

1.3 Overview of the Technique

For the sake of simplicity we concentrate below on the round complexity lower bound of fully black-box reductions of statistically hiding commitment from one-way permutations (see Section 1.3.6 for the communication complexity lower bound). We also assume that the sender's secret in the commitment protocol is a single uniform bit (i.e., it is a bit commitment). Let us start by considering Simon's oracle [71] for ruling out a black-box construction of a family of collision resistant hash functions from one-way permutations.

1.3.1 Simon's Oracle ColFinder

Simon's oracle ColFinder gets as an input a circuit C, possibly with π gates,[7] where π is a random permutation. It then outputs two elements w_1 and w_2 that are uniformly distributed subject to the requirement $C(w_1) = C(w_2)$.[8] Clearly, in the presence of ColFinder no family of collision resistant hash functions exists (the adversary simply queries ColFinder with the hash function circuit to find a collision). In order to rule out the existence of any two-round statistically hiding commitment scheme relative to ColFinder, Fischlin [15] used the following adversary $\widetilde{\mathsf{S}}$ to break any such scheme:

[7] In fact, ColFinder also accepts circuits C with ColFinder gates. [71] use this extension to give a *single* oracle with respect to which one-way permutations exist, but no collision resistance hash functions. Since the focus of our work is fully black-box reductions, we ignore this extension here and leave it as an open problem to extend our approach to the semi black-box setting.

[8] Consider, for example, sampling w_1 uniformly at random from the domain of C, and then sampling w_2 uniformly at random from the set $C^{-1}(C(w_1))$.

assume without loss of generality that the first message q_1 is sent by R and consider the circuit C_{q_1} defined by q_1 and S as follows: C_{q_1} gets as an input the random coins of S and outputs the answer that S replies on receiving the message q_1 from R. In the commit stage after receiving the message q_1, the cheating $\widetilde{\mathsf{S}}$ constructs C_{q_1}, queries $\mathsf{ColFinder}(C_{q_1})$ to get w_1 and w_2, and answers as $\mathsf{S}(w_1)$ would (i.e., by $C_{q_1}(w_1)$). In the reveal stage, $\widetilde{\mathsf{S}}$ 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 S's secret bit and the binding of the commitment will be violated.

In order to obtain the black-box impossibility results (both of [71] and of [15]), 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 (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).[9]

1.3.2 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, $\widetilde{\mathsf{S}}$ follows the protocol and let $q_1, \ldots, q_k$ be the messages that R sent in this stage. In the reveal stage, $\widetilde{\mathsf{S}}$ 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 S and $q_1, \ldots, q_k$ (i.e., $C_{q_1,\ldots,q_k}$ gets as an input the random coins of S and outputs the messages sent by S when 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 $\widetilde{\mathsf{S}}$ *already* gave in the commit stage (we did not encounter this problem when breaking two-round protocols, since $\widetilde{\mathsf{S}}$ could query $\mathsf{ColFinder}$ on C_{q_1} before $\widetilde{\mathsf{S}}$ sends its first and only message). Alternatively, we could have changed $\mathsf{ColFinder}$ 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.[10] We should not be too surprised that both the above attempts failed as they are both completely oblivious of the round complexity of (S, R). Since one-way permutations *do imply* statistically hiding commitments

[9] The proof of our main theorem (see intuition in Section 1.3.5), implies an alternative proof for the above claim.

[10] 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 $w_2 = (0, \pi^{-1}(y))$.

(in a black-box manner) [57, 35, 38, 39], any oracle that breaks statistically hiding commitments could also be used to break the underlying one-way permutations.[11]

So the goal is to extend Simon's oracle to handle interactions, while not making it "too strong" (so that it does not break the one-way permutations). In fact, the more interactive our oracle will be, the more powerful it will be (eventually, it will allow breaking the one-way permutations). Quantifying this growth in power is how we get the tight bounds on the round complexity of the reduction.

1.3.3 The Oracle Sam

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 with $C(w_1) = C(w_2)$. As explained above, an oracle for sampling a second preimage allows inverting the one-way permutations. What saves us in the case of ColFinder, is that w_1 was chosen by ColFinder *after* C is already given. Therefore, an adversary A is very limited in setting up the second distribution from which 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 ColFinder itself.

Extending the above interpretation of ColFinder, our separation oracle Sam is defined as follows: Sam is given as input a query $q = (w, C, C_{\text{next}})$ and outputs a preimage w', where w' is a uniformly distributed preimage of $C(w)$ (the purpose of the circuit C_{next} will be revealed later). In case $C = \perp$, algorithm Sam outputs a uniform element in the domain.

While the above Sam *can* be used for inverting random permutations when used by an *arbitrary* algorithm, it is not the case when used by low-depth *normal form* algorithms; an algorithm A is in a normal-form, if it makes the query $q = (w, C \neq \perp, C_{\text{next}})$ to Sam only if it has *previously* made the query $q' = (\cdot, \cdot, C)$ to Sam, and got w as the answer (namely, the third input to Sam is used for "committing" to C before seeing w).[12] A normal-form algorithm is of *depth* d if d is the length of the longest chain of Sam queries it makes (i.e., $\mathsf{Sam}(\cdot, \cdot, C_2) = w_2, \mathsf{Sam}(w_2, C_2, C_3) = w_3, \ldots, \mathsf{Sam}(w_d, C_d, \cdot) = w_{d+1}$). While restricted, it turns out that normal-form algorithms of depth $(d+1)$ are strong enough, with the aid of Sam, for breaking d-round statically hiding commitments.[13]

Assume there exists a fully black-box reduction from an $o(n/\log n)$-round statically hiding commitments to one-way permutations. By the above observation, the reduction should invert a random permutation when given oracle to an $o(n/\log n)$-depth normal-form algorithm $\widetilde{\mathsf{S}}$ with oracle access to Sam. Since the reduction has no direct access to Sam (but only via accessing $\widetilde{\mathsf{S}}$), it is easy to see that the reduction itself is an $o(n/\log n)$-depth normal-form algorithm. This implies a contradiction, since low-depth normal form algorithms cannot invert random permutations.

[11] In addition, in both these naive attempts the cheating sender $\widetilde{\mathsf{S}}$ follows the commit stage honestly (as 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-message variant of [57] where the receiver's queries are all sent in the first round).

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

[13] In the preliminary versions [36, 37], we equipped Sam with a signature-based mechanism to enforces normal-form behaviour and depth restriction on the queries it is asked upon. While yielding a simpler (and easier to comprehend) characterization of the power of Sam (i.e., useful for breaking commitments, not useful for inverting random permutations), the signature-based mechanism had significantly complicated the whole text.

1.3.4 A $(d+1)$-depth Normal-Form Algorithm that Breaks d-Round Commitments

Given a d-round statistically hiding commitment, the $(d+1)$-depth normal-forma algorithm $\widetilde{\mathsf{S}}$ for breaking the commitments operates as follows: after getting the first message q_1, it constructs C_{q_1} (the circuit that computes S's first message) and queries Sam on $(\perp, \perp, C_{q_1})$ to get input w_1, and sends $C_{q_1}(w_1)$ back to R. On getting the i'th receiver message q_i, the adversary $\widetilde{\mathsf{S}}$ constructs $C_{q_1,\ldots,q_i}$ (the circuit that computes S's first i messages), queries Sam on $(w_{i-1}, C_{q_1,\ldots,q_{i-1}}, C_{q_1,\ldots,q_i})$ to get w_i, and sends the i'th message of $C_{q_1,\ldots,q_i}(w_i)$ back to R. Finally, after completing the commit stage (when answering the last receiver message q_d) it queries Sam on $(w_d, C_{q_1,\ldots,q_d}, \perp)$ to get w_{d+1}. Since both w_d and w_{d+1} are sender's random inputs that are consistent with the commit-stage transcript, with probability roughly half they can be used for breaking the binding of the protocol.

1.3.5 Random Permutations Are Hard For $o(n/\log n)$-Depth Normal-Form Algorithms

To complete our impossibility result, it is left to prove that Sam cannot be used by $d(n) \in o(n/\log n)$-depth normal-form algorithms to invert a random permutation π. Let A be such a $o(n/\log n)$-depth normal-form algorithms. A Sam query $(\cdot, C, \cdot)$ is *y-hitting* (with respect to π), if it is answered with w, such that $C(w')$ queries π on $\pi^{-1}(y)$. Where A *hits* on input y, if it makes a y-hitting query. Given the above definition, our proof is two folded. We first show that a normal-form algorithm that hits on a random y with high probability, implies an algorithm that, with significant probability, inverts π *without hitting* (the proof of this part, influenced by the work of Wee [73], is the most technical part of the paper). We then extend the reconstruction technique of Gennaro and Trevisan [16], to show that a non-hitting algorithm is unlikely to invert π.

From normal-form hitting algorithms to non-hitting inverters. Let A be an algorithm that hits on a random y with high probability. The idea is that if $\mathsf{A}(y)$ hits, then it "knew" how to invert y *before* making the hitting Sam call. Assume for simplicity of notation that $\mathsf{A}(y)$'s queries are of the form $q_1 = (\perp, \perp, C_2), q_2 = (w_2, C_2, C_3), \cdots, q_d = (w_d, C_d, \cdot)$, where w_{i+1} is Sam answer on the query q_i (this essentially follows from A being in a d-depth normal form). And let i^* be such that q_{i^*} hits y with high probability (this follows from the assumption about A being a good hitter). Since $d \in o(n/\log n)$, there exists a location i such that the probability q_i to hit y is larger than the probability that q_{i-1} hits y by a arbitrary large polynomial. Further, an average argument yields that the probability that $C_i(w_i)$ queries π on $\pi^{-1}(y)$, is unlikely to be much smaller than the probability that q_i hits y (which is the probability that $C_i(w_{i+1})$ queries π on $\pi^{-1}(y)$).

Combining the above understandings, we design M that with non-negligible probability, inverts π on y without hitting. Algorithm M emulates A while following each Sam query (w_i, C_i, C_{i+1}) made by A receiving a reply w_{i+1}, it evaluates, in addition, $C_{i+1}(w_{i+1})$. If C_{i+1} queries π on $x = \pi^{-1}(y)$, then M halts and outputs x (otherwise, it continues with the emulation of A). We argue that with sufficiently large probability, if the first hinting query of A is $q_i = (w_i, C_i, C_{i+1})$, 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.

Random permutations are hard for non-hitting inverters. Gennaro and Trevisan [16] presented a very elegant argument for proving that random permutations are hard to invert also for non-uniform adversaries (previous proofs, e.g., [43], 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. [16] showed that π has a "short" description relative to A. (Intuitively, A saves on the description of π as it allows us to reconstruct π on (many of) the x's for which $\mathsf{A}^{\pi}(\pi(x)) = x$). Therefore, by a counting argument, there is only a tiny fraction of permutations which A inverts well.

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 $\mathsf{A}^{\pi}(\pi(x))$ (apart perhaps of the query for $\pi(x)$ itself) to already be reconstructed.

Consider an adversary A that, with significant probability, inverts π without hitting. Recall that when queried on $(w, C, \cdot)$, the oracle Sam returns a random inverse of $C(w)$. We would like to apply the argument of [16] to claim that relative to A and Sam there is a short description of π. We are faced with a substantial obstacle, however, as Sam might make a huge amount of π queries.[14] On the intuitive level, we overcome this obstacle by exploiting the fact that while Sam does not have an efficient *deterministic* implementation, it does have an efficient *non-deterministic* one: simply guess where the collision occur, and verify that this is indeed the case. Formalizing the above approach requires much care both in the definition and analysis of Sam, and critically use the assumption that A is non hitting. We defer more details to Section 5.3.

1.3.6 Low Sender-Communication Commitments

The lower bound for low sender-communication statistically hiding commitment follows from the fact that the oracle Sam, described above, can be used for breaking the binding of such commitments, and moreover, this can be done by low-depth normal-form algorithms. The idea is fairly straightforward; given an $o(n)$-communication commitment, the $o(n/\log n)$-depth normal-form algorithm $\widetilde{\mathsf{S}}$ for breaking the commitment, acts *honestly* in the commit stage (say by committing to zero), and only then uses Sam for finding decommitments to both zero and one.

Specifically, after the commit phase is over $\widetilde{\mathsf{S}}$ partitions trans into $d \in o(n/\log n)$ blocks $\mathsf{trans}_1, \ldots, \mathsf{trans}_d$, where trans_i contains the $(i-1)\cdot\log(n)+1, \cdots, i\cdot\log(n)$ bits sent by S (for simplicity we assume here that in each round the sender communicates a single bit to the receiver). Then $\widetilde{\mathsf{S}}$ iteratively applies Sam, such that after the i'th iteration, $\widetilde{\mathsf{S}}$ obtains random coins w_i that are consistent with $\mathsf{trans}_{1,\ldots,i}$. If successful, $\widetilde{\mathsf{S}}$ makes an additional call $(w_d, C_{q_1,\ldots,q_d}, \perp)$, where $C_{q_1,\ldots,q_d}$ is as in Section 1.3.4, to obtain additional coins w'_d consistent with trans, and then uses w_d and w'_d to break the commitment.

It is left to describe how $\widetilde{\mathsf{S}}$ obtains a consistent w_{i+1}, given that it has previously obtained a consistent w_i. For that, $\widetilde{\mathsf{S}}$ keeps calling Sam on $(w_d, C_{q_1,\ldots,q_i}, C_{q_1,\ldots,q_{i+1}})$ until it is replied with w_{i+1} that is consistent with $\mathsf{trans}_1, \ldots, \mathsf{trans}_{i+1}$. Since trans_{i+1} contains only $\log n$ bits sent by S, we expect $\widetilde{\mathsf{S}}$ to succeeds with high probability after about n such attempts.

1.4 Paper Organization

Notations and formal definitions are given in Section 2, where the oracle Sam and the separation oracle discussed above is formally defined in Section 3. In Section 4 we show how to use Sam (by normal form algorithms) to find collisions in any low-round complexity or low sender communication protocols, wherein Section 5 we show that in the hands of normal-form algorithms, Sam is not useful for inverting random permutations. In Section 6 we combine the above fact to derive our lower

[14] 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.

bounds on statistically hiding commitment schemes, where applications of the above results to other cryptographic protocols, are given in Section 7. Finally, in Appendix A we give a refined reduction of low-communication statistically hiding commitment schemes from low-communication single-server private information retrieval, that implies a lower bound of low-communication private information retrieval schemes.

2 Preliminaries

2.1 Conventions and Basic Notations

All logarithms are in base two. We use calligraphic letters to denote sets, uppercase for random variables, and lowercase for values. Let poly be the set of all polynomials $p : \mathbb{N} \to \mathbb{N}$. A function $\mu \colon \mathbb{N} \to [0,1]$ is *negligible*, denoted $\mu(n) = \text{neg}(n)$, if $\mu(n) < 1/p(n)$ for all $p \in \text{poly}$ and large enough n. For $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. For a finite set $\mathcal{X}$, denote by $x \leftarrow \mathcal{X}$ the experiment of choosing an element of $\mathcal{X}$ according to the uniform distribution, and by U_n the uniform distribution over the set $\{0,1\}^n$. Similarly, for a distribution D over a set $\mathcal{U}$, denote by $u \leftarrow D$ the experiment of choosing an element of $\mathcal{U}$ according to the distribution D. The statistical distance between two distributions P and Q over a set $\mathcal{U}$, denoted $\text{SD}(P,Q)$, is defined as $\frac{1}{2}\sum_{u\in\mathcal{U}} |\Pr_P[u] - \Pr_Q[u]|$. Given an event E, we denote by $\text{SD}(X,Y \mid \mathsf{E})$ the statistical distance between the conditional distributions $X|_{\mathsf{E}}$ and $Y|_{\mathsf{E}}$.

2.2 Algorithms and Circuits

Let PPTM stand for probabilistic algorithm (i.e., Turing machines) that runs in *strict* polynomial time. The input and output length of a circuit C, denoted $m(\mathsf{C})$ and $\ell(\mathsf{C})$, are the number of input wires and output wires in C respectively. Given a circuit family $\mathsf{C} = \{\mathsf{C}_n\}_{n\in\mathbb{N}}$ and input $x \in \{0,1\}^n$, let $\mathsf{C}(x)$ stands for $\mathsf{C}_n(x)$.

An oracle-aided algorithm A is an interactive Turing machines equipped with an additional tape called the *oracle tape*; the Turing machine can make a query to the oracle by writing a string q on its tape. It then receives a string *ans* (denoting the answer for this query) on the oracle tape. Giving a deterministic function $\mathcal{O}$, we denote by $\mathsf{A}^{\mathcal{O}}$ the algorithm defined by A with oracle access $\mathcal{O}$. The definition naturally extends to circuits, where in this case the circuit is equipped with *oracle gates*. In all the above cases, we allow the access to several different oracles, where this is nothing but a syntactic sugar to denote a *single* oracle that answers the queries it is asked upon by the relevant oracle, according to some syntax imposed on the queries (i.e., each query starts with a string telling the oracle it refers to).[15]

When dealing with an execution of an oracle-aided Turing-machines, we identify the queries according to their chronological order, where when dealing with circuits, we assume an arbitrary order that respects the topological structure of the circuit (i.e., a query asked in a gate of depth i, appears before any of the queries asked in gates of depth larger than i).

A q-query oracle-aided algorithm asks at most $q(n)$ oracle queries on input of length n, where in a q-query oracle-aided circuit family $\{\mathsf{A}_n\}_{n\in\mathbb{N}}$, the circuit C_n has at most $q(n)$ oracle gates.

[15] The above only consider "oracles" that implement *deterministic* functions. We will not consider random or state-full oracles.

An oracle-aided function mapping n-bit strings to $\ell(n)$-bit strings, stands for a *deterministic* oracle-aided algorithm that given access to any oracle and n-bit input, outputs $\ell(n)$-bit string.[16]

2.3 Interactive Protocols

A two-party protocol $\pi = (\mathsf{A}, \mathsf{B})$ is a pair of PPTM's. The communication between the Turing machines A and B is carried out in rounds. Each round consists of a message sent from A to B followed by a message sent from the B to A. We call π an m-round protocol, if for *every* possible random coins for the parties, the number of rounds is at *exactly* m. A communication transcript trans (i.e., the "transcript") is the list of messages exchanged between the parties in an execution of the protocol, where $\mathsf{trans}_{1,\ldots,j}$ denotes the first j messages in trans. A view of a party contains its input, its random tape and the messages exchanged by the parties during the execution. Specifically, A's view is a tuple $v_{\mathsf{A}} = (i_{\mathsf{A}}, r_{\mathsf{A}}, \mathsf{trans})$, where i_{A} is A's input, r_{A} are A's coins, and trans is the transcript of the execution. Let the random variable $\langle(\mathsf{A}(i_{\mathsf{A}}), \mathsf{B}(i_{\mathsf{B}})(i))\rangle$ denote the common transcript, the parties' local outputs and the parties's views in a random execution of $(\mathsf{A}(i_{\mathsf{A}}), \mathsf{B}(i_{\mathsf{B}})(i))$ (i.e., the private inputs of A and B are i_{A} and i_{B} respectively, and i is the common input). We naturally refer to the different parts of $\langle\cdot\rangle$ with $\langle\cdot\rangle_{\mathsf{trans}}$, $\langle\cdot\rangle_{\mathrm{out}_{\mathsf{A}}}$, $\langle\cdot\rangle_{\mathrm{out}_{\mathsf{B}}}$, $\langle\cdot\rangle_{\mathrm{view}^{\mathsf{A}}}$ and $\langle\cdot\rangle_{\mathrm{view}^{\mathsf{B}}}$, respectively.

The above notation naturally extends to oracle-aided protocols, where the main distinction is that the view of an oracle-aided party also contains the answers it got from the oracle.

2.4 Random Permutations and One-Way Permutations

For $n \in \mathbb{N}$, let Π_n be the set of all permutations over $\{0,1\}^n$, and let Π be the set all infinite collections $\pi = (\pi_1, \pi_2, \ldots)$ with $\pi_i \in \Pi_i$ for every $n \in \mathbb{N}$ (note that the set Π is not countable, and as a result our probability analysis in this paper deals with non-countable probability spaces). Our lower bound proof is based on analyzing random instances of such permutation collections.

Definition 3 (random permutations)**.** *A random choice of $\pi = \{\pi_n\}_{n\in\mathbb{N}}$ from Π, denoted $\pi \leftarrow \Pi$, means that π_n, for every $i \in \mathbb{N}$, is chosen uniformly at random and independently from Π_n.*

A collection of permutations is hard (i.e., one-way), if no algorithm can invert it with high probability.

Definition 4 (one-way permutations)**.** *A collection of permutations $\pi \in \Pi$ is $s(n)$-*hard*, if for every oracle-aided algorithm A of running time $s(n)$ and all sufficiently large n, it holds that*

$$\Pr_{y\leftarrow\{0,1\}^n}\left[\mathsf{A}^{\pi}(1^n, y) = \pi_n^{-1}(y)\right] \leq \frac{1}{s(n)},$$

where the probability is taken also over the random coins of A. The permutation π is polynomially-hard*, if it is $s(n)$-hard for some $s(n) = n^{\omega(1)}$.*

It is well known (cf., [18, 43]) that random permutations (and also random trapdoor permutations, see below) are hard to invert when given oracle access to the permutation.

[16] Since we only consider deterministic stateless oracles, for any fixing of the oracle, such algorithm indeed computes a function from n bits to $\ell(n)$ bits.

Theorem 5 ([18]). *For large enough $n \in \mathbb{N}$ and any $2^{n/5}$-query circuit C, it holds that*

$$\Pr_{\pi_n \leftarrow \Pi_n}\left[\Pr_{y \leftarrow \{0,1\}^n}\left[\mathsf{C}^{\pi_n}(y) = \pi_n^{-1}(y)\right] > 2^{-n/5}\right] < 2^{-2^{n/2}}.$$

In this paper we make a step further, showing that random permutations (and trapdoor random permutations) are to invert even in the presence of the *exponential-time* oracle Sam.

2.5 Random Trapdoor Permutations and One-Way Trapdoor 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 — $G(td) = pk$ and $F(pk, x) = y$, implies $F^{-1}(td, y) = x$.

Definition 6 (trapdoor permutations). *Let T the set of all function triplets $\tau = (G, F, F^{-1})$ with*

1. *$G \in \Pi$.*
2. *For every $n \in \mathbb{N}$ and $pk \in \{0,1\}^n$, the function F_{pk} over $\{0,1\}^n$ defined as $F_{pk}(x) = F(pk, x)$, is in Π_n.*
3. *For every $n \in \mathbb{N}$ and $sk, x \in \{0,1\}^n$, it holds that $F^{-1}(sk, F(G(sk), x)) = x$.*

A tuple $\tau \in \mathrm{T}$ is called a family of trapdoor permutations.

As in the case of standard permutations, we consider random instances of such trapdoor permutations collections.

Definition 7 (random trapdoor permutations). *A random choice $\tau = (G, F, F^{-1})$ from T, denoted $\tau \leftarrow \mathrm{T}$, means that $G \leftarrow \Pi$, and every $n \in \mathbb{N}$ and $pk \in \{0,1\}^n$, the permutation F_{pk}, defined in Definition 6, is chosen uniformly at random and independently from Π_n.*

A collection of trapdoor permutations is hard, if no algorithm, equipped with only the public key, can invert it with high probability.

Definition 8. *A family of trapdoor permutations $\tau = (G, F, F^{-1}) \in \mathrm{T}$ is $s(n)$-*hard*, if*

$$\Pr_{td \leftarrow \{0,1\}^n; y \leftarrow \{0,1\}^n}\left[\mathsf{A}^{\tau}(1^n, G(td), y) = F^{-1}(td, y)\right] \leq \frac{1}{s(n)},$$

for every oracle-aided algorithm A of running time $s(n)$ and all sufficiently large n, where the probability is also taken over the random coins of A. The family τ is polynomially hard*, if it is $s(n)$-hard for some $s(n) = n^{\omega(1)}$.*

Since we are concerned with providing a lower bound, we do not consider the most general definition of trapdoor permutations. (see [21] for such a definition). In addition, Definition 6 refers to the difficulty of inverting the permutation F_{pk} 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., [22, Appendix C]) that it is hard to invert F_{pk} on y even given

the random coins used in the generation of y. Our formulation captures such hardness condition, and therefore the impossibility results proved in this paper hold also for enhanced trapdoor permutations.[17] Finally, since the generator G of an s-hard trapdoor permutations family, of the above type, is an s-hard *one-way permutation* (i.e., no algorithm of running time $s(n)$ inverts with probability better than $1/s(n)$), the lower bounds we state here with respect to families of trapdoor permutations, yield analog bounds for one-way permutation.

2.6 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.

Definition 9 (bit-commitment scheme)**.** *A* bit-commitment scheme *is a triplet of* PPTM*'s* $(\mathsf{S},\mathsf{R},\mathsf{V})$ *such that*

$$\Pr_{(\mathsf{decom},\mathsf{com})\leftarrow\langle(\mathsf{S}(b),\mathsf{R})(1^n)\rangle_{\mathrm{out}_{\mathsf{S}},\mathrm{out}_{\mathsf{R}}}}[\mathsf{V}(\mathsf{com},\mathsf{decom})=b]=1,$$

for both $b\in\{0,1\}$ *and all* $n\in\mathbb{N}$.[18]

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* commitments.

Definition 10 (statistical hiding)**.** *Let* $\mathsf{Com}=(\mathsf{S},\cdot,\cdot)$ *be a bit-commitment scheme. For algorithm* $\widetilde{\mathsf{R}}$*, bit* b *and integer* n*, let* $\mathrm{Trans}^{\widetilde{\mathsf{R}}}(b,n)=\langle(\mathsf{S}(b),\mathsf{R})(1^n)\rangle_{\mathsf{trans}}$*. The scheme* Com *is* $\rho(n)$-hiding*, if* $\mathrm{SD}\left(\mathrm{Trans}^{\widetilde{\mathsf{R}}}(0,n),\mathrm{Trans}^{\widetilde{\mathsf{R}}}(1,n)\right)\leq\rho(n)$ *for* any *algorithm* $\widetilde{\mathsf{R}}$ *and large enough* n.[19] Com *is* statistically hiding*, if it is* $\rho(n)$*-hiding for some negligible function* $\rho(n)$*. When limiting the above to* $\widetilde{\mathsf{R}}=\mathsf{R}$*, then* Com *is called* honest-receiver $\rho(n)$-hiding/statistically hiding.

Definition 11 (computational binding)**.** *A bit-commitment scheme* $\mathsf{Com}=(\cdot,\mathsf{R},\mathsf{V})$ *is* $\mu(n)$-binding*, if*

$$\Pr\left[((decom,decom'),\mathsf{com})\leftarrow\left\langle(\widetilde{\mathsf{S}},\mathsf{R})(1^n)\right\rangle_{\mathrm{out}_{\widetilde{\mathsf{S}}},\mathrm{out}_{\mathsf{R}}}:\begin{array}{l}\mathsf{V}(\mathsf{com},\mathsf{decom})=0,\\ \mathsf{V}(\mathsf{com},\mathsf{decom}')=1\end{array}\right]<\mu(n)$$

[17] A different enhancement, used by [31], requires the permutations' domain to be polynomially dense in $\{0,1\}^n$. Clearly, our formulation is polynomially dense.

[18] Note that there is no loss of generality in assuming that the decommitment stage in non interactive. This is since any such interactive algorithm can be replaced with a non-interactive one as follows: let decom is the internal state of S when the decommitment starts, and let $\mathsf{V}(\mathsf{decom})$ simulate the sender and the interactive verifier in the interactive decommitment stage.

[19] It is more common to require that $\widetilde{\mathsf{R}}$'s views (and not the transcripts) are statistically close. Since. however, we put no restriction on the computation power of $\widetilde{\mathsf{R}}$, the two definitions are equivalent.

for any PPTM $\widetilde{\mathsf{S}}$ *and sufficiently large* n. Com *is* computationally binding, *if it is* $\mu(n)$*-binding [resp.,* honest-sender $\mu(n)$-binding*] for some negligible function* $\mu(n)$*, and is* weakly binding, *if it is* $(1-1/p(n))$*-binding for some polynomial* $p(n)$.

When limiting the above $\widetilde{\mathsf{S}}$ *that acts* honestly *in the commit stage,*[20] *then* Com *is called* honest-sender $\mu(n)$-binding/computationally binding/weakly binding.

2.7 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 et al. [64], 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 [64] for a more general definition.

Definition 12. *A fully black-box construction of weakly binding, statistically hiding commitment scheme from* $s(n)$*-hard family of trapdoor permutations, is a quadruple of oracle-aided* PPTM*'s* $(\mathsf{S},\mathsf{R},\mathsf{V},\mathsf{A})$ *such that the following hold:*

Correctness and hiding *The scheme* $\mathsf{Com}^{\tau}=(\mathsf{S}^{\tau},\mathsf{R}^{\tau},\mathsf{V}^{\tau})$ *is a correct, honest-receiver statistically hiding commitment scheme for every* $\tau\in\mathrm{T}$.

Black-box proof of binding: *For every* $\tau=(G,F,F^{-1})\in\mathrm{T}$ *and every algorithm* $\widetilde{\mathsf{S}}$ *such that* $\widetilde{\mathsf{S}}$ *breaks the weakly binding of* $(\mathsf{S}^{\tau},\mathsf{R}^{\tau},\mathsf{V}^{\tau})$*, according to Definition 11, it holds that*

$$\Pr_{td\leftarrow\{0,1\}^n;y\leftarrow\{0,1\}^n}\left[\mathsf{A}^{\tau}(1^n,G(td),y)=F^{-1}(td,y)\right]>\frac{1}{s(n)}$$

for infinitely many n*'s.*[21]

The construction is of honest-sender commitment, if the above only considers honest senders.

It would be useful for us to consider the following property of fully black-box reduction: consider a malicious sender $\widetilde{\mathsf{S}}$ 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

[20] I.e., in the commitment stage $\widetilde{\mathsf{S}}$ acts as $\widetilde{\mathsf{S}}(b;r)$, for some $b\in\{0,1\}$ and r that is uniformly chosen from the possible coins for S.

[21] A natural relaxation of Definition 12 is to consider the running time of the "security proof" A as an additional parameter. Allowing it, for instance, to run at exponential time when the trapdoor permutation of interest are "exponentially hard" (i.e., $s(n)=2^{cn}$). For the sake of presentation clarity, however, we chose no to consider such generalization.

parameter 1^n and invokes $\widetilde{\mathsf{S}}$ in a black-box manner. Definition 12, however, does not restrict the range of security parameters that A is allowed to invoke $\widetilde{\mathsf{S}}$ on. For example, A may invoke $\widetilde{\mathsf{S}}$ 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 13. *A black-box construction* $(\mathsf{S},\mathsf{R},\mathsf{V},\mathsf{A})$ *according to Definition 12 is* ℓ-security-parameter expanding*, if for every malicious sender* $\widetilde{\mathsf{S}}$*, the machine* A *on security parameter* 1^n *invokes* $\widetilde{\mathsf{S}}$ *on security parameter at most* $1^{\ell(n)}$*.*

3 The Oracle Sam and the Separation Oracle

In this section we describe the oracle that is later used for proving our lower bounds. The oracle is of the form $\left(\tau,\mathsf{Sam}^{\tau,h}\right)$, where τ is a family of trapdoor permutations (i.e., $\tau\in\mathrm{T}$), and $\mathsf{Sam}^{\tau,h}$ is an oracle that, very informally, receives as input a description of a circuit C (which may contain τ-gates) and a string w, and outputs (using h as its source of "randomness" as described below) a uniformly distributed preimage of $\mathsf{C}(w)$ under the mapping defined by C. For generality, we define Sam for an arbitrary oracle $\mathcal{O}$ and not necessarily for $\tau\in\mathrm{T}$. In Section 5 we use this generalization for first showing that Sam is not useful for inverting random permutations, and then use this result for proving that Sam is not useful for inverting random trapdoor permutations.

Moving to the formal description, a valid input (i.e., query) to Sam is a tuple of the form $(w,\mathsf{C},\mathsf{C}_{\text{next}})$, where C and C_{next} are oracle-aided circuits of the same input length m, and $w\in\{0,1\}^m$. The parameters C and w are allowed to (simultaneously) take the value $\perp$. Let $\mathcal{Q}$ stand for the family of all valid queries, and for $q=(w,\mathsf{C},\mathsf{C}_{\text{next}})\in\mathcal{Q}$ let $m(q)$ stand for the input length of C_{next}. Let $\mathcal{H}$ be the ensemble of permutation families $\left\{h=\{h_q\}_{q\in\mathcal{Q}}:h_q\in\Pi_{m(q)}\right\}$; that is, each $h\in\mathcal{H}$ is an infinite set of hash functions, indexed by $q\in\mathcal{Q}$. The definition yields that for $h\leftarrow\mathcal{H}$, the function h_q is uniformly random permutation over $\{0,1\}^{m(q)}$.We define Sam as follows.

Algorithm 14 (Sam)**.**

Input: $q=(w,\mathsf{C},\mathsf{C}_{\text{next}})\in\mathcal{Q}$.

Oracles: $\mathcal{O}$ *and* $h\in\mathcal{H}$.

Operation: Let $m=m(q)$.

- *If* $\mathsf{C}=\perp$*, output* $h_q(0^m)$.
- *Else, output* $h_q(v)$*, where* v *is the lexicographically smallest* $v\in\{0,1\}^m$ *with* $\mathsf{C}^{\mathcal{O}}(h_q(v))=\mathsf{C}^{\mathcal{O}}(w)$.

Sam answers arbitrarily on queries not in $\mathcal{Q}$. Note that the input parameter C_{next} was merely used to determine the value of m, but it will be crucial for the bookkeeping we employ below.

As mentioned in the introduction, algorithm Sam can be used for inverting *any* oracle, and thus there are no one-way function, or trapdoor permutation, relative to Sam. Below we define a restricted class of algorithms, called "normal form algorithms", for which Sam is not useful for inverting one-way functions, but is useful for breaking the binding of any low round-complexity, or low sender-communication complexity, commitment.

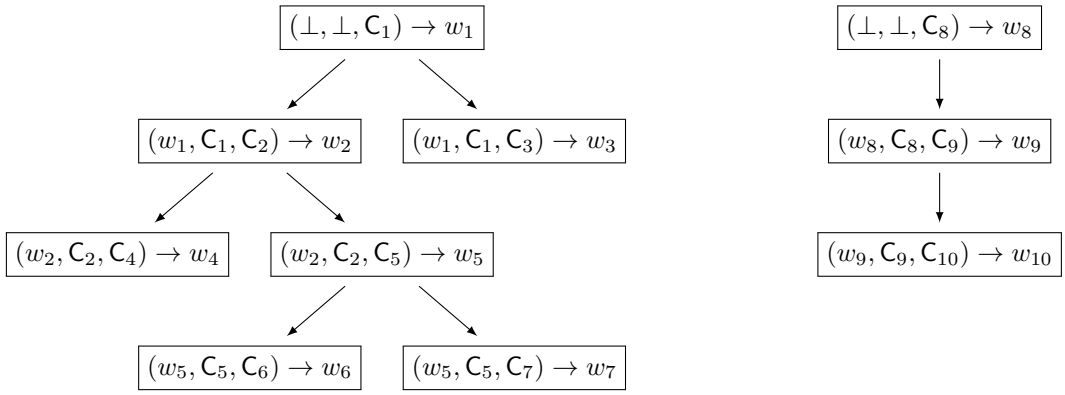

Figure 1: An example of a query forest that consists of two trees.

Normal form algorithms. Towards defining what normal form algorithms are, we associate the following structure with the queries Sam is asked upon (the reader is referred to Section 3 for a specific example).

Definition 15 (Query forest)**.** *Let $\overline{q}$ be an ordered list $\{q_1, w_1, \ldots, q_t, w_t\}$ of Sam queries/answers. A query $q_j = (\cdot, \cdot, \mathsf{C}) \in \overline{q}$ is the* parent*, with respect to $\overline{q}$, of all queries in $\overline{q}$ of the from $q_i = (w_j, \mathsf{C}, \cdot)$ with $i > j$, that do not have a lower index parent in $\overline{q}$. We let $\mathrm{p}(q) = q'$ denote that q' is the parent of q, and let $\mathrm{p}(q) = \bot$ in case q has no parent according to the above definition. The* depth *of $\overline{q}$ is the depth of the above forest. An oracle-aided algorithm A is of* query depth *d, denote a d-depth algorithm, if, when given access to Sam and an n-bit input, the resulting queries/answers list it makes to Sam is of depth at most $d(n)$.*

We also formally define what a "circuit extension" means.

Definition 16 (circuit extension)**.** *A circuit C' is a* extension *of an m-bit input, o-bit output circuit C, if C' has m input wires, and the function defined by the first o output wires of C' (assuming some arbitrary order on the wires) is identical to the function defined by the circuit C.*

Namely, a circuit C' is an "extension" of the circuit C, if it contains C as a "sub-circuit". Equipped with the above two definitions, we define normal form algorithms as follows.

Definition 17 (normal-form algorithms)**.** *An ordered list $\overline{q} = \{q_1, w_1, \ldots, q_t, w_t\}$ of Sam queries/answers is in a* normal form*, if $q_1, \ldots, q_t \in \mathcal{Q}$, exists no $i \neq j \in [t]$ with $q_i = (\cdot, \mathsf{C}_{\mathrm{next}})$ and $q_j = (\cdot, \mathsf{C}_{\mathrm{next}})$ for the* same *circuit $\mathsf{C}_{\mathrm{next}}$, and the following holds for every $q = (w, \mathsf{C} \neq \bot, \mathsf{C}_{\mathrm{next}}) \in \overline{q}$:*

1. *$\mathsf{C}_{\mathrm{next}}$ is an extension of C, and*
2. *$\mathrm{p}(q) \neq \bot$.*

An oracle-aided algorithm A is of a normal form, if, when given access to Sam, the resulting list of queries/answers it makes to Sam is always in normal form.

Note that in the query forest defined by a normal-form algorithms, the roots are all of the form $(\cdot, \perp, \cdot)$. The above definitions naturally extend to oracle-aided (families of) circuits, assuming a reasonable order on the circuit gates (see Section 2.2).

While restricted, normal-from algorithms are not at all useless. Specifically, combining the fully black-box reduction from $\Theta(n/\log n)$-round statistically hiding commitment to one-way permutation due to [33, 47] (extending [57]) and Theorem 21, yields the existence of an $\Theta(n/\log n)$-depth normal-form algorithm, that uses $\mathsf{Sam}^{\pi,\cdot}$ to invert *any* $\pi \in \Pi$. In contrast, in Section 5 we show that an $o(n/\log n)$-depth normal-form algorithm *cannot* invert a random $\pi \in \Pi$.

We will also note that an algorithm with oracle access (i.e., black-box access) to a normal-form algorithm, and without direct access to Sam, is a normal-form algorithm by itself.

Proposition 18. *let* A *be oracle-aided algorithm, let* B *be a* d*-depth normal-from algorithm, and let* C *be the algorithm that given oracle-access to* Sam*, acts as* $\mathsf{A}^{\mathsf{B}^{\mathsf{Sam}}}$ *(in particular* A *does not make direct calls to* Sam*). Then algorithm* C *is in a normal form. Assume further that on input of length* n*, algorithm* A *calls* B *on input of maximum length* $\ell(n)$*, then* C *is of depth* $d(\ell(n))$*.*

Proof. Since A accesses B in a black-box manner, the interaction of C with Sam is the combined (possibly partial) interactions of B with Sam done in this execution. Since B is in a normal form, the list of Sam queries/answers of each of these partial interactions is in a normal form. It follows that the joint list is in such a from, and therefore so is C.

The depth restriction of C immediately follows from the above observation, and the depth restriction of B. □

Augmented query complexity. We use the following measure for the query complexity of Sam-aided algorithms.

Definition 19 (augmented query complexity)**.** *The* augmented query complexity *of an algorithm* A *on input* x *with oracle access to* $\mathsf{Sam}^{\mathcal{O},h}$*, is the number of oracle calls that* A *makes, counting each call of the form* $\mathsf{Sam}(\cdot, \mathsf{C}, \cdot)$ *as* $t(\mathsf{C})$ *— the (standard) query complexity of* C*. Algorithm* A *is has* augmented query complexity t *(sometimes denoted,* A *is a* t*-augQueries algorithm), if on input of length* n*, and any choice of* $\mathcal{O}, h$*, it makes at most* $t(n)$ *augmented queries.*

Trivial circuit extension. While the above definition dictates Sam to use the *same* "randomness" when queried twice on the same query q (same function h_q is used), it is simple to effectively make Sam to use *independent* randomness on the "same" query (i.e., by making a dummy change, one that does not effect the circuit input/output behaviour, to the circuit part of q).

Definition 20 (trivial circuit extension)**.** *A circuit* C' *is a* trivial extension *of the circuit* C*, if both circuits computes the* same *function. For a circuit* C *and* $i \in \mathbb{N}$*, let* $\mathsf{ext}_i(\mathsf{C})$ *be the circuit* C *augmented with* i *OR gates that have no effect on the output (i.e., their output is ignored).*

Note that $\{\mathsf{ext}_i(\mathsf{C})\}_{i\in\mathbb{N}}$ are *distinct, trivial* extensions of C.

4 The Power of Sam

In this section we present normal-form algorithms that use Sam for finding collisions in *any* protocol of low round complexity, or of low communication complexity, aided with *any* oracle. Namely, a

cheating (normal-form) party can use Sam to interact with the other party such that the following hold: 1) at the end of the protocol the cheating party outputs several independent random inputs that are consistent with the execution of the protocol, and 2) the transcript of the resulting execution has the same distribution as of a random honest execution of the protocol. The case of low round complexity follows directly from the definition of Sam, where for the low communication complexity case we have to work slightly harder. Along the way, we show that Sam can be used to find collisions in any oracle-aided function with "short" outputs (i.e., the function output is significantly shorter than its input).

The depth parameter of the attackers presented below are functions of the round or communication complexity of the protocol, or of the function's output length. For being useful in applications such as the ones given in Sections 6 and 7, this parameter needs to be "small". Hence, the attacker described below are only useful for the type of protocols and functions we considered above.

4.1 Finding Collisions in Protocols of Low Round Complexity

In the following we focus on no-input protocols that get the security parameter 1^n as their common input.

Theorem 21. *For every d-round, t-query oracle-aided protocol (A,B), there exists a deterministic normal-from algorithm $\widetilde{\mathsf{A}}$ such that the following hold for every $n,k\in\mathbb{N}$ and function $\mathcal{O}$: let H be uniformly distributed over $\mathcal{H}$ and let $(\widetilde{\text{Trans}},(R_1,\ldots,R_k)) = \left\langle (\widetilde{\mathsf{A}}^{\mathcal{O},\mathsf{Sam}^{\mathcal{O},H}}(1^k),\mathsf{B}^{\mathcal{O}})(1^n)\right\rangle_{\mathsf{trans},\mathrm{out}^{\widetilde{\mathsf{A}}}}$,[22] then*

1. *$\widetilde{\text{Trans}}$ has the same distribution as $\text{Trans} = \left\langle (\mathsf{A}^{\mathcal{O}},\mathsf{B}^{\mathcal{O}})(1^n)\right\rangle_{\mathsf{trans}}$.*
2. *$R_1,\ldots,R_k$ are sampled independently from the distribution over the random coins of A that are consistent with $\mathcal{O}$, $\widetilde{\text{Trans}}$ and 1^n.*
3. *$\widetilde{\mathsf{A}}$ makes queries of depth at most $d(n)+1$.*
4. *$\widetilde{\mathsf{A}}$ makes $k+d(n)$ Sam-queries, all on $t(n)$-query circuits.*
5. *Assuming A is a PPTM, then $\widetilde{\mathsf{A}}$ runs in time $p(n)\cdot k$ for some $p\in\text{poly}$.*

Proof. For ease of notation, we assume that B sends the first message in (A,B). We fix n and k, and omit the security parameter 1^n whenever its value is clear from the context.

In order for $\widetilde{\mathsf{A}}$ to interact with Sam, it identifies A with the sequence of circuits $\mathsf{A}_1,\ldots,\mathsf{A}_d$ for which the following is an accurate description of A's actions: upon reliving the i'th message b_i from B, A sends $a_i=\mathsf{A}_i(r_{\mathsf{A}},(b_1,\ldots,b_i))$ to B, where r_{A} are A's random coins, and $b_1,\ldots,b_{i-1}$ are the first $i-1$ messages sent by B. We assume without loss of generality that each message a_i contains the previous messages $a_1,\ldots a_{i-1}$ as its prefix, and therefore each circuit A_i is an *extension* of A_{i-1} (as discussed in Section 3). Note that assuming A is a PPTM, then descriptions of the circuits $\mathsf{A}_1,\ldots,\mathsf{A}_d$ can be computed in polynomial-time from the description of A.

Given the above discussion, the oracle-aided interactive algorithm $\widetilde{\mathsf{A}}$ is defined as follows.

Algorithm 22 $(\widetilde{\mathsf{A}})$**.**

[22] I.e., the common transcript and A's output in a random execution of $(\widetilde{\mathsf{A}}^{\mathsf{Sam}^{\mathcal{O},h}}(1^k),\mathsf{B}^{\mathcal{O}})(1^n)$.

Input: 1^n *and* $k \in \mathbb{N}$.

Oracles: $\mathcal{O}$ *and* $\mathsf{Sam}^{\mathcal{O},h}$ *for some* $h \in \mathcal{H}$.

Operation:

Round $1 \leq i \leq d = d(n)$**:** *upon receiving the* i*'th message* b_i *from* B.

1. *Let* $\mathsf{A}_{b_1,\cdots,b_i}$ *be the circuit* A_i *defined above with* $(b_1,\cdots,b_i)$ fixed *as its second input (i.e., as* B*'s first* i *messages).*
2. *In case* $i = 1$ *(first round), set* $r_1 = \mathsf{Sam}^{\mathcal{O},h}(\perp,\perp,\mathsf{A}_{b_1})$.
   Otherwise, set $r_i = \mathsf{Sam}^{\mathcal{O},h}(r_{i-1},\mathsf{A}_{b_1,\cdots,b_{i-1}},\mathsf{A}_{b_1,\cdots,b_i})$.
3. *Send* $\mathsf{A}^{\mathcal{O}}_{(b_1,\cdots,b_i)}(r_i)$ *back to* B.

Output phase:

1. *For* $j \in [k]$ *set* $r_{d,j} = \mathsf{Sam}^{\mathcal{O},h}(r_d,\mathsf{A}_{b_1,\cdots,b_d},\mathsf{ext}_j(\mathsf{A}_{b_1,\cdots,b_d}))$.[23]
2. *Output* $r_{d,1},\ldots,r_{d,k}$.

Note that the only role of the circuits $\{\mathsf{ext}_j(\mathsf{A}_{b_1,\cdots,b_d})\}_{j\in[d]}$ used in the output phase of the above description of $\widetilde{\mathsf{A}}$, is causing Sam to use *independent* randomness per call (i.e., by using a different function from h). It is also easy to verify that $\widetilde{\mathsf{A}}$ queries Sam up to depth at most $d(n)+1$, performs at most $(d(n)+k)\cdot t(n)$ augmented oracle queries, and that $\widetilde{\mathsf{A}}$ runs in polynomial time (excluding the oracle calls) assuming that A is a PPTM. Since $\widetilde{\mathsf{A}}$ queries Sam up to depth $d(n)+1$ and the assumption that A_i is an extension of A_{i-1}, yields that $\widetilde{\mathsf{A}}$ is indeed in a normal form. The above observation yields that $r_{d,1},\ldots,r_{d,k}$ are independently distributed conditioned on trans, where each of them is uniformly distributed over the random coins of A that are consistent with $\mathcal{O}$ and trans – the transcript generated by the interaction of $(\widetilde{\mathsf{A}}^{\mathsf{Sam}^{\mathcal{O},H}}(1^k),\mathsf{B}^{\mathcal{O}})$. Hence, for completing the proof all we need to prove is that the transcript induced by a random execution of $(\widetilde{\mathsf{A}}^{\mathsf{Sam}^{\mathcal{O},H}}(1^k),\mathsf{B}^{\mathcal{O}})$, which we denote here by $\widetilde{\text{Trans}}$, has the same distribution as that induced by a random execution $(\mathsf{A}^{\mathcal{O}},\mathsf{B}^{\mathcal{O}})$, denoted here as Trans.

Claim 23. $\widetilde{\text{Trans}}$ *and* Trans *are identically distributed.*

Proof. Notice that in each round of the protocol, $\widetilde{\mathsf{A}}$ acts exactly like A would on the given (partial) transcript. That is, like A does on random coins that are sampled according to the right distribution: the distribution of A's coin in a random execution of (A,B) that yields this transcript. The formal (and somewhat tedious) proof follows.

The proof is by induction on i, the number of messages sent so far in the protocol, that $\widetilde{\text{Trans}}_{1,\ldots,i}$ and $\text{Trans}_{1,\ldots,i}$ are identically distributed. The base case $i = 0$ is trivial. In the following we condition on $\widetilde{\text{Trans}}_{1,\ldots,i} = \text{Trans}_{1,\ldots,i} = \mathsf{trans}$, and prove that under this conditioning $\widetilde{\text{Trans}}_{1,\ldots,i+1}$ and $\text{Trans}_{1,\ldots,i+1}$ are identically distributed.

Note that both in $(\widetilde{\mathsf{A}}^{\mathsf{Sam}^{\mathcal{O},h}},\mathsf{B}^{\mathcal{O}})$ and in $(\mathsf{A}^{\mathcal{O}},\mathsf{B}^{\mathcal{O}})$, the distribution of the (conditional) parties' joint view, is a *product* distribution. (This hold since the only oracle shared by the parties, i.e., $\mathcal{O}$,

[23] Recall that $\{\mathsf{ext}_j(\mathsf{A}_{b_1,\cdots,b_d})\}_{j\in[d]}$ are arbitrary *distinct* extensions of $\mathsf{A}_{b_1,\cdots,b_d}$. The role of these extensions is to make Sam to use fresh randomness in each call (i.e., to apply a different part of h).

is fixed.) In particular, the distribution of B's coins in both protocols is uniform over the possible coins for B that are consistent with $\mathcal{O}$ and trans (and the definition of B). Since the next message of a party is a deterministic function of $\mathcal{O}$, trans and its random coins, in case the $i+1$ message is in B's control, it holds that $\widetilde{\mathrm{Trans}}_{1,\ldots,i+1}$ and $\mathrm{Trans}_{1,\ldots,i+1}$ are identically distributed.

The complimentary case, where the $i+1$ message is in $\widetilde{\mathsf{A}}$'s or in A's control, is slightly more complicated. Note that the $i+1$ message sent by $\widetilde{\mathsf{A}}$ is determined by the value of r_{i+1}, returned by Sam, exactly in the same way that the $i+1$ message sent by A is determined by its random coins r_{A}; in both cases, the same deterministic function is applied to $\mathcal{O}$, trans and the coins. We complete the proof showing that r_{i+1} and r_{A} are identically distributed.

Similarly to the coins of B discussed above, r_{A} are uniformly distributed over the possible coins for A that are consistent with $\mathcal{O}$ and trans (and the definition of A). The value of r_{i+1} on the other hand, is determined by value of $H_{q_{i+1}}$, where q_{i+1} is the query $\widetilde{\mathsf{A}}$ makes to Sam in the $i+1$ round. Since $H_{q_{i+1}}$ was not queried by Sam in the first i rounds of $(\widetilde{\mathsf{A}},\mathsf{B})$, under the above the conditioning $H_{q_{i+1}}$ is a uniformly chosen permutation over the coins of A. Hence, the definition of Sam yields that, again, under the above conditioning, the coins it returns are uniformly distributed over the coins of A that are consistent with $\mathcal{O}$ and trans, yielding that r_{i+1} and r_{A} are identically distributed. □

□

4.2 Inverting Functions of Short Outputs

In this section we show how to use Sam to invert any function (i.e., deterministic algorithm) with oracle access to a trapdoor permutation oracle, given that the function output is "short". Combined with the results of Section 5, this would imply, for instance, that it is impossible to use in a fully black-box manner an n-bit one-way function to construct an $o(n)$-bit one-way function.[24]

Theorem 24. *For every t-query oracle-aided function $f\colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$, there exists a deterministic normal-from algorithm Inv such that the following holds for every $n,k,d \in \mathbb{N}$, $\varepsilon \in (0,1]$ and a function $\mathcal{O}$: let $(X_1,\ldots,X_k) = \mathsf{Inv}^{\mathcal{O},\mathsf{Sam}^{\mathcal{O},H}}(1^n,k,d,\varepsilon,f^{\mathcal{O}}(X))$, where X and H are uniformly chosen from $\{0,1\}^n$ and $\mathcal{H}$ respectively, then*

1. $\Pr[(X_1,\ldots,X_k) = \perp] \leq \varepsilon$.
2. *Conditioned on $(X_1,\ldots,X_k) \neq \perp$, the variables $X_1,\ldots,X_k$ are iid over $(f^{\mathcal{O}})^{-1}(f^{\mathcal{O}}(X))$.*
3. *Inv makes queries of depth at most $d+1$.*
4. *Inv makes at most $k + d \cdot 2^{\lceil \ell(n)/d \rceil}/\varepsilon$ Sam queries, all on t-query circuits.*
5. *Assuming f is polynomial-time computable, then Inv runs in time $p(n)\cdot(2^{\lceil \ell(n)/d \rceil} + k)$, for some $p \in \mathrm{poly}$.*

Proof. Fix n, k and $\mathcal{O}$. For ease of notation we and omit the security parameter 1^n and assume ℓ is a multiple of d. Let $v = \ell/d$, and for $x \in \{0,1\}^n$ and $i \in [v]$, let $f(x)_{(i)}$ denote the i'th block of $f(x)$, i.e., $f(x)_{(i-1)d+1,\ldots,id}$. For $i \in [d]$, let f_i be the circuit that on input $x \in \{0,1\}^n$ outputs

[24] Note that the following theorem does not stand in contradicting with the one-wayness of a random permutation in the presence of Sam, proved in Section 5. The functions in consideration there have long outputs.

$f(x)_{(1)}, \ldots, f(x)_{(i)}$. We invert f on $y = (y_1, \ldots, y_d) \in (\{0,1\}^v)^d$, by gradually causing Sam to output x_i with $f_i(x_i) = (y_1, \ldots, y_i)$ for $i = 1$ to d. Doing that for $i = 1$ is easy: keep calling Sam on input $(\perp, \perp, f_1)$, until it returns x_1 with $f_1(x_1) = y_1$. Since a call to $\mathsf{Sam}(\perp, \perp, f_1)$ returns *uniform and independent* element in $\{0,1\}^n$, about 2^v Sam calls yield the desired answer. Assuming that we have successfully made Sam to answer on $(\cdot, \cdot, f_{i-1})$ with x_{i-1} such that $f_{i-1}(x_{i-1}) = (y_1, \ldots, y_{i-1})$, we make Sam answer with x_i such that $f_i(x_i) = (y_1, \ldots, y_i)$ using similar means to the ones used to get x_i; keep calling Sam on input (x_{i-1}, f_{i-1}, f_i), until it returns the right x_i. As in the first round, about 2^v Sam calls suffices to get the desired answer. The formal definition of algorithm Inv is given below.

Algorithm 25 (Inv)**.**

Input: 1^n, $k, d \in \mathbb{N}$, $\varepsilon \in (0,1]$ *and* $y = (y_1, \ldots, y_d) \in (\{0,1\}^v)^d$.

Oracles: $\mathcal{O}$ *and* $\mathsf{Sam}^{\mathcal{O},h}$, *for some* $h \in \mathcal{H}$.

Operation:

1. *For* $i = 1$ *to* d *do:*

   Set $j = 0$, *and do the following loop:*

   (a) j^{++}.

   (b) *Let* $x_i = \mathsf{Sam}^{\mathcal{O},h}(x_{i-1}, f^*_{i-1}, \mathsf{ext}_j(f_i))$. *In the case* $i = 1$, *set* $f^*_{i-1} = x_{i-1} = \perp$.

   (c) *If* $f_i(x_i) = (y_1, \ldots, y_i)$, *set* $f^*_i = \mathsf{ext}_j(f_i)$ *and break the inner loop.*

   (d) *If* overall *number of* Sam *calls exceeds* $d \cdot 2^v/\varepsilon$, *return* $\perp$ *and abort.*

2. *For* $j = 1$ *to* k: *set* $x_{d,j} = \mathsf{Sam}^{\mathcal{O},h}(x_d, f^*_d, \mathsf{ext}_j(f^*_d))$.

3. *Return* $x_{d,1}, \ldots, x_{d,k}$.

The second and third properties of Inv immediately follow from the definition of Sam, so the only interesting part is showing that Inv aborts (i.e., outputs $\perp$) with probability at most ε. Let $\widetilde{\mathsf{Inv}}$ be the unbounded version of Inv, i.e., Step 1.(d) is removed. It is clear that $\widetilde{\mathsf{Inv}}$'s output is identical to that of Inv conditioned on Inv not aborting, and that the probability that Inv aborts is the probability that $\widetilde{\mathsf{Inv}}$ make more than $d \cdot 2^v/\varepsilon$ Sam calls. We show that the expected number of Sam calls made by $\widetilde{\mathsf{Inv}}$ is bounded by $d \cdot 2^v$, and proof follows by a Markov bound.

We bound the expected number of overall Sam calls made by $\widetilde{\mathsf{Inv}}$ in a single round of Step 1, and the proof follows by linearity of expectation. Fix a value for $y_1, \ldots, y_{i-1}$. Let $Y = f(X)_{(i)}$ conditioned that $f_{i-1}(X) = y_1, \ldots, y_{i-1}$, and let Y_j be the value of $f(x_i)_{(i)}$ sampled in the j'th inner loop of a random execution of $\widetilde{\mathsf{Inv}}(k, d, \varepsilon, y_1, \ldots, y_{i-1}, \ldots)$. If less than j inner loops happen, we let Y_j be an independent copy of Y_1. The definition of Sam yields that over a random choice of h, the variables $Y, Y_1, Y_2, \ldots$ are iid over $\{0,1\}^v$. It follows that $\Pr[Y = Y_j] \geq 2^{-v}$ for every j, and the expected value of the first j with $Y_j = Y$ is bounded by 2^v. Hence, the expected number of Sam calls made by $\widetilde{\mathsf{Inv}}$ (over the choice of X and h) is bounded by $d \cdot 2^v$. □

4.3 Finding Collisions in Low Communication Complexity Protocols

The following theorems show how to find collision in protocols in which the communication of the "attacking" party is low.

Theorem 26. *Let $\pi = (\mathsf{A}, \mathsf{B})$ oracle-aided protocol in which A, on input of length n, makes at most $t(n)$ oracle-queries and sends at most $c(n)$ bits. Then there exists a deterministic normal-from algorithm Inv such that the following holds for every $n, k, d \in \mathbb{N}$, $\varepsilon \in (0,1]$ and function $\mathcal{O}$: let H be uniformly distributed over $\mathcal{H}$ and let $(X_1, \ldots, X_k) = \mathsf{Inv}^{\mathcal{O}, \mathsf{Sam}^{\mathcal{O},H}}(1^n, 1^k, d, \varepsilon, \mathrm{Trans})$, where $\mathrm{Trans} = \langle \pi^{\mathcal{O}}(1^n) \rangle_{\mathsf{trans}}$, then*

1. $\Pr[(X_1, \ldots, X_k) = \perp] \leq \varepsilon$,
2. *Conditioned on $(X_1, \ldots, X_k) \neq \perp$, the variables $X_1, \ldots, X_k$ are iid over the random coins of A that are consistent with* Trans.
3. *Inv makes queries of depth at most $d+1$, and*
4. *Inv makes at most $k + d \cdot 2^{\lceil c(n)/d \rceil}/\varepsilon$ Sam-queries, all on $t(n)$-query circuits.*
5. *Assuming that π is polynomial-time computable, then Inv runs in time $p(n) \cdot (2^{\lceil c(n)/d \rceil} + k)$, for some $p \in \mathrm{poly}$.*

Remark 27 (Comparing Theorem 26 to Theorem 21)**.** *Both Theorem 21 and Theorem 26 are useful for finding collisions in the given protocols. While the attacker of Theorem 21 never fails, the attacker of Theorem 26 (who might fail) has the advantage of not using Sam through the execution, but only after it ends. We use this property in Section 6.2 to rule out constructions of* honest-sender *low sender-communication commitments from trapdoor functions.*

Proof of Theorem 26. We start by assuming that B is deterministic. Let $f\colon \{0,1\}^n \mapsto \{0,1\}^{c(n)}$ map A's random coins to the messages it send to B in π. Consider the algorithm Inv_D that on input (x, trans) returns $\mathsf{Inv}_f(x, \mathsf{trans}_{\mathsf{A}})$ (with the same oracles), for Inv_f being the inverter Theorem 24 guarantees for the function f, and $\mathsf{trans}_{\mathsf{A}}$ being A's part in trans. By Theorem 24, algorithm Inv_D satisfies the first three and fifth properties, stated in the theorem, and makes at most $k + d \cdot 2^{\lceil c(n)/d \rceil}/\varepsilon$ Sam-queries. Algorithm Inv_D, however, might apply Sam on circuits of query complexity larger than $t(n)$ (as they contain the queries made by B).

Consider the following variant of Inv_D. For a transcript trans of π, let $g_{\mathsf{trans}}\colon \{0,1\}^n \mapsto \{0,1\}^{c(n)}$ map A's random coins to the messages it sends to B in π, assuming that B sends A the message it sends in trans. On input (x, trans), algorithm Inv returns $\mathsf{Inv}_{g_{\mathsf{trans}}}(x, \mathsf{trans}_{\mathsf{A}})$ (with the same oracles), for $\mathsf{Inv}_{g_{\mathsf{trans}}}$ being the inverter Theorem 24 guarantees for the function g_{trans}, and $\mathsf{trans}_{\mathsf{A}}$ being A's part in trans. The point to notice is that by construction, on the same input and a random choice of h, algorithms Inv and Inv_D have *exactly* the same output distribution. In follows that Inv satisfies all the properties satisfied by Inv_D, where by construction, on only invoke Sam on $t(n)$-query circuits. Furthermore, since the implementation of Inv in obvious to the definition of B, it has the same success probability also when considering a probabilistic B. □

5 Random Permutations are Hard for Low-Depth Normal-Form Algorithms

In this section we prove that for low-depth normal-from algorithms, Sam is not useful for inverting random permutations and random trapdoor permutations. We start with random permutations, and then extend the result to random trapdoor permutations.

Following [18], we state our results in the stronger non-uniform setting. Hence, our goal is to upper bound the success probability of a circuit family having oracle access to Sam in the task of inverting a uniformly chosen permutation $\pi \in \Pi$ on a uniformly chosen image $y \in \{0,1\}^n$. We relate this success probability to the maximal depth of the Sam-queries made by the circuit family and to the augmented query complexity of the family (see Definition 19). We prove the following theorem.

Theorem 28. *The following holds for large enough $n \in \mathbb{N}$: for every t-augQueries, d-depth, normal-form circuit A such that $t^{3d+1} < 2^{n/8}$, it holds that*

$$\Pr_{\substack{\pi \leftarrow \Pi, h \leftarrow \mathcal{H} \\ y \leftarrow \{0,1\}^n}} \left[\mathsf{A}^{\pi, \mathsf{Sam}^{\pi,h}}(y) = \pi^{-1}(y) \right] \leq 2/t.$$

Before turning to prove Theorem 28, we first provide a brief overview of the structure of the proof. Consider a normal-from circuit A trying to invert an input $y \in \{0,1\}^n$ (i.e., to find $\pi^{-1}(y)$), while having oracle access to both π and Sam. We distinguish between two cases: one in which A obtains information on the value $\pi^{-1}(y)$ via one of its Sam-queries, and the other in which none of A's Sam-queries provides sufficient information for retrieving $\pi^{-1}(y)$. Specifically, we define:

Definition 29 (Hits)**.** *An execution $\mathsf{A}^{\pi, \mathsf{Sam}^{\pi,h}}(y)$ is* hitting*, denoted by the event $\mathsf{Hit}_{\mathsf{A},\pi,h}(y)$, if A makes a Sam-query $q = (\cdot, \mathsf{C}, \cdot)$, replied with w such that the computation $\mathsf{C}^{\pi}(w)$ queries π on $\pi^{-1}(y)$.*

The proof proceeds in two modular parts. In the first part of the proof, we consider the case that the event $\mathsf{Hit}(y) = \mathsf{Hit}_{\mathsf{A},\pi,h}(y)$ does not occur, and prove a "reconstruction lemma" that extends an information-theoretic argument of Gennaro and Trevisan [16]. They showed that if a circuit A manages to invert a permutation π on a relatively large set of images, then this permutation has a rather short representation given A. We generalize their argument to deal with circuits having oracle access to Sam. In this part we do not restrict the depth of A, neither require it to be in a normal form.

Lemma 30. *The following holds for large enough $n \in \mathbb{N}$: let A be a $2^{n/5}$-augQueries circuit, then*

$$\Pr_{\pi \leftarrow \Pi, h \leftarrow \mathcal{H}} \left[\Pr_{y \leftarrow \{0,1\}^n} \left[\mathsf{A}^{\pi, \mathsf{Sam}^{\pi,h}}(y) = \pi^{-1}(y) \ \wedge \neg \mathsf{Hit}_{\mathsf{A},\pi,h}(y) \right] \geq 2^{-n/5} \right] \leq 2^{-2^{\frac{3n}{5}}}.$$

Namely in the "non-hitting case", oracle access to Sam does not improve ones chances to invert a random permutation.

In the second part of the proof, we show that the case where the event $\mathsf{Hit}(y)$ does occur, can be reduced to the case where the event $\mathsf{Hit}(y)$ does not occur. 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. For this part, the query complexity of the circuit, its depth restriction and it being in a normal form, all play an instrumental role.

Lemma 31. *For every t-augQueries, d-depth normal-from circuit A there exists a $2t$-augQueries circuit M such that the following holds: assuming that*

$$\Pr_{\substack{\pi\leftarrow\Pi,h\leftarrow\mathcal{H}\\ y\leftarrow\{0,1\}^n}}\left[\mathsf{Hit}_{\mathsf{A},\pi,h}(y)\right]\geq\varepsilon$$

for $\varepsilon\in[0,1/t]$, then

$$\Pr_{\substack{\pi\leftarrow\Pi,h\leftarrow\mathcal{H}\\ y\leftarrow\{0,1\}^n}}\left[\mathsf{M}^{\pi,\mathsf{Sam}^{\pi,h}}(y)=\pi^{-1}(y)\wedge\neg\mathsf{Hit}_{\mathsf{M},\pi,h}(y)\right]\geq(\varepsilon/2)^{3d+1}.$$

In what follows we show that Theorem 28 is a straightforward corollary of Lemmas 30 and 31. In Section 5.1 we extend our statement to deal with trapdoor permutations. Then, in Sections 5.2 and 5.3 we prove Lemmas 30 and 31, respectively.

Proof of Theorem 28. Assume towards a contradiction that for infinitely many n's, there exists a t-query, d-depth normal-from circuit A such that $t^{3d+1}<2^{n/8}$ and

$$\Pr_{\substack{\pi\leftarrow\Pi,h\leftarrow\mathcal{H}\\ y\leftarrow\{0,1\}^n}}\left[\mathsf{A}^{\pi,\mathsf{Sam}^{\pi,h}}(y)=\pi^{-1}(y)\right]\geq 2/t.$$

Consider now the circuit A' that emulates A and makes sure that whenever A inverts y then the event $\mathsf{Hit}_{\mathsf{A}',\pi,h}(y)$ occurs. Note that A' can be easily implemented based on A by performing two additional queries to Sam (containing a circuit with a π-gate that has hardwired the output of A). Thus, A' is a $(t+2)$-query d-depth normal-from circuit, and it holds that

$$\Pr_{\substack{\pi\leftarrow\Pi,h\leftarrow\mathcal{H}\\ y\leftarrow\{0,1\}^n}}\left[\mathsf{Hit}_{\mathsf{A}',\pi,h}(y)\right]\geq 2/t.$$

Lemma 31 implies that for infinitely many n's there exists an $(2(t+2)\leq 2^{n/7})$-augQueries circuit M such that

$$\Pr_{\substack{\pi\leftarrow\Pi,h\leftarrow\mathcal{H}\\ y\leftarrow\{0,1\}^n}}\left[\mathsf{M}^{\pi,\mathsf{Sam}^{\pi,h}}(y)=\pi^{-1}(y)\wedge\neg\mathsf{Hit}_{\mathsf{M},\pi,h}(y)\right]\geq\left(\frac{1}{t}\right)^{3d+1}>\frac{1}{2^{n/8}},$$

in contradiction to Lemma 30. □

5.1 Extension to Trapdoor Permutations

We prove the following theorem:

Theorem 32. *For t-augQueries, d-depth normal-form circuit A with $(3t)^{3d+1}<2^{n/8}$ and large enough n, it holds that*

$$\alpha:=\Pr_{\substack{\tau=(G,F,F^{-1})\leftarrow\mathrm{T},h\leftarrow\mathcal{H}\\ td\leftarrow\{0,1\}^n,y\leftarrow\{0,1\}^n}}\left[\mathsf{A}^{\tau,\mathsf{Sam}^{\tau,h}}(G(td),y)=F^{-1}(td,y)\right]\leq 4/t.$$

Assume A inverts F with probability $5/t$. If A queries $F^{-1}(td,\cdot)$ with probability $2.5/t$, then it can be used to invert (the random permutation) G with this probability, in contradiction to Theorem 28. If the latter does not happen, then A inverts F with probability $2.5/t$ *without* using F^{-1}, which is again in contradiction to Theorem 28. Formal proof follows.

Proof. Let n be sufficiently large as required for Theorem 28. For $\tau \in \mathrm{T}$ and $td \in \{0,1\}^n$, let $\tau_{\neg td}$ be the variant of τ that answers on queries of the form $F^{-1}(td,\cdot)$ with $\perp$. We claim that

$$\beta := \Pr_{\substack{\tau=(G,F,F^{-1})\leftarrow \mathrm{T}, h\leftarrow\mathcal{H} \\ td\leftarrow\{0,1\}^n, y\leftarrow\{0,1\}^n}} \left[\mathsf{A}^{\tau_{\neg td},\mathsf{Sam}^{\tau_{\neg td},h}}(G(td),y) = F^{-1}(td,y)\right] > \alpha - 2/t. \tag{1}$$

Assuming Equation (1) holds, then it still holds for some fixing of G, td and $\left\{F_{pk'}\right\}_{pk'\neq G(tk)}$. Hence, there exists a t-augQueries, d-depth normal-form circuit B, with the above fixing "hardwired" into it, such that

$$\Pr_{\substack{F_{pk}\leftarrow\Pi, h\leftarrow\mathcal{H} \\ y\leftarrow\{0,1\}^n}} \left[\mathsf{B}^{F_{pk},\mathsf{Sam}^{F_{pk},h}}(\pi(y)) = (F_{pk})^{-1}(y)\right] > \alpha - 2/t,$$

and Theorem 28 yields that $\alpha \leq 4/t$.

The rest of the proof is devoted for proving Equation (1). The augmented queries made by an algorithm with oracle to Sam, are those queries made by the algorithm directly, plus those queries made by $\mathsf{C}(w)$ and $\mathsf{C}(w')$, for each query $w' = \mathsf{Sam}(w,\mathsf{C},\cdot)$ made by the algorithm. It is easy to verify that

$$\Pr_{\substack{\tau=(G,F,F^{-1})\leftarrow \mathrm{T}, h\leftarrow\mathcal{H} \\ td\leftarrow\{0,1\}^n, y\leftarrow\{0,1\}^n}} \left[\mathsf{A}^{\tau_{\neg td},\mathsf{Sam}^{\tau_{\neg td},h}}(G(td),y) \text{ makes an augmented query } F^{-1}(td,\cdot)\right] \geq \alpha - \beta \tag{2}$$

For a circuit C and $pk \in \{0,1\}^n$, let C^{pk} be the variant of C that before each query of the form $F^{-1}(td,\cdot)$, it queries G on td, and if the answer is pk, it replies to the query F^{-1} with $\perp$ (without making the call). Let D the variant of A^{pk} that on input (pk,y), replaces each Sam query $(\cdot,\mathsf{C},\mathsf{C}_{\text{next}})$ done by A^{pk}, with the query $(\cdot,\mathsf{C}^{pk},\mathsf{C}^{pk}_{\text{next}})$. That is, $\mathsf{D}^{\tau,\mathsf{Sam}^{\tau,h}}(G(td),y)$ emulates $\mathsf{A}^{\tau_{\neg td},\mathsf{Sam}^{\tau_{\neg td},h}}(G(td),y)$. It follows that

$$\Pr_{\substack{\tau=(G,F,F^{-1})\leftarrow \mathrm{T}, h\leftarrow\mathcal{H} \\ td\leftarrow\{0,1\}^n, y\leftarrow\{0,1\}^n}} \left[\mathsf{D}^{\tau,\mathsf{Sam}^{\tau,h}}(pk = G(td),y) \text{ makes the augmented query } G(td)\right] \geq \alpha - \beta \tag{3}$$

Let E be the variant of D, that if one of its augmented queries is of the form $G(td') = pk$, it halts and return td'. It is clear that

$$\Pr_{\substack{\tau=(G,F,F^{-1})\leftarrow \mathrm{T}, h\leftarrow\mathcal{H} \\ td\leftarrow\{0,1\}^n, y\leftarrow\{0,1\}^n}} \left[\mathsf{E}^{\tau,\mathsf{Sam}^{\tau,h}}(pk = G(td),y) = td\right] \geq \alpha - \beta \tag{4}$$

In particular, there exists a *fix* value (y,F) for which the above holds with respect to this fixing.[25] Let I be the function that inverts F given only the public key. That is, $I(pk,y) = (F_{pk})^{-1}$ (recall that $F_{pk}(y) = F(pk,y)$). Let M the variant of E with this fixed value of (y,F,I) "hardwired" into it that replaces each call $F^{-1}(td',y')$ made by E, with $I(G(td',y'))$. It is clear that

$$\Pr_{\substack{G\leftarrow\Pi, h\leftarrow\mathcal{H} \\ td\leftarrow\{0,1\}^n}} \left[\mathsf{M}^{G,\mathsf{Sam}^{G,h}}(G(td)) = G^{-1}(td)\right] \geq \alpha - \beta,$$

and, by inspection, M is a $3t$-augQueries, d-depth normal-form circuit. Theorem 28 yields that $\alpha - \beta \leq 2/t$, and Equation (1) follows. □

[25] By the definition of T, such fixing does not change the distribution of G (i.e., G is a uniform random permutation giving this fixing).

5.2 The Reconstruction Lemma — Proving Lemma 30

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.

Claim 33. *There exists a deterministic algorithm* $\mathsf{Decoder}$ *such that the following holds for every t-augQueries circuit* A*,* $\pi \in \Pi$*,* $h \in \mathcal{H}$ *and* $n \in \mathbb{N}$*. Assuming that*

$$\Pr_{y \leftarrow \{0,1\}^n}\left[\mathsf{A}^{\pi,\mathsf{Sam}^{\pi,h}}(y) = \pi_n^{-1}(y) \wedge \neg \mathsf{Hit}_{\mathsf{A},\pi,h}(y)\right] \geq \epsilon,$$

then there exists an $\left(2\log\binom{2^n}{a} + \log((2^n-a)!)\right)$*-bit string* aux*, such that* $\mathsf{Decoder}(\mathsf{aux},\mathsf{A},h,\pi_{-n}) = \pi_n$*, where* $a \geq \epsilon 2^n/(2t)$ *and* $\pi_{-n} = \{\pi_i\}_{i\in\mathbb{N}\setminus\{n\}}$*.*

Proof. Denote by $\mathcal{I} \subseteq \{0,1\}^n$ the set of points $y \in \{0,1\}^n$ on which $\mathsf{A}^{\pi,\mathsf{Sam}^{\pi,h}}$ successfully inverts π_n with no y-hits. We claim that there exists a relatively large set $\mathcal{Y} \subseteq \mathcal{I}$, such that the value of π_n^{-1} on the set $\mathcal{Y}$, is determined by the description of, A, h, π_{-n}, and the set $\mathcal{Z} = \left\{(y,\pi_n^{-1}(y)) \colon y \in \{0,1\}^n \setminus \mathcal{Y}\right\}$.

The set $\mathcal{Y}$ is defined via the following process.

Algorithm 34.

Set $\mathcal{Y} = \emptyset$*, and repeat until* $\mathcal{I} = \emptyset$*:*

1. *Remove the lexicographically smallest element* y *from* $\mathcal{I}$ *and insert it into* $\mathcal{Y}$*.*
2. *Let* $\{(w_1,\mathsf{C}_1,\cdot),w_1',\ldots,(w_t,\mathsf{C}_t,\cdot),w_t'\}$ *be the queries made by* $\mathsf{A}^{\pi,\mathsf{Sam}^{\pi,h}}(y)$ *to* Sam *and their answers, and let* $y_1,\ldots,y_t$ *be the outputs of the* π_n*-gates in the computations of* $\mathsf{C}_1^{\pi}(w_1),\mathsf{C}_1^{\pi}(w_1'),\ldots,\mathsf{C}_t^{\pi}(w_t),\mathsf{C}_t^{\pi}(w_t')$ *and the outputs of all* A*'s direct queries to* π_n*. Then, remove* $y_1,\ldots,y_t$ *from* $\mathcal{I}$*.*

Since at each iteration of the above process one element is inserted into the set $\mathcal{Y}$ and at most $2t$ elements are removed from the set $\mathcal{I}$, and since the $\mathcal{I}$ initially contains at least $\epsilon 2^n$ elements, when the process terminates we have that

$$a := |\mathcal{Y}| \geq \epsilon 2^n/2t \tag{5}$$

In addition, note that given the set $\mathcal{Y}$ and $\mathcal{X} = \pi_n^{-1}(\mathcal{Y})$, the set $\mathcal{Z}$ can be described using $\log((2^n - |\mathcal{Y}|)!)$ bits (by giving the order of the elements of $\{0,1\}^n \setminus \mathcal{Y}$, induced by applying π_n on $\mathcal{X}$). It follows that $\mathcal{Y}$, can be described by a string aux with

$$|\mathsf{aux}| \leq 2\log\binom{2^n}{a} + \log((2^n-a)!) \tag{6}$$

We complete the proof by presenting the algorithm $\mathsf{Decoder}$ that reconstructs π_n from the description of A, h, π_{-n}, $\mathcal{Y}$ and $\mathcal{Z}$.

Algorithm 35 ($\mathsf{Decoder}$)**.**

Input: The description of A*,* h*,* π_{-n}*,* $\mathcal{Y}$ *and* $\mathcal{Z}$*.*

Operation: For each $y \in \mathcal{Y}$ *taken in lexicographical increasing order:*

1. *Emulate* $\mathsf{A}^{\pi,\mathsf{Sam}^{\pi,h}}(y)$ *by answering* A*'s query as follows:*
   (a) *On a* π*-query* $q \in \{0,1\}^*$*:*
      - *If* $\pi(q)$ *is defined by* π_{-n}*, the set* $\mathcal{Z}$ *or the previously reconstructed values of* π_n*, answer with this value.*
      - *Else, halt the emulation and set* $\pi_n^{-1}(y) = q$*.*
   (b) *On a* Sam*-query* $q = (w, \mathsf{C}, \mathsf{C}_{\text{next}}) \in \mathcal{Q}$*:*[26]
      i. *If* $\mathsf{C} = \perp$ *answer with* $h_q(0^m)$*, where* m *is the input length of* C_{next}*.*
      ii. *Else answer with* $h_q(v)$*, where* $v \in \{0,1\}^n$ *is the minimal element such that* $\mathsf{C}^\pi(h_q(v))$ *can be evaluated (i.e., all* π_n*-queries made by* C *are defined by* π_{-n}*, the set* $\mathcal{Z}$ *or the previously reconstructed values of* π_n*) and its resulting value is* $\mathsf{C}^\pi(w)$*.*
2. *If the emulation reached its end and output* x*, set* $\pi_n^{-1}(y) = x$*.*

Assume that $\mathsf{Decoder}$ has recontracted π_n correctly for the first k elements of $\mathcal{Y}$, we proved that it also does so for the $(k+1)$ element y of $\mathcal{Y}$. To do that we show that on each query q asked by A during the emulation done by $\mathsf{Decoder}$, either $\mathsf{Decoder}$ halts on q and then $q = \pi_n^{-1}(y)$, or $\mathsf{Decoder}$ answers q correctly. (Hence, $\mathsf{Decoder}$ sets the right value for $\pi_n^{-1}(y)$).

We first handle the case that q is a π-query. It is easy to verify that $\mathsf{Decoder}$ answers correctly in case it does not halt. If halting, it must be the case that $q \in \pi_n^{-1}(\mathcal{Y})$ and $\pi_n(q) \geq_{\mathsf{lex}} y$ (otherwise, $\pi_n(q)$ would have been previously constructed). On the other hand, the definition of $\mathcal{Y}$ yields that $\pi_n(q) \leq_{\mathsf{lex}} y$ (otherwise, $(\pi_n(q), q)$ would have added to $\mathcal{Z}$), yielding that $q = \pi_n^{-1}(y)$.

In case q is a Sam-query $(w, \mathsf{C}, \mathsf{C}_{\text{next}}) \in \mathcal{Q}$, we assume without loss of generality that $\mathsf{C} \neq \perp$ (the case $\mathsf{C} = \perp$ is clear), and show that $\mathsf{Decoder}$ returns $h_q(v)$ for the lexicographically smallest v such that $\mathsf{C}^\pi(h(v)) = \mathsf{C}^\pi(w)$ (hence, it answers correctly). Let v_0 be this minimal v. It is sufficient to show that $\mathsf{Decoder}$ has enough information to evaluate $\mathsf{C}^\pi(h_q(v_0))$. Indeed, since no y-hit happens in the computation of $\mathsf{A}^{\pi,\mathsf{Sam}^{\pi,h}}(y)$, the evaluation of $\mathsf{C}^\pi(h_q(v_0))$ does *not* query π_n on $\pi_n^{-1}(y)$. Hence, the definition of $\mathcal{Y}$ guarantees that the answers to *all* queries asked by $\mathsf{C}^\pi(h_q(v_0))$ are described in $\mathcal{Z}$, or were previously reconstructed during the emulation of $\mathsf{Decoder}$. □

Now we are able to prove the following lemma, which (by holding for any *fix* choice of π_{-n} and h) is a stronger form of Lemma 30.

Lemma 36. *The following holds for all sufficiently large* n*,* $\pi_{-n} = \{\pi_i \in \Pi_i\}_{i \in \mathbb{N} \setminus \{n\}}$*,* $h \in \mathcal{H}$ *and* $2^{n/5}$*-augQueries circuit* A*:*

$$\Pr_{\pi_n \leftarrow \Pi_n}\left[\Pr_{y \leftarrow \{0,1\}^n}\left[\mathsf{A}^{\pi,\mathsf{Sam}^{\pi,h}}(y) = \pi_n^{-1}(y) \ \wedge \neg \mathsf{Hit}_{\mathsf{A},\pi,h}(y)\right] \geq 2^{-n/5}\right] \leq 2^{-2^{\frac{3n}{5}}}.$$

[26] We assume without loss of generality that A's Sam-queries are always in $\mathcal{Q}$, since it can answer other Sam-queries (i.e., not in $\mathcal{Q}$) by itself (by answering $\perp$).

Proof. Claim 33 implies that the fraction of permutations $\pi_n \in \Pi_n$ for which

$$\Pr_{y \leftarrow \{0,1\}^n}\left[\mathsf{A}^{\pi,\mathsf{Sam}^{\pi,h}}(y) = \pi_n^{-1}(y) \ \ \wedge \neg \mathsf{Hit}_{\mathsf{A},\pi,h}(y)\right] \geq 2^{-n/5} \tag{7}$$

is at most $\alpha = \frac{\binom{2^n}{a}^2 (2^n-a)!}{2^n!} = \frac{\binom{2^n}{a}}{a!}$, for $a \geq 2^{-n/5} \cdot 2^n/(2 \cdot 2^{n/5}) = 2^{\frac{3n}{5}}/2$. Using the inequalities $a! \geq (a/e)^a$ and $\binom{2^n}{a} \leq (2^n e/a)^a$, it holds that $\alpha \leq \left(\frac{2^n e^2}{a^2}\right)^a \leq \left(\frac{4e^2}{2^{n/5}}\right)^a \leq 2^{-a}$ for sufficiently large n. □

5.3 Avoiding y-Hits by Sam – Proving Lemma 31

Fix $n \in \mathbb{N}$ and a t-augQueries, d-depth normal-from circuit A such that

$$\Pr_{\substack{\pi \leftarrow \Pi, h \leftarrow \mathcal{H} \\ y \leftarrow \{0,1\}^n}}[\mathsf{Hit}_{\mathsf{A},\pi,h}(y)] \geq \varepsilon \tag{8}$$

for $\varepsilon \in [0, 1/t]$. We prove Lemma 31 by presenting a $2t$-augQueries, d-depth normal-from circuit M, with

$$\Pr_{\substack{\pi \leftarrow \Pi, h \leftarrow \mathcal{H} \\ y \leftarrow \{0,1\}^n}}\left[\mathsf{M}^{\pi,\mathsf{Sam}^{\pi,h}}(y) = \pi^{-1}(y) \wedge \neg \mathsf{Hit}_{\mathsf{M},\pi,h}(y)\right] \geq (\varepsilon/2)^{3d+1} \tag{9}$$

Let $q = (w, \mathsf{C}, \cdot)$ be a Sam-query asked in $\mathsf{A}^{\mathsf{Sam}^{\pi,h}}(y)$ that produces a y-hit (i.e., $\mathsf{C}^{\pi}(\mathsf{Sam}^{\pi,h}(q))$ queries π on $\pi^{-1}(y)$). Since A is in a normal form, *previously* to asking q it made a Sam-query $q' = (\cdot, \cdot, \mathsf{C})$, and answered by w. The main observation (see Sections 5.3.1 and 5.3.2) is that, with high probability, $\mathsf{C}^{\pi}(w)$ also queries π on $\pi^{-1}(y)$. This suggests the following circuit for inverting random permutations with no hits.

Algorithm 37 (M)**.**

Input: $y \in \{0,1\}^n$.

Oracle: $\pi \in \Pi$ *and* $\mathsf{Sam} = \mathsf{Sam}^{\pi,h}$ *for some* $h \in \mathcal{H}$.

Operation:

1. *Emulate* $\mathsf{A}^{\pi,\mathsf{Sam}^{\pi,h}}(y)$ *while adding the following check each* Sam*-query* $(\cdot, \cdot, \mathsf{C}_{\text{next}})$ *that* A *makes that answered with* w*:*

   If $\mathsf{C}^{\pi}_{\text{next}}(w)$ *queries* π *on* $x = \pi^{-1}(y)$*, return* x *and halt.*

2. *Return* $\perp$.

The rest of the proof is devoted for proving that Equation (9) holds for the above definition of M. The heart of the proof lies in the following lemma.

Lemma 38. *The following holds for every* $\pi \in \Pi$ *and* $y \in \{0,1\}^n$*. Assume that*

$$\Pr_{h \leftarrow \mathcal{H}}[\mathsf{Hit}_{\mathsf{A},\pi,h}(y)] \geq \delta \tag{10}$$

for $\delta \in [0, 1/t]$*, then*

$$\Pr_{h \leftarrow \mathcal{H}}\left[\mathsf{M}^{\pi,\mathsf{Sam}^{\pi,h}}(y) = \pi^{-1}(y) \wedge \neg \mathsf{Hit}_{\mathsf{M},\pi,h}(y)\right] \geq \delta^{3d}.$$

We prove Lemma 38 in the next section, but first let us use it for concluding the proof Lemma 31.

Proof of Lemma 31. Let $T = \{(y,\pi) \in \{0,1\}^n \times \Pi\colon \Pr_{h\leftarrow\mathcal{H}}[\mathsf{A}^{\pi,\mathsf{Sam}^{\pi,h}}(y) = \pi^{-1}(y) \wedge \mathsf{Hit}_{\mathsf{A},\pi,h}(y)] \geq \varepsilon/2\}$. The assumed success probability of A (as stated in Equation (8)) together with a simple averaging argument, yield that

$$\Pr_{y\leftarrow\{0,1\}^n,\pi\leftarrow\Pi}[(y,\pi) \in T] \geq \varepsilon/2 \tag{11}$$

Hence, by Lemma 38

$$\Pr_{h\leftarrow\mathcal{H}}\left[\mathsf{M}^{\pi,\mathsf{Sam}^{\pi,h}}(y) = \pi^{-1}(y) \wedge \neg\mathsf{Hit}_{\mathsf{M},\pi,h}(y)\right] \geq (\varepsilon/2)^{3d}$$

for every $(y,\pi) \in T$. We conclude that

$$\begin{aligned}
&\Pr_{\substack{\pi\leftarrow\Pi,h\leftarrow\mathcal{H}\\ y\leftarrow\{0,1\}^n}}\left[\mathsf{M}^{\pi,\mathsf{Sam}^{\pi,h}}(y) = \pi^{-1}(y) \wedge \neg\mathsf{Hit}_{\mathsf{M},\pi,h}(y)\right]\\
&\geq \Pr_{\pi\leftarrow\Pi,y\leftarrow\{0,1\}^n}[(y,\pi) \in T] \cdot \Pr_{\substack{\pi\leftarrow\Pi,h\leftarrow\mathcal{H}\\ y\leftarrow\{0,1\}^n}}\left[\mathsf{M}^{\pi,\mathsf{Sam}^{\pi,h}}(y) = \pi^{-1}(y) \wedge \neg\mathsf{Hit}_{\mathsf{M},\pi,h}(y) \mid (y,\pi) \in T\right]\\
&\geq \varepsilon/2 \cdot (\varepsilon/2)^{3d} = (\varepsilon/2)^{3d+1}.
\end{aligned}$$

□

5.3.1 Proving Lemma 38 — The Single-Path Case

In this section we prove Lemma 38 for a simplified case that captures the main difficulties of the proof. The extension for the general case is given in Section 5.3.2. In this simplified case A queries Sam on exactly d queries that lie *along a single path* — A queries Sam with $q_1,\ldots,q_d$ satisfying $\mathrm{p}(q_i) = q_{i-1}$ for every $2 \leq i \leq d$ (i.e., $q_i = (w_i,\mathsf{C}_i,\mathsf{C}_{\mathrm{next},i})$ implies that w_i is Sam's answer on $q_{i-1} = (\cdot,\cdot,\mathsf{C}_i)$).

In the following we fix $y \in \{0,1\}^n$, $\pi \in \Pi$ and $\delta \in [0,1/t]$ such that

$$\Pr_{h\leftarrow\mathcal{H}}[\mathsf{Hit} := \mathsf{Hit}_{\mathsf{A},\pi,h}(y)] \geq \delta. \tag{12}$$

We let $\mathrm{hit}(\mathsf{C},w)$, for circuit C and string w, be the event that $\mathsf{C}(w)$ queries π on $\pi^{-1}(y)$ (hereafter, $\mathsf{C}(x)$ stands for $\mathsf{C}^{\pi}(x)$), and use the following random variables.

Definition 39. *The following random variables are defined with respect to a random execution of* $\mathsf{A}^{\pi,\mathsf{Sam}^{\pi,H}}(y)$*, where* H *is uniformly drawn from* $\mathcal{H}$*.*[27]

- $Q_1 = (W_1 = \perp, \mathsf{C}_1 = \perp, \mathsf{C}_2), Q_2 = (W_2,\mathsf{C}_2,\mathsf{C}_3),\ldots,Q_d = (W_d,\mathsf{C}_d,\cdot)$ *denote* A*'s queries to* Sam*.*[28]
- Hit_i*, for* $i \in [d]$*, is the event* $\mathrm{hit}(\mathsf{C}_i,W_{i+1})$*, letting* $\mathrm{hit}(\perp,\cdot) = \emptyset$*, and let* $\mathsf{Hit}_{\leq i} := \bigcup_{j\in[i]}\mathsf{Hit}_j$*. (Note that* $\mathsf{Hit} = \mathsf{Hit}_{\leq d}$*.)*

[27] Since A is a circuit, and hence deterministic, these random variables are functions of H.

[28] Our simplifying assumption yields that A's queries are indeed of the above structure, and that W_{i+1} is Sam's answer on Q_i for every $i \in [d-1]$. In particular, it holds that $Q_1,\ldots,Q_i$ are determined by $W_1,\ldots,W_i$ and $\mathsf{C}_1,\ldots,\mathsf{C}_i$.

- *D_i, for $i \in [d]$, is the distribution of* Sam*'s answer on the i'th query Q_i — D_1 is the uniform distribution over $\{0,1\}^m$, and for $2 \leq i \leq d$, D_i is the uniform distribution over the set $\mathsf{C}_i^{-1}(\mathsf{C}_i(W_i))$.*
- *$\alpha_0 = \alpha_1 = 0$, and for $2 \leq i \leq d$ let $\alpha_i = \Pr_{w \leftarrow D_i}[\mathrm{hit}(\mathsf{C}_i, w)]$.*
- *α-Jump$_i$, for $i \in [d]$, is the event $\alpha_i > \max\left\{\frac{8}{\delta^2} \cdot \alpha_{i-1}, (\frac{\delta^2}{8})^{d+1}\right\}$. α-Jump$_{\leq i} := \bigcup_{j \in [i]} \alpha$-Jump$_j$ and α-Jump $:= \alpha$-Jump$_{\leq d}$.*

It is instructive to view the interaction between A and Sam as a d round game, where in the i'th round A chooses a query Q_i, and the oracle Sam samples W_{i+1} from the distribution D_i. The goal of the circuit A in this game is to cause $\mathrm{hit}(\mathsf{C}_i, W_{i+1})$ (i.e., causing the event Hit to happen).

By Equation (12), A produces a y-hit (i.e., causing the event Hit) with probability at least δ, and therefore wins the game with at least this probability. Our first observation is that the latter induces that the event α-Jump occurs with probability at least $\delta/2$. Intuitively, in case α-Jump does not occur, then the α_i's are too small in order to produce a y-hit with noticeable probability.

Claim 40. $\Pr[\alpha\text{-Jump}] > \delta/2$.

The proof of Claim 40 immediately follows from the next observation.

Claim 41. $\Pr\left[\exists i \in [d] \colon \mathsf{Hit}_{\leq i} \wedge \neg\alpha\text{-Jump}_{\leq i}\right] \leq \delta^5/512$.

Namely, we expect no hit unless a jump has previously occurred.

Proof of Claim 41. We prove that

$$\Pr[\mathsf{Hit}_i \mid \neg\alpha\text{-Jump}_{\leq i}] \leq \frac{1}{d} \cdot \frac{\delta^5}{512} \tag{13}$$

for every $2 \leq i \leq d$, and the proof of the Claim 41 follows by union bound.

Assuming $\left\{\neg\alpha\text{-Jump}_{\leq i}\right\}$, we first show that

$$\alpha_j \leq \left(\frac{\delta^2}{8}\right)^{d-j+3} \tag{14}$$

for every $2 \leq j \leq i$. For $j = 2$ compute

$$\alpha_2 \leq \max\left\{\frac{8}{\delta^2} \cdot \alpha_1, \left(\frac{\delta^2}{8}\right)^{d+1}\right\} = \max\left\{\frac{8}{\delta^2} \cdot 0, \left(\frac{\delta^2}{8}\right)^{d+1}\right\} = \left(\frac{\delta^2}{8}\right)^{d-2+3}, \tag{15}$$

where the inequality holds since we assume $\neg\alpha$-Jump$_{\leq i}$. Assuming Equation (14) holds for $2 \leq j-1 \leq i-1$, compute

$$\begin{aligned}
\alpha_j &\leq \max\left\{\frac{8}{\delta^2} \cdot \alpha_{j-1}, \left(\frac{\delta^2}{8}\right)^{d+1}\right\} \\
&\leq \max\left\{\frac{8}{\delta^2} \cdot \left(\frac{\delta^2}{8}\right)^{d-j+4}, \left(\frac{\delta^2}{8}\right)^{d+1}\right\} \\
&= \left(\frac{\delta^2}{8}\right)^{d-j+3}
\end{aligned}$$

proving Equation (14). The first inequality holds since we assume $\neg\alpha$-Jump$_{\leq i}$, and the second one by the induction hypothesis. It follows that

$$\Pr[\mathsf{Hit}_i \mid \neg\alpha\text{-Jump}_{\leq i}] \leq \left(\frac{\delta^2}{8}\right)^3 \leq \frac{1}{d} \cdot \frac{\delta^5}{512},$$

where the last inequality holds since (by the statement of the lemma) $\delta \leq 1/t \leq 1/d$. □

Given the above, the proof of Claim 40 is immediate.

Proof of Claim 40. Compute

$$\begin{aligned}\Pr[\alpha\text{-Jump}] &\geq \Pr[\mathsf{Hit}] - \Pr[\mathsf{Hit} \wedge \neg\alpha\text{-Jump}] \\ &\geq \delta - \delta^5/512 > \delta/2,\end{aligned}$$

where the second inequality follows by Claim 41. □

Consider M's point of view in the aforementioned game. Recall that following each query $Q_i = (W_i, \mathsf{C}_i, \mathsf{C}_{i+1})$, the circuit M evaluates $\mathsf{C}_{i+1}(W_{i+1})$, and if a π-gate in this computation has input $\pi^{-1}(y)$, then M outputs $\pi^{-1}(y)$ and halts. Algorithm M "wins" the game (i.e., inverts π with no hit), if it manages to retrieve $\pi^{-1}(y)$ *before* A produces any y-hits. Let β_i be the probability that M outputs $\pi^{-1}(y)$ and halts after query q_i.

Definition 42. *For $i \in [d]$ let $\beta_i = \Pr_{w \leftarrow D_i}[\mathrm{hit}(\mathsf{C}_{i+1}, w)]$.*

The game between A and M 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+1}, which determines α_{i+1}. If q_i implies "high" β_i, then M has high probability in winning the game (i.e., with high probability the computation $\mathsf{C}_{i+1}(w_{i+1})$ done by M finds $\pi^{-1}(y)$. Therefore to win the game, A should not choose high β_i. We claim, however, that if β_i is low, then with high probability α_{i+1} will be low as well. But 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 produce high β_i.

The following claim states that conditioned on $Q_1, \ldots, Q_i$, the expectation of α_{i+1} is β_i. Therefore, if β_i is low then α_{i+1} is low with high probability. Note that under the above conditioning (which determines the value of $W_1 \cdots, W_i$), the value of β_i is determined, while α_{i+1} is still a random variable, to be determined by the value of $W_{i+1} = \mathsf{Sam}(Q_i)$.

Claim 43. $\mathrm{E}[\alpha_{i+1} \mid W_1, \ldots, W_i] = \beta_i$ *for every* $i \in [d-1]$.

Proof. Fix $i \in [d-1]$ and a fixing $w_1, \ldots, w_i$ for $W_1, \ldots, W_i$ (which implies a fixing $q_1 = (w_1, \mathsf{C}_1, \mathsf{C}_2), \ldots, q_i = (w_i, \mathsf{C}_i, \mathsf{C}_{i+1})$ for $Q_1, \ldots, Q_i$). We write

$$\begin{aligned}\mathrm{E}[\alpha_{i+1}] &= \sum_{z \in \{0,1\}^\ell} \Pr_{w \leftarrow D_i}[\mathsf{C}_{i+1}(w) = z] \cdot \Pr_{w \leftarrow \mathsf{C}_{i+1}^{-1}(z)}[\mathrm{hit}(\mathsf{C}_{i+1}, w)] \qquad (16) \\ &= \sum_z \frac{\left|\mathsf{C}_{i+1}^{-1}(z)\right|}{|\mathcal{S}|} \cdot \frac{\left|\left\{w \in \mathsf{C}_{i+1}^{-1}(z) \colon \mathrm{hit}(\mathsf{C}_{i+1}, w)\right\}\right|}{\left|\mathsf{C}_{i+1}^{-1}(z)\right|} \\ &= \sum_z \frac{\left|\left\{w \in \mathsf{C}_{i+1}^{-1}(z) \colon \mathrm{hit}(\mathsf{C}_{i+1}, w)\right\}\right|}{|\mathcal{S}|},\end{aligned}$$

where $\mathcal{S} := \mathsf{C}_i^{-1}(\mathsf{C}_i(w_i))$ and ℓ is the output length of C_{i+1}. Note that while Q_{i+1} is not determined by $q_1, \ldots, q_i$, the circuit C_{i+1} is. In addition, since A is in a normal form,[29] the circuit C_{i+1} is an extension of C_i. Thus

$$\begin{aligned}
\mathrm{E}[\alpha_{i+1}] &= \sum_{z \in \{0,1\}^\ell} \frac{\left|\left\{w \in \mathsf{C}_{i+1}^{-1}(z) \colon \mathrm{hit}(\mathsf{C}_{i+1}, w)\right\}\right|}{|\mathcal{S}|} \\
&= \frac{\left|\left\{w \in \mathcal{S} \colon \mathrm{hit}(\mathsf{C}_{i+1}, w)\right\}\right|}{|\mathcal{S}|} \\
&= \Pr_{w \leftarrow \mathcal{S}}\left[\mathrm{hit}(\mathsf{C}_{i+1}, w)\right] \\
&= \Pr_{w \leftarrow D_i}\left[\mathrm{hit}(\mathsf{C}_{i+1}, w)\right] \\
&= \beta_i.
\end{aligned}$$

□

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, implying that α-Jump occurs. We have also seen that the latter requires A to choose a query q_i that determines a high β_i. We claim that in this case, it holds that β_i is *significantly larger* than α_i. Formally,

Definition 44. *Let* $\alpha\beta$-Gap_i*, for* $i \in [d]$*, be the event* $\left\{\beta_i > \max\left\{2\alpha_i, \left(\frac{\delta^2}{8}\right)^{d+2}\right\}\right\}$.

The following claim states that with a noticeable probability, there exists an index i such that $\alpha\beta$-Gap_i occurs and α-$\mathrm{Jump}_{\leq i}$ does not occur. In other words, β_i is significantly larger than α_j for all $j \leq i$. We later show that this β_i enables M to retrieve $\pi^{-1}(y)$ before A produces any y-hits.

Claim 45. *Let* GapFirst *be the event* $\left\{\exists i \in [d] \colon \alpha\beta\text{-}\mathrm{Gap}_i \wedge \neg\alpha\text{-}\mathrm{Jump}_{\leq i}\right\}$*, then* $\Pr\left[\mathrm{GapFirst}\right] \geq \delta/4$.

For proving Claim 45 we use the following claim, showing that unless β_i is significantly larger than α_i, then α_{i+1} is not significantly larger than α_i.

Claim 46. $\Pr[\alpha\text{-}\mathrm{Jump}_{i+1} \mid \neg\alpha\beta\text{-}\mathrm{Gap}_i] \leq \delta^2/4$ *for every* $i \in [d-1]$.

Proof. We write

$$\Pr\left[\alpha\text{-}\mathrm{Jump}_{i+1} \mid \neg\alpha\beta\text{-}\mathrm{Gap}_i\right] \leq \Pr\left[\alpha_{i+1} > \beta_i \cdot \delta^2/4\right] + \Pr\left[\alpha\text{-}\mathrm{Jump}_{i+1} \mid \neg\alpha\beta\text{-}\mathrm{Gap}_i \wedge \left\{\alpha_{i+1} \leq \beta_i \cdot \delta^2/4\right\}\right] \tag{17}$$

Claim 43 and Markov's inequality imply that

$$\Pr\left[\alpha_{i+1} > \frac{4}{\delta^2} \cdot \beta_i\right] \leq \delta^2/4 \tag{18}$$

Since the event $\left\{\alpha\text{-}\mathrm{Jump}_{i+1} \cap \neg\alpha\beta\text{-}\mathrm{Gap}_i \cap \left\{\alpha_{i+1} \leq \frac{4}{\delta^2} \cdot \beta_i\right\}\right\}$ is empty, we conclude that

$$\Pr\left[\alpha\text{-}\mathrm{Jump}_{i+1} \mid \neg\alpha\beta\text{-}\mathrm{Gap}_i\right] \leq \delta^2/4.$$

□

[29] This is the only place throughout the whole proof, where the normal-from assumption that A is in a normal form is being used.

Given Claim 46, we prove Claim 45 as follows.

Proof of Claim 45. Compute

$$\begin{aligned}\Pr\left[\alpha\text{-Jump} \wedge \neg\text{GapFirst}\right] &\leq \sum_{i\in[d]} \Pr\left[\alpha\text{-Jump}_i \wedge \neg\alpha\text{-Jump}_{\leq i-1} \wedge \neg\alpha\beta\text{-Gap}_i\right] \\ &\leq \sum_{i\in[d]} \Pr\left[\alpha\text{-Jump}_i \mid \neg\alpha\beta\text{-Gap}_i\right] \\ &\leq d\cdot\delta^2/4 \\ &\leq \delta/4,\end{aligned} \tag{19}$$

where the before to last inequality holds by Claim 46, and last inequality holds since $\delta \leq 1/t \leq 1/d$.

Since (by Claim 40) $\Pr\left[\alpha\text{-Jump}\right] \geq \delta/2$, it follows that $\Pr\left[\text{GapFirst}\right] \geq \delta/4$. □

Putting it together Given the above observation, we are ready to prove Lemma 38.

Proof of Lemma 38 (Single-path case). Let I be the smallest index in $[d-1]$ for which $\alpha\beta\text{-Gap}_i$ occurs, letting $I = \perp$ in case no such event happens. Note that whenever GapFirst happens, then $I \neq \perp$ and $\alpha\text{-Jump}_i$ does not occur for all $i \in [I]$. Compute

$$\begin{aligned}\Pr[\mathsf{Hit}_{\leq I-1} \mid \text{GapFirst}] &= \Pr[\mathsf{Hit}_{\leq I-1} \wedge \neg\alpha\text{-Jump}_{\leq I-1} \mid \text{GapFirst}] \\ &\leq \Pr\left[\exists i \in [d]\colon \mathsf{Hit}_{\leq i} \wedge \neg\alpha\text{-Jump}_{\leq i}\right]\cdot\frac{1}{\Pr[\text{GapFirst}]} \\ &\leq \frac{\delta^5}{512}\cdot\frac{4}{\delta} < 1/2,\end{aligned} \tag{20}$$

where the second inequality holds by Claims 41 and 45. In addition, for the event $E = \{\text{hit}(\mathsf{C}_{I+1}, W_{I+1}) \cap \neg\text{hit}(\mathsf{C}_I, W_{I+1})\}$ it holds that

$$\begin{aligned}&\Pr[E \mid \text{GapFirst}, \neg\mathsf{Hit}_{\leq I-1}] \\ &\geq \Pr\left[\text{hit}(\mathsf{C}_{I+1}, W_{I+1}) \mid \text{GapFirst}, \neg\mathsf{Hit}_{\leq I-1}\right] - \Pr\left[\text{hit}(\mathsf{C}_I, W_{I+1}) \mid \text{GapFirst}, \neg\mathsf{Hit}_{\leq I-1}\right] \\ &= \mathrm{E}\left[\beta_I \mid \text{GapFirst}, \neg\mathsf{Hit}_{\leq I-1}\right] - \mathrm{E}\left[\alpha_I \mid \text{GapFirst}, \neg\mathsf{Hit}_{\leq I-1}\right] \\ &= \mathrm{E}\left[\beta_I - \alpha_I \mid \text{GapFirst}, \neg\mathsf{Hit}_{\leq I-1}\right] \\ &\geq \frac{1}{2}\cdot\left(\frac{\delta^2}{8}\right)^{d+2},\end{aligned} \tag{21}$$

where the second inequality holds by the definition of $\alpha\beta$-Gap. We conclude that

$$\begin{aligned}
&\Pr\left[\mathsf{M}^{\pi,\mathsf{Sam}^{\pi,H}}(y) = \pi^{-1}(y) \wedge \neg\mathsf{Hit}\right] \\
&= \Pr\left[\exists i\colon \mathrm{hit}(\mathsf{C}_{i+1}, W_{i+1}) \wedge \neg\mathsf{Hit}_{\leq i}\right] \\
&\geq \Pr\left[\mathrm{GapFirst}\right] \cdot \Pr\left[\mathrm{hit}(\mathsf{C}_{I+1}, W_{I+1}) \wedge \neg\mathsf{Hit}_{\leq I} \mid \mathrm{GapFirst}\right] \\
&= \Pr\left[\mathrm{GapFirst}\right] \cdot \Pr\left[E \wedge \neg\mathsf{Hit}_{\leq I-1} \mid \mathrm{GapFirst}\right] \\
&\geq \Pr\left[\mathrm{GapFirst}\right] \cdot \Pr[\neg\mathsf{Hit}_{\leq I-1} \mid \mathrm{GapFirst}] \cdot \Pr[E \mid \mathrm{GapFirst}, \neg\mathsf{Hit}_{\leq I-1}] \\
&\geq \frac{\delta}{4} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \left(\frac{\delta^2}{8}\right)^{d+2} \\
&= \left(\frac{\delta^2}{8}\right)^{d+3} \\
&\geq \delta^{3d}.
\end{aligned}$$

□

5.3.2 Proving Lemma 38 — The General Case

This section extends the proof given in Lemma 38 to the general case where A's queries are not necessarily along a single path. The extension below is mainly technical, and requires no more than refining some events and notations.

Assuming that $Q_1, \ldots, Q_t$ are the Sam-queries asked by A, let $\mathrm{p}(i)$, for $i \in [s]$, be the index of the query $\mathrm{p}(Q_i)$ (i.e., the index of Q_i's parent, see Definition 15) in the above query list, letting $\mathrm{p}(i) = 0$ in case $\mathrm{p}(Q_j) = \perp$. Note that unlike the single-path case studies in Section 5.3.1, the values of $\mathrm{p}(1), \ldots, \mathrm{p}(t)$ are not predetermined (in particular $\mathrm{p}(i)$ might not be $i-1$). This difference reflects the fact that A might *repetitively* ask the same query, each time dictating Sam to use *fresh randomness* by slightly modifying the value of the parameter $\mathsf{C}_{\mathrm{next}}$, until the answer serves it best: until there is a big jump in the value of α. It turns out that while repeating a query does increase the probability of A to "win" the game against M (i.e., to make a y-hit before M inverts y), since the expected value α_i is the value of $\alpha_{\mathrm{p}(i)}$, such repetition does not increase the A's winning probability by too much.

We now describe in detail the required technical changes to the proof of Lemma 38. Fix $y \in \{0,1\}^n$, $\pi \in \Pi$ and $\delta \in [0, 1/t]$ such that $\Pr_{h \leftarrow \mathcal{H}}[\mathsf{Hit} := \mathsf{Hit}_{\mathsf{A},\pi,h}(y)] \geq \delta$. The definitions of the following random variables are natural generalization of those given in Section 5.3.1. Recall that $\mathrm{hit}(\mathsf{C}, w)$ is the event that $\mathsf{C}(w)$ queries π on $\pi^{-1}(y)$. The index i in the following definitions takes values in $[t]$.

Definition 47. *The following random variables are defined with respect to a random execution of* $\mathsf{A}^{\pi,\mathsf{Sam}^{\pi,H}}(y)$*, where* H *is uniformly drawn from* $\mathcal{H}$*.*

- $Q_1 = (W_1, \mathsf{C}_1, \mathsf{C}_{\mathrm{next},1}, \ldots, Q_s = (W_t, \mathsf{C}_t, \mathsf{C}_{\mathrm{next},t})$*, denote* A*'s queries to* Sam*, and* $W_1^{\mathrm{Ans}}, \ldots, W_t^{\mathrm{Ans}}$ *denote their answers.*[30]
- Hit_i *is the event* $\mathrm{hit}(\mathsf{C}_i, W_i^{\mathrm{Ans}})$*, letting* $\mathrm{hit}(\perp, \cdot) = \emptyset$*, and* $\mathsf{Hit}_{\leq i} := \bigcup_{j \in [i]} \mathsf{Hit}_j$*.*

[30] Since A is in a normal form, for every $i \in [t]$ it holds that $W_i = W_{\mathrm{p}(i)}^{\mathrm{Ans}}$ and $\mathsf{C}_i = \mathsf{C}_{\mathrm{next},\mathrm{p}(i)}$, letting $W_0 = \mathsf{C}_{\mathrm{next},0} = \perp$. Note that $Q_1, \ldots, Q_i$ are determined by $W_1^{\mathrm{Ans}}, \ldots, W_{i-1}^{\mathrm{Ans}}$ and the circuits $\mathsf{C}_1, \mathsf{C}_{\mathrm{next},1}, \ldots, \mathsf{C}_i, \mathsf{C}_{\mathrm{next},i}$.

- D_i *is the uniform distribution over* $\mathsf{C}_i^{-1}(\mathsf{C}_i(W_i))$ *in case* $W_i \neq \perp$*, and the uniform distribution over* $\{0,1\}^m$ *otherwise.*
- $\alpha_i = \Pr_{w \leftarrow D_i}[\mathrm{hit}(\mathsf{C}_i, w)]$ *in case* $W_i \neq \perp$*, and* $\alpha_i = 0$ *otherwise.*
- $\beta_i = \Pr_{w \leftarrow D_i}[\mathrm{hit}(\mathsf{C}_{\mathrm{next},i}, w)]$.
- $\alpha\beta\text{-Gap}_i$ *is the event* $\beta_i > \max\left\{2\alpha_i, \left(\frac{\delta^2}{8}\right)^{d+2}\right\}$

In addition, we make use of the following definition.

Definition 48.

- D_i^{Dec} *is the uniform distribution over* $\mathsf{C}_{\mathrm{next},i}^{-1}(\mathsf{C}_{\mathrm{next},i}(W_i^{\mathrm{Ans}}))$*, and* $\alpha_i^{\mathrm{Dec}} = \Pr_{w \leftarrow D_i^{\mathrm{Ans}}}[\mathrm{hit}(\mathsf{C}_{\mathrm{next},i}, w)]$.[31]
- $\alpha\text{-PJump}_i$ *(for "Potential* α-Jump*") is the event* $\alpha_i^{\mathrm{Ans}} > \max\left\{\frac{8}{\delta^2} \cdot \alpha_i\right\}$.

The following claims are analogues to the claims given in Section 5.3.1.

Claim 49. $\Pr[\alpha\text{-PJump}] > \delta/2$.

Proof. Same as the proof of Claim 40, replacing Claim 41 with Claim 50. □

Claim 50. $\Pr\left[\exists i \in [t] \colon \mathsf{Hit}_{\leq i} \wedge \neg\alpha\text{-PJump}_{\leq i-1}\right] \leq \delta^5/512$.

Proof. Same as the proof of Claim 40, replacing Equation (14) with

$$\alpha_j \leq \left(\frac{\delta^2}{8}\right)^{d-\mathrm{depth}(j)+3}, \tag{22}$$

where $\mathrm{depth}(j) = 0$ for $j = 0$, and $\mathrm{depth}(\mathrm{p}(j)) + 1$ otherwise. □

Claim 51. $\mathrm{E}[\alpha_i^{\mathrm{Ans}} \mid W_1^{\mathrm{Ans}}, \ldots, W_{i-1}^{\mathrm{Ans}}] = \beta_i$ *for every* $i \in [t]$.

Proof. Same as the proof of Claim 43, replacing C_{i+1} with $\mathsf{C}_{\mathrm{next},i}$. □

Claim 52. *Let* GapFirst *the event* $\left\{\exists i \in [t] \colon \alpha\beta\text{-Gap}_i \wedge \neg\alpha\text{-PJump}_{\leq i-1}\right\}$*, then* $\Pr[\mathrm{GapFirst}] \geq \delta/4$.

Proof. Same as the proof of Claim 45, replacing Claim 46 with Claim 53, and recalling that $\delta < 1/t$. □

Claim 53. $\Pr[\alpha\text{-PJump}_i \mid \neg\alpha\beta\text{-Gap}_i] \leq \delta^2/4$ *for every* $i \in [t]$.

Proof. Same as the proof of Claim 46, replacing α_{i+1} with α_i^{Ans}, and $\alpha\text{-Jump}_{i+1}$ with $\alpha\text{-PJump}_i$. □

As in Section 5.3.1, the proof of Lemma 38 (here for the general case) easily follow the above claims.

Proof of Lemma 38 (General case). Same lines as the proof given in Section 5.3.1, replacing Claims 41 and 45 with Claims 50 and 52. □

[31] Note that for any j with $p(j) = i$, if such exists, it holds that $D_i^{\mathrm{Dec}} = D_j$ and $\alpha_i^{\mathrm{Dec}} = \alpha_j$.

6 Lower Bounds on Statistically Hiding commitments

In this section we combine the results presented in Sections 4 and 5 to derive our lower bounds on black-box constructions of statistically-hiding commitments from trapdoor permutations. Throughout the section, we assume for ease of notation that the integer functions d, c and s, measuring the round and sender communication complexity of the considered commitment scheme, and the hardness of the considered trapdoor permutations family, respectively, are non-decreasing.

6.1 The Round Complexity Lower Bound

In this section we give lower bound on the *round complexity* of black-box constructions of statistically hiding commitment from trapdoor permutations. We first give two results for the case where the reduction is to polynomially hard families . The first result is for "security-preserving" constructions, and the second one is for arbitrary ones.

Theorem 54 (restating Theorem 1)**.** *Any $O(n)$-security-parameter-expanding, fully black-box construction of a weakly binding and honest-receiver statistically hiding commitment scheme from a polynomially hard family of trapdoor permutations has $\Omega\left(n/\log n\right)$ communication rounds.*

Theorem 55. *Any fully black-box construction of a weakly binding and honest-receiver statistically hiding commitment scheme from a polynomially hard family of trapdoor permutations has $n^{\Omega(1)}$ communication rounds.*

The above two theorems are in fact corollaries of the more general statement given below, stated for trapdoor permutations of arbitrary hardness.

Theorem 56. *For every ℓ-security-parameter-expanding, fully black-box construction of a d-round weakly binding and honest-receiver statistically hiding commitment scheme from an $s \geq n^{\omega(1)}$-hard family of trapdoor permutations, it holds that $d(\ell(n)) \in \Omega\left(n/\log s(n)\right)$.*

Proof. Let $(\mathsf{Com} = (\mathsf{S}, \mathsf{R}, \mathsf{V}))$ be an ℓ-security-parameter-expanding fully black-box construction of a d-round, δ-binding and honest-receiver, honest-sender, statistically hiding commitment scheme from an s-hard family of trapdoor permutations, where $\delta(n) = 1 - 1/p(n)$ for some $p \in \text{poly}$, and let $m = m(n)$ be a bound on the running time of R on security parameter n. Theorem 21 yields that relative to most fixing of $(\tau, \mathsf{Sam}^{\tau,h})$, there exists an efficient breaker for the binding of Com.

Claim 57. *There exists a $(d+1)$-depth, deterministic* poly*-augQueries, normal-from oracle-aided algorithm $\widetilde{\mathsf{S}}$ such that the following holds for every $\tau \in \mathrm{T}$: for $h \in \mathcal{H}$ and $r_{\mathsf{R}} \in \{0,1\}^m$ let $\mathsf{TwoOpenings}_n^h(r_{\mathsf{R}})$ be the event that $\mathsf{V}(\mathsf{com}, \mathsf{decom}) \neq \mathsf{V}(\mathsf{com}, \mathsf{decom}') \in \{0,1\}$ for $((\mathsf{decom}, \mathsf{decom}'), \mathsf{com}) = \left\langle (\widetilde{\mathsf{S}}^{\tau, \mathsf{Sam}^{\tau,h}}, \mathsf{R}^{\tau}(r_{\mathsf{R}}))(1^n) \right\rangle_{\mathrm{out}^{\widetilde{\mathsf{S}}}, \mathrm{out}^{\mathsf{R}}}$, and let $\mathsf{NoBreak}_n^{\tau,h}$ be the event that $\Pr_{r_{\mathsf{R}} \leftarrow \{0,1\}^m}\left[\neg \mathsf{TwoOpenings}_n^h(r_{\mathsf{R}})\right] \geq \delta(n)$. Then $\Pr_{h \leftarrow \mathcal{H}}\left[\mathsf{NoBreak}_n^h\right] \in O(1/n^2)$.*

We defer the proof of Claim 57 of to Section 6.1.1, and first use it for proving Theorem 56. Claim 57 yields that

$$\sum_{n=1}^{\infty} \Pr_{h \leftarrow \mathcal{H}}\left[\mathsf{NoBreak}_n^{\tau,h}\right] < \infty \tag{23}$$

for any $\tau \in \mathrm{T}$, where $\widetilde{\mathsf{S}}$ and $\mathsf{NoBreak}_n^h$ are as in the claim statement. By the Borel-Cantelli lemma, the probability over the choices of $h \leftarrow \mathcal{H}$ that $\mathsf{NoBreak}_n^h$ occurs for infinitely many n's, is *zero*. It follows that with probability one over the choice of $(\tau, h) \leftarrow \mathrm{T} \times \mathcal{H}$, it holds that $\widetilde{\mathsf{S}}^{\tau,\mathsf{Sam}^{\tau,h}}$ breaks the weak binding of Com. Hence, with probability one over the choice of $(\tau = (G, F, F^{-1}), h)$, it holds that

$$\Pr_{td \leftarrow \{0,1\}^n, y \leftarrow \{0,1\}^n}\left[\mathsf{A}^{\tau,\mathsf{S}^{\tau,\mathsf{Sam}^{\tau,h}},h}(1^n, G(td), y) = F^{-1}(td, y)\right] > \frac{1}{s(n)} \tag{24}$$

for infinitely many n's. Since Equation (24) holds with respect to measure one of the oracles (τ, h), we have that

$$\Pr_{\substack{\tau \leftarrow \mathrm{T}, h \leftarrow \mathcal{H} \\ td \leftarrow \{0,1\}^n, y \leftarrow \{0,1\}^n}}\left[\mathsf{A}^{\tau,\widetilde{\mathsf{S}}^{\tau,\mathsf{Sam}^{\tau,h}},h}(1^n, G(td), y) = F^{-1}(td, y)\right] > \frac{1}{s(n)} \tag{25}$$

for infinitely many n's.

By Proposition 18, the circuit $\mathsf{A}^{\widetilde{\mathsf{S}}}$ (i.e., the circuit that given oracle access to τ and $\mathsf{Sam}^{\tau,h}$, acts as $\mathsf{A}^{\tau,\widetilde{\mathsf{S}}^{\tau,\mathsf{Sam}^{\tau,h}},h}$) is in a normal form and of depth $d'(n) = d(\ell(n))+1$. Hence, Equation (25) yields the existence of an $t = 4s$-augQueries, normal form, d'-depth, oracle-aided circuit family $\widetilde{\mathsf{A}} = \left\{\widetilde{\mathsf{A}}_n\right\}_{n\in\mathbb{N}}$ with

$$\Pr_{\substack{\tau \leftarrow \mathrm{T}, h \leftarrow \mathcal{H} \\ td \leftarrow \{0,1\}^n, y \leftarrow \{0,1\}^n}}\left[\widetilde{\mathsf{A}}_n^{\tau,\mathsf{Sam}^{\tau,h},h}(G(td), y) = F^{-1}(td, y)\right] > \frac{4}{t(n)} \tag{26}$$

for infinitely many n's.

Theorem 32 yields that $2^{n/8} \leq (4t(n))^{3d(\ell(n))+1} \leq (4s(n))^{6d(\ell(n))+2}$, implying that $d(\ell(n)) \in \Omega(n/\log s(n))$. □

6.1.1 Proving Claim 57

Proof Claim 57. Let $\widetilde{\mathsf{A}}$ be the deterministic, polynomial-augQueries algorithm guaranteed by Theorem 21 for the protocol (S, R). Recall the following holds for every $h \in \mathcal{H}$ and $k \in \mathbb{N}$: following the execution of $(\widetilde{\mathsf{A}}^{\mathsf{Sam}^{\tau,h},h}(1^k), \mathsf{R}^\tau)(1^n))$ that yields a transcript trans, algorithm A outputs a set $\{(b_i, r_i))\}_{i\in[k]}$ such that the k pairs are independent uniform values for the input and random coins of S, consistent with trans. Also recall that over a uniform choice of $h \leftarrow \mathcal{H}$, the value of trans has the same distribution has the one induced by $\langle \mathsf{S}^\tau, \mathsf{R}^\tau)(1^n)\rangle$.

Algorithm S with oracle access to τ and $\mathsf{Sam}^{\tau,h}$, acts through the interaction with R as $\mathsf{A}^{\tau,\mathsf{Sam}^{\tau,h},h}$ would on input $(1^n, 1^n)$ (i.e., we set $k = n$). If in the set output by A there exist two pairs $(0, r_0)$ and $(1, r_1)$, $\widetilde{\mathsf{S}}$ uses them to generate two decommitments decom_0 and decom_1. Note that, if such pairs were found, then it holds that $\mathsf{V}(\mathsf{com}, \mathsf{decom}_0) = 0$ and $\mathsf{V}(\mathsf{com}, \mathsf{decom}_1) = 1$, where com is the commitment output by R when interacting with $\widetilde{\mathsf{S}}$. In the following we prove that $\widetilde{\mathsf{S}}$ finds such a good couple of pairs with save but negligible probability over the choice of $h \in \mathcal{H}$.

We next define a set of "good" transcripts that enable $\widetilde{\mathsf{S}}$ to reveal to both 0 and 1 with overwhelming probability. For $n \in \mathbb{N}$ and $b \in \{0,1\}$, let $\mathrm{Trans}_n^b = \langle \mathsf{S}^\tau(b), \mathsf{R}^\tau)(1^n)\rangle_{\mathsf{trans}}$ (i.e., the random variable induced by the transcript of a random execution of $(\mathsf{S}^\tau, \mathsf{R}^\tau)$, where S's input bit is b), let $\mathrm{Trans} = \mathrm{Trans}_n^u$, for $u \leftarrow \{0,1\}$, and let $\mathsf{Balanced}_n =$

$\left\{\mathsf{trans} \in \mathrm{Supp}(\mathrm{Trans}_n)\colon \frac{1}{2} \leq \frac{\Pr_{\mathrm{Trans}_n^0}[\mathsf{trans}]}{\Pr_{\mathrm{Trans}_n^1}[\mathsf{trans}]} \leq \frac{3}{2}\right\}$. Since Com is statistically hiding (at least, against the honest receiver), it follows that

$$\Pr_{\mathrm{Trans}_n}[\neg\mathsf{Balanced}_n] = \mathrm{neg}(n) \tag{27}$$

For $h \in \mathcal{H}$ and $r \in \{0,1\}^m$, let $\mathsf{trans}_n^{h,r_\mathsf{R}} = \left\langle \widetilde{\mathsf{S}}^{\tau,\mathsf{Sam}^{\tau,h}}, \mathsf{R}^\tau(r_\mathsf{R}))(1^n)\right\rangle_{\mathsf{trans}}$. Theorem 21 yields that $\mathsf{trans}_n^{h,r_\mathsf{R}}$, for uniformly chosen values of h and r_R, and Trans_n, are identically distributed, and that

$$\mathop{\mathrm{E}}_{h\leftarrow\mathcal{H},r_\mathsf{R}\leftarrow\{0,1\}^m}\left[\neg\mathsf{TwoOpenings}_n^h(r_\mathsf{R}) \mid \mathsf{trans}_n^{h,r_\mathsf{R}} \in \mathsf{Balanced}\right] \leq \left(\frac{2}{3}\right)^{n-1} \tag{28}$$

We conclude that

$$\begin{aligned}
&\Pr_{h\leftarrow\mathcal{H}}\left[\mathsf{NoBreak}_n^h\right] \\
&= \Pr_{h\leftarrow\mathcal{H}}\left[\Pr_{r_\mathsf{R}\leftarrow\{0,1\}^m}\left[\neg\mathsf{TwoOpenings}_n^h(r_\mathsf{R})\right] \geq \delta(n)\right] \\
&\leq \frac{\mathrm{E}_{h\leftarrow\mathcal{H},r_\mathsf{R}\leftarrow\{0,1\}^m}\left[\neg\mathsf{TwoOpenings}_n^h(r_\mathsf{R})\right]}{\delta(n)} \\
&\leq \frac{1}{\delta(n)}\cdot\left(\Pr_{\mathrm{Trans}_n}[\neg\mathsf{Balanced}_n] + \mathop{\mathrm{E}}_{h\leftarrow\mathcal{H},r_\mathsf{R}\leftarrow\{0,1\}^m}\left[\neg\mathsf{TwoOpenings}_n^h(r_\mathsf{R}) \mid \mathsf{trans}_n^{h,r_\mathsf{R}} \in \mathsf{Balanced}\right]\right) \\
&\leq \mathrm{neg}(n),
\end{aligned}$$

where the last inequality follows from Equations (27) and (28). □

6.2 The Communication Complexity Lower Bound

In this section we give lower bound on the *sender communication complexity* of black-box constructions of statistically hiding commitment from trapdoor permutations. We first give two results for the case where the reduction is to polynomially hard families. The first result is for "security-preserving" construction, and the second one is for arbitrary one.

Theorem 58 (restating Theorem 2)**.** *In every $O(n)$-security-parameter-expanding, fully black-box construction of a weakly binding, honest-receiver, honest-sender statistically hiding commitment scheme from a polynomially hard family of trapdoor permutations, the sender sends $\Omega(n)$ bit.*

Theorem 59. *In every fully black-box construction of a weakly binding and honest-receiver, honest-receiver statistically hiding commitment scheme from a polynomially hard family of trapdoor permutations, the sender sends $n^{\Omega(1)}$ bits.*

The above two theorems are in fact corollaries of the more general statement given below, for trapdoor permutations of arbitrary hardness.

Theorem 60. *In every ℓ-security-parameter-expanding fully black-box construction of a d-round weakly binding and honest-receiver, and honest-receiver statistically hiding commitment scheme from an $s(n) \geq n^{\omega(1)}$-hard family of trapdoor permutations in which the sender communicates $c(\cdot)$ bits, it holds that $c(\ell(n)) \in \Omega(n)$.*

Proof. The proof follows in large parts the proof of Theorem 56 given above, so we only mention the significant differences.

Let $(\mathsf{Com} = (\mathsf{S}, \mathsf{R}, \mathsf{V}))$ be an ℓ-security-parameter-expanding fully black-box construction of a c-communication-complexity, δ-binding and honest-receiver, honest-sender, statistically hiding commitment scheme from an s-hard family of trapdoor permutations, where $\delta(n) = 1 - 1/p(n)$ for some $p \in \text{poly}$, and let $m = m(n)$ be a bound on the running time of S and R on security parameter n. Theorem 26 yields that the following holds.

Claim 61. *There exists a* $d = \left(\left\lceil \frac{4c}{\log s} \right\rceil + 1\right)$*-depth, deterministic* $O(\sqrt[3]{s})$*-augQueries, normal-from oracle-aided algorithm* $\widetilde{\mathsf{S}}$ *such that the following holds for every* $\tau \in \mathrm{T}$*: for* $h \in \mathcal{H}$ *and* $r_{\mathsf{S}}, r_{\mathsf{R}} \in \{0,1\}^m$*, let* $\mathsf{TwoOpenings}_n^h(r_{\mathsf{S}}, r_{\mathsf{R}})$ *be one if and only if* $\mathsf{V}(\mathsf{com}, \mathsf{decom}) \neq \mathsf{V}(\mathsf{com}, \mathsf{decom}') \in \{0,1\}$ *for* $\mathsf{com} = \langle (\mathsf{S}^\tau(0, r_{\mathsf{S}}), \mathsf{R}(r_{\mathsf{R}})^\tau)(1^n) \rangle_{\mathsf{out}_{\mathsf{R}}}$ *and* $(\mathsf{decom}, \mathsf{decom}') = \widetilde{\mathsf{S}}^{\tau, \mathsf{Sam}^{\tau,h}}(\mathsf{com})$*, and let* $\mathsf{NoBreak}_n^{\tau,h}$ *be the event that* $\Pr_{r_{\mathsf{S}} \leftarrow \{0,1\}^m, r_{\mathsf{R}} \leftarrow \{0,1\}^m} \left[\neg \mathsf{TwoOpenings}_n^h(r_{\mathsf{S}}, r_{\mathsf{R}})\right] \geq \delta(n)$*. Then* $\Pr_{h \leftarrow \mathcal{H}} \left[\mathsf{NoBreak}_n^h\right] \in O(1/n^2)$*.*

The proof of Claim 61 follows similar lines to that of Claim 57, see more details in Section 6.2.1. Similarly to the proof of Theorem 56, Claim 61 yields that there exists an $t = 4s$-augQueries, normal form, $d+1$-depth, oracle-aided circuit family $\widetilde{\mathsf{A}} = \left\{\widetilde{\mathsf{A}}_n\right\}_{n \in \mathbb{N}}$ with

$$\Pr_{\substack{\tau \leftarrow \mathrm{T}, h \leftarrow \mathcal{H} \\ td \leftarrow \{0,1\}^n, y \leftarrow \{0,1\}^n}} \left[\widetilde{\mathsf{A}}_n^{\tau, \mathsf{Sam}^{\tau,h}}(G(td), y) = F^{-1}(td, y)\right] > \frac{4}{t(n)} \tag{29}$$

for infinitely many n's. By Theorem 32, it follows that $d(\ell(n)) \in \Omega(n/\log s(n))$. Since, by our simplifying assumption, s is non-decreasing, it follows that $c(\ell(n)) \in \Omega\left(\frac{n \cdot \log(s(\ell(n)))}{\log s(n)}\right) \in \Omega(n)$. □

6.2.1 Proving Claim 61

Proof Claim 61. The proof follows in large parts the proof of Claim 57 given above, so we only mention the significant differences.

Let Inv be the algorithm guaranteed Theorem 26 for the protocol (S, R). Following an execution $\langle (\mathsf{S}^\tau(0), \mathsf{R}^\tau)(1^n) \rangle$ resulting in transcript trans, algorithm $\mathsf{S}^{\tau, \mathsf{Sam}^{\tau,h}}$ calls $\mathsf{Inv}^{\tau, \mathsf{Sam}^{\tau,h}}(1^n, 1^n, d(n) - 1, \varepsilon(n) = \delta(n)/n^2, \mathsf{trans})$ to get set of pairs $\{(b_i, r_i))\}_{i \in [n]}$. If there exists two pairs $(0, r_0)$ and $(1, r_1)$ in the above set, $\widetilde{\mathsf{S}}$ uses them to generate two decommitments decom_0 and decom_1. Note that number of augmented queries done by $\widetilde{\mathsf{S}}$ is bounded by $\text{poly}(n) \cdot 2^{\lceil c(n)/d(n) \rceil} \leq \text{poly}(n) \cdot \sqrt[4]{s(n)} \in O(\sqrt[3]{s(n)})$.

Let $\mathsf{Balanced}_n$ be as in Claim 61, and for $r_{\mathsf{S}}, r_{\mathsf{R}} \in \{0,1\}^m$, let $\mathsf{trans}_n^{r_{\mathsf{S}}, r_{\mathsf{R}}} = \langle (\mathsf{S}^\tau(0, r_{\mathsf{S}}), \mathsf{R}^\tau(r_{\mathsf{R}}))(1^n) \rangle_{\mathsf{trans}}$. Since Com is statistically hiding, it follows (see Claim 61) that

$$\Pr_{r_{\mathsf{S}} \leftarrow \{0,1\}^m, r_{\mathsf{R}} \leftarrow \{0,1\}^m} \left[\mathsf{trans}_n^{r_{\mathsf{S}}, r_{\mathsf{R}}} \notin \mathsf{Balanced}_n\right] = \text{neg}(n) \tag{30}$$

Let $\mathsf{Fail}_n = \left\{(h, r_{\mathsf{S}}, r_{\mathsf{R}}) \in \mathcal{H} \times (\{0,1\}^m)^2 \colon \mathsf{Inv}^{\tau, \mathsf{Sam}^{\tau,h}}(1^n, 1^n, d(n) - 1, \varepsilon(n), \mathsf{trans}_n^{r_{\mathsf{S}}, r_{\mathsf{R}}}) = \perp\right\}$. Theorem 21 yields that

$$\Pr_{h \leftarrow \mathcal{H}, r_{\mathsf{S}} \leftarrow \{0,1\}^m, r_{\mathsf{R}} \leftarrow \{0,1\}^m} \left[\mathsf{Fail}_n\right] \leq \varepsilon(n) \tag{31}$$

It is also easy to verify that (see again Claim 61) that

$$\mathop{\mathrm{E}}_{\substack{h\leftarrow\mathcal{H},r_{\mathsf{S}}\leftarrow\{0,1\}^m\\ r_{\mathsf{R}}\leftarrow\{0,1\}^m}}\left[\neg\mathsf{TwoOpenings}_n^h(r_{\mathsf{S}},r_{\mathsf{R}})\mid \mathsf{trans}_n^{r_{\mathsf{R}}}\in\mathsf{Balanced}_n\wedge(h,r_{\mathsf{S}},r_{\mathsf{R}})\notin\mathsf{Fail}_n\right]=\mathrm{neg}(n) \quad (32)$$

We conclude that

$$\begin{aligned}
&\Pr_{h\leftarrow\mathcal{H}}\left[\mathsf{NoBreak}_n^h\right]\\
&=\Pr_{h\leftarrow\mathcal{H}}\left[\Pr_{r_{\mathsf{S}}\leftarrow\{0,1\}^m,r_{\mathsf{R}}\leftarrow\{0,1\}^m}\left[\neg\mathsf{TwoOpenings}_n^h(r_{\mathsf{S}},r_{\mathsf{R}})\right]\geq\delta(n)\right]\\
&\leq\frac{\mathrm{E}_{h\leftarrow\mathcal{H},r_{\mathsf{S}}\leftarrow\{0,1\}^m,r_{\mathsf{R}}\leftarrow\{0,1\}^m}\left[\neg\mathsf{TwoOpenings}_n^h(r_{\mathsf{S}},r_{\mathsf{R}})\right]}{\delta(n)}\\
&\leq\frac{1}{\delta(n)}\cdot\left(\Pr_{h\leftarrow\mathcal{H},r_{\mathsf{S}}\leftarrow\{0,1\}^m,r_{\mathsf{R}}\leftarrow\{0,1\}^m}\left[\mathsf{trans}_n^{r_{\mathsf{S}},r_{\mathsf{R}}}\notin\mathsf{Balanced}_n\vee(h,r_{\mathsf{S}},r_{\mathsf{R}})\in\mathsf{Fail}_n\right]\right.\\
&\quad\left.+\mathop{\mathrm{E}}_{h\leftarrow\mathcal{H},r_{\mathsf{R}}\leftarrow\{0,1\}^m}\left[\neg\mathsf{TwoOpenings}_n^h(r_{\mathsf{S}},r_{\mathsf{R}})\mid\mathsf{trans}_n^{h,r_{\mathsf{S}},r_{\mathsf{R}}}\in\mathsf{Balanced}\wedge(h,r_{\mathsf{S}},r_{\mathsf{R}})\notin\mathsf{Fail}_n\right]\right)\\
&\leq\frac{\varepsilon(n)+\mathrm{neg}(n)}{\delta(n)}\in O(1/n^2),
\end{aligned}$$

where the last inequality follows from Equations (30) to (32) □

7 Implications to Other Cryptographic Protocols

Our lower bounds on the round complexity and the communication complexity of statistically hiding commitment schemes imply similar lower bounds for several other cryptographic protocols. Specifically, our results can be extended to any cryptographic protocol that can be used to construct a weakly-binding statistically hiding commitment scheme in a fully-black-box manner while essentially preserving the round complexity or communication complexity of the underlying protocol. In this section we derive new such lower bound for interactive hashing, oblivious transfer, and single-server private information retrieval protocols. For simplicity, we state these lower bounds for constructions that are security preserving (i.e., $O(n)$-security-parameter expanding), and we note that more general statements, as in Theorem 56, could be derived as well.

We note that our lower bound proof for the round complexity of statistically hiding commitment schemes did not rely on any malicious behavior by the receiver. Therefore, our lower bound holds even for schemes in which the statistical hiding property is guaranteed only against honest receivers. Similarly, our lower bound proof for the communication complexity of statistically hiding commitment schemes did not rely on any malicious behavior by the sender during the commit stage. Therefore, our lower bound holds even for schemes in which the (weak) binding property is guaranteed only against honest senders.

7.1 Interactive Hashing

Interactive hashing was introduced by Naor et al. [57] and is a protocol that allows a sender S to commit to a value y while only revealing to the receiver R the value $(h, z = h(y))$, where h is

a 2-to-1 hash function chosen interactively during the protocol.[32] The two security properties of interactive hashing are binding (S is bound by the protocol to producing at most one value of y which is consistent with the transcript) and hiding (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, and showed that it implies in a fully black-box manner a statistically-hiding commitment scheme. The construction of Naor et al. preserves the communication complexity of the underlying interactive hashing protocol, but it does not preserve the round complexity. However, in subsequent work [34, 47] it was shown that it is in fact possible to preserve the number of rounds. Combined with our lower bounds on the round complexity and communication complexity of statistically-hiding commitment schemes, this directly implies the following corollary:

Corollary 62. *Any $O(n)$-security-parameter expanding fully black-box construction of an interactive hashing protocol from a family of trapdoor permutations has round complexity $\Omega(n/\log n)$ and communication complexity $\Omega(n)$.*

We note that Wee [73] showed that a restricted class of fully black-box constructions of interactive hashing from one-way permutations has $\Omega\left(n/\log n\right)$ rounds. Thus, Corollary 62 extends Wee's lower bound both to include the most general form of such constructions, and to trapdoor permutations.

7.2 Oblivious Transfer

Oblivious transfer (OT), introduced by Rabin [63], is a fundamental primitive in cryptography. In particular, it was shown to imply secure multiparty computation [27, 44, 75]. OT has several equivalent formulations, and we consider the formulation of $\binom{2}{1}$-OT, defined by Even, Goldreich, and Lempel [14]. $\binom{2}{1}$-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 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 the sender, Fischlin [15] showed how to construct a weakly-binding statistically hiding commitment scheme. The construction is fully black-box and preserves the round complexity and the communication complexity. In addition, Wolf and Wullschleger [74] showed that any $\binom{2}{1}$-OT protocol that guarantees statistical security for the sender can be transformed into a $\binom{2}{1}$-OT protocol that guarantees statistical security for the receiver. Their transformation is full black-box and preserves the round complexity and the communication complexity. Thus, by combining these with our lower bounds we obtain the following corollary:

Corollary 63. *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 round complexity $\Omega(n/\log n)$ and communication complexity $\Omega(n)$.*

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 [55]). Hence, as for statistically

[32] Several extensions to this definition were suggested, see [34, 58].

hiding commitment schemes, 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.

7.3 Single-Server Private Information Retrieval

A single-server private information retrieval (PIR) scheme [10] 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 [7, 9, 19, 51, 48], and a recent survey by Ostrovsky and Skeith [59]). The only non-trivial construction based on general computational assumptions is due to Kushilevitz and Ostrovsky [49]. 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 fully black-box and preserves the number of rounds. Thus, by combining this with our lower bound on the round complexity for statistically hiding commitment schemes, we obtain the following corollary:

Corollary 64. *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 communication complexity $\Omega(n/\log n)$.*

Corollary 62 yields in particular an $\Omega(n/\log n)$ lower bound on the communication complexity of such single-server PIR protocols (and, in particular, on the number of bits that the server must communicate). We note that the construction of Beimel et al. does not preserve the communication complexity of the underlying PIR protocol. Therefore, our lower bound on the communication complexity of statistically hiding commitment schemes cannot be directly used for deriving a similar lower bound for PIR protocols. Nevertheless, in Appendix A we refine the construction of Beimel et al. to a construction which, in particular, preserves the communication complexity. We thus obtain the following corollary:

Corollary 65. *In 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, the server communicates $\Omega(n)$ bits.*

Acknowledgment

We thank Mohammad Mahmoody and Rafael Pass for useful discussions.

References

[1] W. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. In *Advances in Cryptology – EUROCRYPT 2001*, 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 (STOC)*, 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] Z. Brakerski, J. Katz, G. Segev, and A. Yerukhimovich. Limits on the power of zero-knowledge proofs in cryptographic constructions. In *Proceedings of the 8th Theory of Cryptography Conference*, pages 559–578, 2011.

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

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

[8] 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.

[9] 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.

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

[11] 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.

[12] Y. Dodis, I. Haitner, and A. Tentes. On the instantiability of hash-and-sign rsa signatures. In *Theory of Cryptography, Ninth Theory of Cryptography Conference, TCC 2012*, pages 112–132, 2012.

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

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

[15] 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.

[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 (FOCS)*, pages 305–313, 2000.

[17] 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 (STOC)*, pages 417–425, 2003.

[18] 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.

[19] 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.

[20] 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 (FOCS)*, pages 325–335, 2000.

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

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

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

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

[27] 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 (STOC)*, pages 218–229, 1987.

[28] 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.

[29] S. D. Gordon, H. Wee, D. Xiao, and A. Yerukhimovich. On the round complexity of zero-knowledge proofs based on one-way permutations. In *LATINCRYPT*, pages 189–204, 2010.

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

[31] I. Haitner. Implementing oblivious transfer using collection of dense trapdoor permutations. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, pages 394–409, 2004.

[32] I. Haitner and T. Holenstein. On the (im)possibility of key dependent encryption. In *Theory of Cryptography, Sixth Theory of Cryptography Conference, TCC 2009*, 2009.

[33] I. Haitner and O. Reingold. A new interactive hashing theorem. To appear in *Journal of Cryptology*.

[34] I. Haitner and O. Reingold. A new interactive hashing theorem. In *Proceedings of the 21st Annual IEEE Conference on Computational Complexity*, 2007.

[35] 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 2005*, pages 58–77, 2005.

[36] I. Haitner, J. J. Hoch, O. Reingold, and G. Segev. Finding collisions in interactive protocols – A tight lower bound on the round complexity of statistically hiding commitments. In *Proceedings of the 48th Annual Symposium on Foundations of Computer Science (FOCS)*, 2007.

[37] I. Haitner, J. J. Hoch, and G. Segev. A linear lower bound on the communication complexity of single-server private information retrieval. In *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008*, pages 394–409, 2008.

[38] I. Haitner, M. Nguyen, S. J. Ong, O. Reingold, and S. Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM Journal on Computing*, 39(3):1153–1218, 2009.

[39] I. Haitner, O. Reingold, S. Vadhan, and H. Wee. Inaccessible entropy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, 2009.

[40] I. Haitner, M. Mahmoody-Ghidary, and D. Xiao. On basing constant-round statistically hiding commitments on np-hardness. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity*, pages 76–87, 2010.

[41] 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.

[42] 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.

[43] 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 (STOC)*, pages 44–61, 1989.

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

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

[46] 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 (FOCS)*, pages 535–542, 1999.

[47] 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.

[48] 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.

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

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

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

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

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

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

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

[56] 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 (STOC)*, pages 33–43, 1989.

[57] 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.

[58] 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 Symposium on Foundations of Computer Science (FOCS)*, pages 3–14, 2006.

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

[60] R. Pass and M. Venkitasubramaniam. Private coins versus public coins in zero-knowledge proof systems. In *Theory of Cryptography, Seventh Theory of Cryptography Conference, TCC 2010*, pages 588–605, 2010.

[61] K. Pietrzak. Compression from collisions, or why crhf combiners have a long output. In *Advances in Cryptology – CRYPTO 2008*, pages 413–432, 2008.

[62] K. Pietrzak, A. Rosen, and G. Segev. Lossy functions do not amplify well. In *Theory of Cryptography, Ninth Theory of Cryptography Conference, TCC 2012*, pages 458–475, 2012.

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

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

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

[66] A. Rosen. A note on constant-round zero-knowledge proofs for NP. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, pages 191–202, 2004.

[67] A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In *Proceedings of the 6th Theory of Cryptography Conference*, pages 419–436, 2009.

[68] A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. *SIAM Journal on Computing*, 39(7):3058–3088, 2010.

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

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

[71] 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.

[72] A. Srinivasan and D. Zuckerman. Computing with very weak random sources. *SIAM Journal on Computing*, 28(4):1433–1459, 1999.

[73] H. Wee. One-way permutations, interactive hashing and statistically hiding commitments. In *Theory of Cryptography, Fourth Theory of Cryptography Conference, TCC 2007*, pages 419–433, 2007.

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

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

A From PIR to Statistically-Hiding Commitments

The relation between single-server PIR and commitment schemes was first explored by Beimel, Ishai, Kushilevitz, and Malkin [2], who showed that any single-server PIR protocol in which the server communicates at most $n/2$ bits to the user (where n is the size of the server's database), can be used to construct a weakly binding statistically hiding bit-commitment scheme. In particular, this served as the first indication that the existence of low-communication PIR protocols implies the existence of one-way functions. In this section we refine the relation between these two fundamental primitives by improving their reduction. Our improvements are the following:

1. The construction of [2] preserves the round complexity of the underlying single-server PIR, but it does not preserve its communication complexity. In their construction the sender is always required to send $\Omega(n)$ bits during the commit stage of the commitment scheme. We show that it is possible to preserve both the round complexity and the communication complexity. In our construction the number of bits sent by the sender during the commit stage of the commitment scheme is essentially the number of bits sent by the server in the PIR protocol.

2. The construction of [2] requires an execution of the single-server PIR protocol for every committed bit (that is, they constructed a bit-commitment scheme). We show that it is possible to commit to a super-logarithmic number of bits while executing the underlying single-server PIR protocol only once.

3. The construction of [2] was presented for single-server PIR protocols in which the server communicates at most $n/2$ bits. Our construction applies to any single-server PIR protocol in which the server communicates up to $n - \omega(\log n)$ bits.

In the remainder of this section we first state the theorem resulting from our construction. Then, we formally define single-server PIR, provide a few additional preliminaries, and present our construction.

Theorem 66. *Assume there exists a single-server PIR protocol in which the server communicates $n - k(n)$ bits, where n is the size of the server's database and $k(n) \geq 2d(n)$ for $d(n) \in \omega(\log n)$.*

Then, there exists a weakly binding statistically hiding commitment scheme for $d(n)/6$ bits, in which the sender communicates at most $n - k(n) + 2d(n)$ bits during the commit stage. Moreover, the construction is fully black box.

An overview of the construction. Let $(\mathsf{Server}, \mathsf{User})$ be a single-server PIR protocol in which the server communicates $n - \omega(\log n)$ bits, where n is the size of the server's database. Consider the following commitment scheme to a string s. The commit stage consists of the sender and the receiver first choosing random inputs $x \in \{0,1\}^n$ and $i \in [n]$, respectively, and executing the PIR protocol $(\mathsf{Server}, \mathsf{User})$ on these inputs (that is, the sender plays the role of the server with database x, and the receiver plays the role of the user with index i). As a consequence, the receiver obtains a bit x_i, which by the correctness of the PIR protocol is the i'th bit of x. Notice that since the sender communicated only $n - \omega(\log n)$, the random variable corresponding to x still has $\omega(\log n)$ min-entropy from the receiver's point of view. We take advantage of this fact, and have the sender choose a uniform seed t for a strong-extractor Ext, and send the pair $(t, \mathrm{Ext}(x,t) \oplus s)$ to the receiver. That is, we exploit the remaining min-entropy of the database x in order to mask the committed

string s in a statistical manner. In the reveal stage, the sender sends the pair (x, s) to the receiver. The binding property follows from the security of the PIR protocol: in the reveal stage, the sender must send a value x whose i'th bit is consistent with the bit obtained by the receiver during the commit stage – but this bit not known to the sender.

A.1 Single-Server Private Information Retrieval — Definition

A single-server Private Information Retrieval (PIR) scheme 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. The user wishes to retrieve the i'th entry of the database, without revealing the index i to the server. More formally, a single-server PIR scheme is defined via a pair of probabilistic polynomial-time Turing-machines $(\mathsf{Server}, \mathsf{User})$ such that:

- Server receives as input a string $x \in \{0,1\}^n$. Following its interaction it does not have any output.
- User receives as input an index $i \in [n]$. Following its interaction it outputs a value $b \in \{0, 1, \perp\}$.

Denote by $b \leftarrow \langle \mathsf{Server}(x), \mathsf{User}(i) \rangle$ the experiment in which Server and User interact (using the given inputs and uniformly chosen random coins), and then User outputs the value b. It is required that there exists a negligible function $\nu(n)$, such that for all sufficiently large n, and for every string $x = x_1 \circ \cdots \circ x_n \in \{0,1\}^n$, it holds that $x_i \leftarrow \langle \mathsf{Server}(x), \mathsf{User}(i) \rangle$ with probability at least $1 - \nu(n)$ over the random coins of both Server and User.

In order to define the security properties of such schemes, we first introduce the following notation. Given a single-server PIR scheme $(\mathsf{Server}, \mathsf{User})$ and a Turing-machine $\widetilde{\mathsf{Server}}$ (a malicious server), we denote by $\mathsf{view}_{\langle \widetilde{\mathsf{Server}}, \mathsf{User}(i) \rangle}(n)$ the distribution on the view of $\widetilde{\mathsf{Server}}$ when interacting with $\mathsf{User}(i)$ where $i \in [n]$. This view consists of its random coins and of the sequence of messages it receives from User, and the distribution is taken over the random coins of both $\widetilde{\mathsf{Server}}$ and User.

Definition 67. *A single-server PIR scheme $(\mathsf{Server}, \mathsf{User})$ is secure if for every probabilistic polynomial-time Turing-machines $\widetilde{\mathsf{Server}}$ and D, and for every two sequences of indices $\{i_n\}_{n=1}^{\infty}$ and $\{j_n\}_{n=1}^{\infty}$ where $i_n, j_n \in [n]$ for every n, it holds that*

$$\left| \Pr\left[v \leftarrow \mathsf{view}_{\langle \widetilde{\mathsf{Server}}, \mathsf{User}(i_n) \rangle}(n) \colon \mathsf{D}(v) = 1 \right] - \Pr\left[v \leftarrow \mathsf{view}_{\langle \widetilde{\mathsf{Server}}, \mathsf{User}(j_n) \rangle}(n) : \mathsf{D}(v) = 1 \right] \right| \leq \nu(n),$$

for some negligible function $\nu(n)$ and for all sufficiently large n.

A.2 Additional Preliminaries

The min-entropy of a distribution D over a set $\mathcal{X}$ is defined as $\mathrm{H}_\infty(D) = \min_{x \in \mathcal{X}} \log 1/\Pr_D[x]$. The following standard fact (cf., [70, Fact 2.6]) will be useful for us in analyzing statistically close distributions.

Fact 68. *Let P and Q be two distributions with $\mathrm{SD}(P, Q) < \epsilon$, then*

$$\Pr_{x \leftarrow P}\left[(1 - \sqrt{\epsilon}) \cdot \Pr_P[x] < \Pr_Q[x] < (1 + \sqrt{\epsilon}) \cdot \Pr_P[x] \right] \geq 1 - 2\sqrt{\epsilon}.$$

Definition 69. *A function* $E\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k,ϵ)-extractor*, if for every distribution* X *over* $\{0,1\}^n$ *with* $\mathrm{H}_\infty(X) \geq k$ *the distribution* $E(X,U_d)$ *is* ϵ*-close to uniform.* E *is a* strong (k,ϵ)-extractor*, if the function* $E'(x,y) = y \circ E(x,y)$ *is a* (k,ϵ)*-extractor (where* $\circ$ *denotes concatenation).*

In our construction of a statistically hiding commitment from single-server PIR, we will be using the following explicit construction of strong extractors, which is an immediate corollary of [72, Corollary 3.4].

Proposition 70. *For any* $k \in \omega(\log n)$*, there exists an explicit strong* $(k, 2^{1-k})$*-extractor* $\mathrm{Ext}\colon \{0,1\}^n \times \{0,1\}^{3k} \to \{0,1\}^{k/2}$.

A.3 The Construction

Fix $d(n)$, $k(n)$ and a single-server PIR protocol $\mathsf{P} = (\mathsf{Server}, \mathsf{User})$ as in Theorem 66. Protocol 71 describes our construction of the commitment scheme $\mathsf{Com} = (\mathsf{S}, \mathsf{R})$. In the construction we use a strong $\big(d(n)/3, 2^{1-d(n)/3}\big)$-extractor $\mathrm{Ext}\colon \{0,1\}^n \times \{0,1\}^{d(n)} \to \{0,1\}^{d(n)/6}$ whose existence is guaranteed by Proposition 70.

Protocol 71 (Protocol $\mathsf{Com} = (\mathsf{S}, \mathsf{R})$)**.**

Common input: security parameter 1^n.

Sender's input: $s \in \{0,1\}^{d(n)/6}$.

Commit stage:

1. S *chooses a uniformly distributed* $x \in \{0,1\}^n$.
2. R *chooses a uniformly distributed index* $i \in [n]$.
3. S *and* R *execute the single-server PIR protocol* $(\mathsf{Server}, \mathsf{User})$ *for database of length* n*, where* S *acts as the server with input* x *and* R *acts as the user with input* i*. As a result,* R *obtains a bit* $x_i \in \{0,1\}$.
4. S *chooses a uniformly distributed seed* $t \in \{0,1\}^{d(n)}$*, computes* $y = \mathrm{Ext}(x,t) \oplus s$*, and sends* (t,y) *to* R.

Reveal stage:

1. S *sends* (s,x) *to* R.
2. *If the* i*'th bit of* x *equals* x_i *and* $y = \mathrm{Ext}(x,t) \oplus s$*, then* R *outputs* s.
   Otherwise, R *outputs* $\perp$.

The correctness of Com follows directly from the correctness of the PIR protocol. In addition, notice that the total number of bits communicated by the sender in the commit stage is the total number of bits that the server communicates in the PIR protocol plus the seed length and the output length of the extractor Ext. Thus, the sender communicates less than $n - k(n) + 2d(n)$ bits

during the commit stage. In Lemma 72 we prove that Com is statistically hiding, and in Lemma 74 we prove that Com is weakly binding. We note that the proof of hiding does not rely on any computational properties of the underlying PIR protocol, but only on the assumed bound on the number of bits communicated by the server.

Lemma 72. Com *is statistically hiding.*

Proof. We have to show that for any computationally unbounded receiver R^* and for any two strings s_0 and s_1, the statistical distance between the distributions $\{\mathsf{view}_{\langle \mathsf{S}(s_0),\mathcal{R}^*\rangle}(n)\}$ and $\{\mathsf{view}_{\langle \mathsf{S}(s_1),\mathcal{R}^*\rangle}(n)\}$ (see Definition 11) is negligible in n. The transcript of the commit stage consists of the transcript $\mathsf{trans}_\mathsf{P}$ of the execution of P and of the pair $(t, \mathrm{Ext}(x,t) \oplus s)$, where s is the committed string. Note that since $\mathsf{trans}_\mathsf{P}$ is independent of the committed string, it is sufficient to prove that the statistically distance between the distribution of $(t, \mathrm{Ext}(x,t))$ given $\mathsf{trans}_\mathsf{P}$ and the uniform distribution is negligible in n.

We argue that due to the bound on the number of bits communicated by the server in P, then even after executing P, the database x still has sufficient min-entropy in order to guarantee that $(t, \mathrm{Ext}(x,t))$ is sufficiently close to uniform. More specifically, let R^* be an all-powerful receiver (recall that without loss of generality such an R^* is deterministic), and denote by X the random variable corresponding to the value x in Com. The following claim states the with high probability X has high min-entropy from R^*'s point of view.

Claim 73. *It holds that*

$$\Pr_{\mathsf{trans}_\mathsf{P} \leftarrow \mathsf{Com}}\left[\mathrm{H}_\infty(X \mid \mathsf{trans}_\mathsf{P}) < \frac{k(n)}{6}\right] < 2^{-\frac{k(n)}{4}},$$

where $\mathsf{trans}_\mathsf{P}$ *is the transcript of the embedded execution of* P *in* Com.

Proof. For any value of r, the random coins used by S in the execution of P, let $f_r : \{0,1\}^n \mapsto \{0,1\}^{n-k(n)}$ be the function that maps x to the value of $\mathsf{trans}_\mathsf{P}$ generated by the interaction of $(\mathsf{S}(x,r), \mathsf{R}^*)$, and let $\mathrm{Col}(x,r) := \{x' \in \{0,1\}^n : f_r(x') = f_r(x)\}$. Since f_r has at most $2^{n-k(n)}$ possible outputs, it follows that

$$\Pr_{x,r}\left[|\mathrm{Col}(x,r)| < 2^{\frac{k(n)}{2}+1}\right] < \frac{2^{n-k(n)} \cdot 2^{\frac{k(n)}{2}+1}}{2^n} = 2^{1-\frac{k(n)}{2}}. \tag{33}$$

Let

$$\mathrm{BAD} = \left\{\mathsf{trans}_\mathsf{P} \;:\; \Pr_{x,r}\left[|\mathrm{Col}(x,r)| < 2^{\frac{k(n)}{2}+1} \;\middle|\; \mathsf{trans}_\mathsf{P}\right] > 2^{\frac{k(n)}{4}} \cdot 2^{1-\frac{k(n)}{2}}\right\},$$

a a standard averaging argument yields that

$$\Pr_{\mathsf{trans}_\mathsf{P} \leftarrow \mathsf{Com}}[\mathsf{trans}_\mathsf{P} \in \mathrm{BAD}] \leq 2^{-\frac{k(n)}{4}} \tag{34}$$

Denote by U_r the random variable corresponding to r in the execution of Com. The following

holds every value of x and $\mathsf{trans}_{\mathsf{P}}$:

$$\begin{aligned}
&\Pr\left[X = x \mid \mathsf{trans}_{\mathsf{P}}\right] \\
&\quad = \Pr\left[X = x \wedge |\mathrm{Col}(X, U_r)| < 2^{\frac{k(n)}{2}+1} \;\middle|\; \mathsf{trans}_{\mathsf{P}}\right] \\
&\qquad + \Pr\left[X = x \wedge |\mathrm{Col}(X, U_r)| \geq 2^{\frac{k(n)}{2}+1} \;\middle|\; \mathsf{trans}_{\mathsf{P}}\right] \\
&\quad \leq \Pr\left[|\mathrm{Col}(X, U_r)| < 2^{\frac{k(n)}{2}+1} \;\middle|\; \mathsf{trans}_{\mathsf{P}}\right] + 2^{-\left(\frac{k(n)}{2}+1\right)}.
\end{aligned} \tag{35}$$

Note that if $\mathrm{H}_\infty(X \mid \mathsf{trans}_{\mathsf{P}}) < k(n)/6$ for some $\mathsf{trans}_{\mathsf{P}}$, then there exists an x for which

$$\Pr\left[X = x \mid \mathsf{trans}_{\mathsf{P}}\right] \geq 2^{-\frac{k(n)}{6}},$$

and therefore Equation (35) implies that

$$\Pr\left[|\mathrm{Col}(X, U_r)| < 2^{\frac{k(n)}{2}+1} \;\middle|\; \mathsf{trans}_{\mathsf{P}}\right] > 2^{-\frac{k(n)}{6}} - 2^{-\left(\frac{k(n)}{2}+1\right)} > 2^{1-\frac{k(n)}{4}} \tag{36}$$

Thus,

$$\begin{aligned}
\Pr_{\mathsf{trans}_{\mathsf{P}} \leftarrow \mathsf{Com}}\left[\mathrm{H}_\infty(X \mid \mathsf{trans}_{\mathsf{P}}) < \frac{k(n)}{6}\right] &\leq \Pr_{\mathsf{trans}_{\mathsf{P}} \leftarrow \mathsf{Com}}\left[\Pr\left[|\mathrm{Col}(X, U_r)| < 2^{\frac{k(n)}{2}+1} \;\middle|\; \mathsf{trans}_{\mathsf{P}}\right] > 2^{1-\frac{k(n)}{4}}\right] \\
&\leq \Pr_{\mathsf{trans}_{\mathsf{P}} \leftarrow \mathsf{Com}}\left[\mathsf{trans}_{\mathsf{P}} \in \mathrm{BAD}\right] \\
&\leq 2^{-\frac{k(n)}{4}}.
\end{aligned}$$

□

Since $d(n) \in \omega(\log n)$ and $k(n)/6 \geq d(n)/3$, Claim 73 implies that with probability $1 - \mathrm{neg}(n)$, the extractor Ext guarantees that the statistical distance between the pair $(t, \mathrm{Ext}(x,t))$ (given $\mathsf{trans}_{\mathsf{P}}$) and the uniform distribution is at most $2^{1-d(n)/3}$ (which is again negligible in n). Therefore Com is statistically hiding. More specifically, for every string $s \in \{0,1\}^{d(n)/6}$ it holds that

$$\begin{aligned}
&\mathrm{SD}\left(\{\mathsf{trans}_{\mathsf{P}}, t, \mathrm{Ext}(X,t) \oplus s\}, \{\mathsf{trans}_{\mathsf{P}}, U_{7d(n)/6}\}\right) \\
&\quad \leq \Pr\left[\mathrm{H}_\infty(X \mid \mathsf{trans}_{\mathsf{P}}) < \frac{k(n)}{6}\right] \\
&\qquad + \mathrm{SD}\left(\{\mathsf{trans}_{\mathsf{P}}, t, \mathrm{Ext}(X,t) \oplus s\}, \{\mathsf{trans}_{\mathsf{P}}, U_{7d(n)/6}\} \;\middle|\; \mathrm{H}_\infty(X \mid \mathsf{trans}_{\mathsf{P}}) \geq \frac{k(n)}{6}\right) \\
&\quad \leq 2^{-\frac{k(n)}{4}} + 2^{1-\frac{d(n)}{3}}.
\end{aligned} \tag{37}$$

Therefore, for any two strings $s_0, s_1 \in \{0,1\}^{d(n)/6}$ it holds that

$$\begin{aligned}
&\mathrm{SD}\left(\left\{\mathsf{view}_{\langle \mathsf{S}(s_0), \mathcal{R}^* \rangle}(n)\right\}, \left\{\mathsf{view}_{\langle \mathsf{S}(s_1), \mathcal{R}^* \rangle}(n)\right\}\right) \\
&\quad = \mathrm{SD}\left(\{\mathsf{trans}_{\mathsf{P}}, t, \mathrm{Ext}(X,t) \oplus s_0\}, \{\mathsf{trans}_{\mathsf{P}}, t, \mathrm{Ext}(X,t) \oplus s_1\}\right) \\
&\quad \leq 2 \cdot \left(2^{-\frac{k(n)}{4}} + 2^{1-\frac{d(n)}{3}}\right),
\end{aligned}$$

which is negligible in n as required. □

Let U_r be the random variable taking the value of r in the execution of Com. By the above equation, the following holds every value of x and $\mathsf{trans}_\mathsf{P}$.

$$\begin{aligned}
&\Pr[X = x \mid \mathsf{trans}_\mathsf{P}] \\
&= \Pr[X = x \wedge |\mathrm{Col}(X, U_r)| < 2^{\frac{k(n)}{2}+1} \mid \mathsf{trans}_\mathsf{P}] + \Pr[X = x \wedge |\mathrm{Col}(X, U_r)| \geq 2^{\frac{k(n)}{2}+1} \mid \mathsf{trans}_\mathsf{P}] \\
&\leq \Pr[|\mathrm{Col}(X, U_r)| < 2^{\frac{k(n)}{2}+1} \mid \mathsf{trans}_\mathsf{P}] + 2^{-(\frac{k(n)}{2}+1)}.
\end{aligned}$$

We conclude that,

$$\begin{aligned}
&\Pr[\mathsf{trans}_\mathsf{P} \leftarrow \mathsf{Com}\colon \mathrm{H}_\infty(X \mid \mathsf{trans}_\mathsf{P}) < \frac{k(n)}{2}] \\
&\quad = \Pr\big[\mathsf{trans}_\mathsf{P} \leftarrow \mathsf{Com}\colon \max_{x \in \{0,1\}^n} \{\Pr[X = x \mid \mathsf{trans}_\mathsf{P}]\} > 2^{\frac{k(n)}{2}}\big] \\
&\quad \leq \Pr\big[\mathsf{trans}_\mathsf{P} \leftarrow \mathsf{Com}\colon \Pr[|\mathrm{Col}(X, U_r)| < 2^{\frac{k(n)}{2}+1} \mid \mathsf{trans}_\mathsf{P}]\big] \\
&\quad = \Pr[|\mathrm{Col}(X, U_r)| < 2^{\frac{k(n)}{2}+1}] < 2^{1-k(n)/2}.
\end{aligned}$$

Recall that R^*'s view is the concatenation of the values of $\mathsf{trans}_\mathsf{P}$, $\mathrm{Ext}(x,t) \oplus s$ and t. Using standard reduction it follows that for any two strings $s_1, s_2 \in \{0,1\}^{\lfloor d(n)/2 \rfloor}$, the statistical difference between view$[s_1]$ and view$[s_2]$ is at most twice the statistical difference between $(\mathsf{trans}_\mathsf{P}, \mathrm{Ext}(x,t), t)$ and $(\mathsf{trans}_\mathsf{P}, U_{\lfloor d(n)/2 \rfloor}, t)$, where the values of $\mathsf{trans}_\mathsf{P}$, x and t are induced by a random execution of Com. The following concludes the proof of the lemma by showing that the latter distance is negligible.

$$\begin{aligned}
&\mathrm{SD}\Big((\mathsf{trans}_\mathsf{P}, \mathrm{Ext}(X,t), t), (\mathsf{trans}_\mathsf{P}, U_{\lfloor d(n)/2 \rfloor}, t)\Big) \\
&\quad \leq \Pr\left[\mathrm{H}_\infty(X \mid \mathsf{trans}_\mathsf{P}) < \frac{k(n)}{2}\right] \\
&\qquad + \mathrm{SD}\left((\mathsf{trans}_\mathsf{P}, \mathrm{Ext}(X,t), t), (\mathsf{trans}_\mathsf{P}, U_{\lfloor d(n)/2 \rfloor}, t) \mid \mathrm{H}_\infty(X \mid \mathsf{trans}_\mathsf{P}) \geq \frac{k(n)}{2}\right) \\
&\quad \leq 2^{1-\frac{k(n)}{2}} + 2^{2-\frac{d(n)}{3}} = \mathrm{neg}(n).
\end{aligned}$$

Lemma 74. Com *is weakly binding.*

Proof. We show that Com is $(1 - 1/n^2)$-binding. Given any malicious sender $\widetilde{\mathsf{S}}$ that violates the binding of the commitment scheme Com with probability at least $1 - 1/n^2$, we construct a malicious server $\widetilde{\mathsf{Server}}$ that breaks the security of the single-server PIR protocol P.

Let $\widetilde{\mathsf{S}}$ be a polynomial-time malicious sender that violates the binding of Com with probability at least $1 - 1/n^2$. As an intermediate step, we first construct a malicious server that has a non-negligible advantage in predicting a uniformly chosen index held by the user in P. More specifically, we construct a malicious server $\widetilde{\mathsf{Server}}$ and a predictor D' such that

$$\Pr\left[v \leftarrow \mathsf{view}_{\langle \widetilde{\mathsf{Server}}, \mathsf{User}(i) \rangle}(n)\colon \mathsf{D}'(v) = i\right] \geq \frac{1}{n} + \frac{1}{n^2},$$

where the probability is taken over the uniform choice of $i \in [n]$ and over the coin tosses of $\widetilde{\mathsf{Server}}$, D' and User. Recall that $\mathsf{view}_{\langle \widetilde{\mathsf{Server}}, \mathsf{User}(i) \rangle}(n)$ denotes the distribution on the view of $\widetilde{\mathsf{Server}}$ when

interacting with $\mathsf{User}(i)$ where $i \in [n]$. This view consists of its random coins and of the sequence of messages it receives from User.

The malicious server $\widetilde{\mathsf{Server}}$ follows the malicious sender $\widetilde{\mathsf{S}}$ in the embedded execution of P in Com. Following the interaction, $\widetilde{\mathsf{Server}}$ proceeds the execution of $\widetilde{\mathsf{S}}$ to obtain a pair (t, y) and two decommitments (x_1, s_1) and (x_2, s_2). If $x_1 = x_2$, then $\widetilde{\mathsf{Server}}$ fails. Otherwise, denote by $j \in [n]$ the minimal index such that $x_1[j] \neq x_2[j]$. Now, the predictor D' outputs a uniformly distributed value i' from the set $[n] \setminus \{j\}$.

In order to analyze the success probability in predicting i, note that if (x_1, s_1) and (x_2, s_2) are valid decommitments and $s_1 \neq s_2$ (i.e., $\widetilde{\mathsf{S}}$ broke the binding of Com), then it must hold that $x_1 \neq x_2$. In this case, let $j \in [n]$ be the minimal index such that $x_1[j] \neq x_2[j]$, then it must be the case that $i \neq j$, as otherwise R will not accept the two decommitments. Therefore, when the predictor D' outputs a uniformly distributed $i' \in [n] \setminus \{j\}$, it will output i with probability $1/(n-1)$. Thus,

$$\begin{aligned} \Pr\left[v \leftarrow \mathsf{view}_{\langle \widetilde{\mathsf{Server}}, \mathsf{User}(i)\rangle}(n) \colon \mathsf{D}'(v) = i\right] &\geq \left(1 - \frac{1}{n^2}\right) \cdot \frac{1}{n-1} \\ &= \frac{n+1}{n^2} \\ &= \frac{1}{n} + \frac{1}{n^2}. \end{aligned} \tag{38}$$

In the remainder of the proof we apply a rather standard argument in order to be fully consistent with Definition 67 of the security of single-server PIR. That is, we show that there exists a pair of indices $i, j \in [n]$, a malicious server $\widetilde{\mathsf{Server}}$ and a distinguisher D such that

$$\left|\Pr\left[v \leftarrow \mathsf{view}_{\langle \widetilde{\mathsf{Server}}, \mathsf{User}(i)\rangle}(n) \colon \mathsf{D}(v) = 1\right] - \Pr\left[v \leftarrow \mathsf{view}_{\langle \widetilde{\mathsf{Server}}, \mathsf{User}(j)\rangle}(n) \colon \mathsf{D}(v) = 1\right]\right| \geq \frac{1}{p(n)}, \tag{39}$$

for some polynomial $p(n)$. We prove that this holds for independently and uniformly chosen $i, j \in [n]$ (and therefore there exist i and j for which this holds) where $\widetilde{\mathsf{Server}}$ is the malicious server described above, and $\mathsf{D} = \mathsf{D}_{i,j}$ is a distinguisher that uses D' as follows:

- If D' outputs i, then D outputs 1.
- If D' outputs j, then D outputs 0.
- Otherwise, D outputs a uniformly distributed $b \in \{0, 1\}$.

It follows that

$$\begin{aligned} &\Pr\left[v \leftarrow \mathsf{view}_{\langle \widetilde{\mathsf{Server}}, \mathsf{User}(i)\rangle}(n) \colon \mathsf{D}(v) = 1\right] \\ &\quad = \Pr\left[v \leftarrow \mathsf{view}_{\langle \widetilde{\mathsf{Server}}, \mathsf{User}(i)\rangle}(n) \colon \mathsf{D}'(v) = i\right] \\ &\qquad + \frac{1}{2} \cdot \Pr\left[v \leftarrow \mathsf{view}_{\langle \widetilde{\mathsf{Server}}, \mathsf{User}(i)\rangle}(n) \colon \mathsf{D}'(v) \notin \{i, j\}\right] \\ &\quad \geq \frac{1}{n} + \frac{1}{n^2} + \frac{1}{2} \cdot \Pr\left[v \leftarrow \mathsf{view}_{\langle \widetilde{\mathsf{Server}}, \mathsf{User}(i)\rangle}(n) \colon \mathsf{D}'(v) \notin \{i, j\}\right], \end{aligned} \tag{40}$$

and

$$\begin{aligned}
&\Pr\left[v \leftarrow \mathsf{view}_{\widetilde{\langle \mathsf{Server}, \mathsf{User}(j)\rangle}}(n)\colon \mathsf{D}(v) = 1\right] \qquad (41)\\
&\quad = \Pr\left[v \leftarrow \mathsf{view}_{\widetilde{\langle \mathsf{Server}, \mathsf{User}(j)\rangle}}(n)\colon \mathsf{D}'(v) = i\right]\\
&\qquad + \frac{1}{2} \cdot \Pr\left[v \leftarrow \mathsf{view}_{\widetilde{\langle \mathsf{Server}, \mathsf{User}(j)\rangle}}(n)\colon \mathsf{D}'(v) \notin \{i, j\}\right]\\
&\quad = \frac{1}{n} + \frac{1}{2} \cdot \Pr\left[v \leftarrow \mathsf{view}_{\widetilde{\langle \mathsf{Server}, \mathsf{User}(j)\rangle}}(n)\colon \mathsf{D}'(v) \notin \{i, j\}\right],
\end{aligned}$$

where the last equality holds since both i and j are independently chosen. Finally, note that

$$\Pr\left[v \leftarrow \mathsf{view}_{\widetilde{\langle \mathsf{Server}, \mathsf{User}(i)\rangle}}(n)\colon \mathsf{D}'(v) \notin \{i, j\}\right] = \Pr\left[v \leftarrow \mathsf{view}_{\widetilde{\langle \mathsf{Server}, \mathsf{User}(j)\rangle}}(n)\colon \mathsf{D}'(v) \notin \{i, j\}\right],$$

and we conclude that

$$\left|\Pr\left[v \leftarrow \mathsf{view}_{\widetilde{\langle \mathsf{Server}, \mathsf{User}(i)\rangle}}(n)\colon \mathsf{D}(v) = 1\right] - \Pr\left[v \leftarrow \mathsf{view}_{\widetilde{\langle \mathsf{Server}, \mathsf{User}(j)\rangle}}(n)\colon \mathsf{D}(v) = 1\right]\right| \geq \frac{1}{n^2}.$$

□