On Hardness Amplification of One-Way Functions

Henry Lin*, Luca Trevisan**, and Hoeteck Wee***

Computer Science Division
UC Berkeley

Abstract. We continue the study of the efficiency of black-box reductions in cryptography. We focus on the question of constructing strong one-way functions (respectively, permutations) from weak one-way functions (respectively, permutations). To make our impossibility results stronger, we focus on the weakest type of constructions: those that start from a weak one-way permutation and define a strong one-way function. We show that for every "fully black-box" construction of a $\epsilon(n)$-secure function based on a $(1-\delta(n))$-secure permutation, if $q(n)$ is the number of oracle queries used in the construction and $\ell(n)$ is the input length of the new function, then we have $q \geq \Omega(\frac{1}{\delta} \cdot \log \frac{1}{\epsilon})$ and $\ell \geq n + \Omega(\log 1/\epsilon) - O(\log q)$. This result is proved by showing that fully black-box reductions of strong to weak one-way functions imply the existence of "hitters" and then by applying known lower bounds for hitters. We also show a sort of reverse connection, and we revisit the construction of Goldreich et al. (FOCS 1990) in terms of this reverse connection.

Finally, we prove that any "weakly black-box" construction with parameters $q(n)$ and $\ell(n)$ better than the above lower bounds implies the unconditional existence of strong one-way functions (and, therefore, the existence of a weakly black-box construction with $q(n) = 0$). This result, like the one for fully black-box reductions, is proved by reasoning about the function defined by such a construction when using the identity permutation as an oracle.

1 Introduction

We continue the study of efficiency of reductions in cryptography, and we focus on the question of constructing strong one-way functions or permutations from weak one-way functions or permutations.

1.1 Efficiency of Cryptographic Reductions

Several fundamental results in the foundations of cryptography, most notably the proof that pseudorandom generators exist if one-way functions exist [HILL99],

* `henrylin@cs.berkeley.edu`.
** `luca@cs.berkeley.edu`. Work supported by US-Israel BSF Grant 2002246.
*** `hoeteck@cs.berkeley.edu`. Work supported by US-Israel BSF Grant 2002246.

are proved via constructions and reductions that are too inefficient to be used in practice. It is natural to ask whether such inefficiency is a necessary consequence of the proof techniques that are commonly used, namely "black-box" constructions and reductions.

The first proof of a lower bound to the efficiency of a reduction for constructing a cryptographic primitive from another was by Kim, Simon and Tetali [KST99], in the context of constructing one-way hash functions from one-way permutations. Later work by Gennaro and Trevisan [GT00] and by Gennaro, Gertner and Katz [GGK03] has focused on constructions of pseudorandom generators from one-way permutations and of signature schemes and encryption schemes from trapdoor permutations.

The study of limitations of black-box reductions was initiated by Impagliazzo and Rudich [IR89], who showed that key agreement and public-key encryption cannot be based on one-way functions or one-way permutations using black-box reductions. Several other impossibility results for black-box reductions are known, including a result of Rudich [Rud88] and Khan, Saks, and Smyth [KSS00] ruling out constructions of one-way permutations based on one-way functions, a result of Rudich [Rud91] ruling out round-reduction procedures in public key encryption, and results of Gertner et al. [GKM$^+$00] and Gertner, Malkin and Reingold [GMR01] giving a hierarchy of assumptions in public key encryption that cannot be proved equivalent using black-box reductions.

1.2 Black-box Constructions

To illustrate the definition of a black-box construction, consider for example the notion of black-box construction of a key agreement protocol based on one-way functions formalized in [IR89]. In this model, the one-way function $f()$ is given as an oracle, and the protocols A and B for Alice and Bob are oracle procedures with access to $f()$. This, for example, means that a protocol where the code (or circuit) of $f()$ is used in the interaction is not black-box as defined above. The security of the protocol is also defined in a black-box way as follows: we assume that there is a security reduction R (a probabilistic polynomial time oracle algorithm) such that if E is a procedure for Eve (of arbitrary complexity) that breaks (A^f, B^f), then $R^{E,f}$ inverts f on a noticeable fraction of inputs. Notice that a proof of security in which the code (or circuit) of the adversary E is used in the reduction would not fit the above model. This model, in which both the "one-way function" $f()$ and the adversary E are allowed to be of arbitrary complexity, is called the *fully black-box* model in [RTV04]. In all the above cited papers [IR89,Rud88,Rud91,KST99,KSS00,GT00,GKM$^+$00,GMR01,GGK03], as well as in the results of this paper, fully black-box reductions are ruled out unconditionally.

A less restrictive model of black-box construction, introduced in [GT00] and formally defined in [RTV04], is the *weak black-box* model. As before, in a weakly black-box construction of key agreement from one-way functions, the algorithms for Alice and Bob are oracle algorithms that are given access to a function $f()$. A proof of security, however, only states that if $f()$ is hard to invert for efficient procedures that are given oracle access to $f()$, then the protocol is secure in the standard sense (that is, for adversaries that are ordinary probabilistic polynomial time algorithms with no oracles). In this model one is still not allowed to use the code of $f()$ in the construction or in the security analysis. The code of the adversary, however, may be used in the security analysis. Note that if we have a provably secure construction of, say, a key agreement protocol, then we also have a weakly black-box construction of key agreement based on one-way functions: just make the algorithms for Alice and Bob be oracle algorithms that never use the oracle. For this reason, one cannot unconditionally rule out weakly black-box constructions: at most, one can show that a weakly black-box construction implies the unconditional existence of some cryptographic primitive and possibly complexity theoretic separations that we do not know how to prove. Negative results for weakly black-box constructions are proved in [GT00] and [GGK03], where the authors show that weakly black-box constructions that make a small number of oracle queries imply the existence of one-way functions. The other lower bounds cited above [IR89,Rud88,Rud91,KST99,KSS00,GKM$^+$00,GMR01], however, do not rule out weakly black-box reductions.

The reason for this lack of negative results about weakly black-box reductions is partly explained in [RTV04]. For example, Reingold et al. [RTV04] prove that, unless one-way functions exist and key agreement is impossible in the real world, then there is a weak black-box construction of key agreement based on one-way functions. In other words, from the existence of a weakly black-box construction of key agreement from one-way functions it is impossible to derive any other consequence besides the obvious one that the existence of one-way functions implies the existence of key agreement schemes. See [RTV04] for a precise statement of this result and for a discussion of its interpretation.

In this paper, we are able to prove unconditional lower bounds for fully black-box reductions, and to show that a weakly black-box reduction improving on our lower bounds implies the unconditional existence of one-way functions.

1.3 Amplification of Hardness

We say that a function $f()$ is $\alpha(n)$-secure[1] if for every family of polynomial-size oracle circuits $\{C_n\}$ and for all n, the probability that $C_n^f(f(x))$ outputs a preimage of $f(x)$ is at most $\alpha(n)$, where the probability is taken over the uniform choice of x from $\{0,1\}^n$. We say that a function $f()$ is a *strong one-way function* if it is computable in polynomial time and is also $\epsilon(n)$-secure, for $\epsilon(n) = n^{-\omega(1)}$. We say that $f()$ is a *weak one-way function* if it is computable in polynomial time and is also $(1-\delta(n))$-secure, for $\delta(n) = n^{-O(1)}$.

The problem of "amplification of hardness" is to deduce the existence of strong one-way functions (respectively, permutations) from the existence of weak one-way functions (respectively, permutations).

[1] We avoid definitions and statements in terms of concrete security as they do not directly apply to adversaries of arbitrary complexity in fully black-box reductions. The results in Section 5 may be restated with concrete security parameters in a straight-forward manner.

The "direct product" construction is a simple approach to prove amplification of hardness results. Given a weak one-way function $f()$ we define a new function $f'()$ as $f'(x_1, x_2, \ldots, x_{q(n)}) = (f(x_1), f(x_2), \ldots, f(x_{q(n)}))$, where $q(n)$ is a polynomial. The function f' is still computable in polynomial time, and a non-trivial analysis shows that if f is weak one-way then f' is strong one-way.[2] Furthermore, if $f()$ is a permutation then $f'()$ is a permutation. See for example [Gol01, Sec 2] for more details.

The direct product construction is not, however, "security-preserving," in that the input length of the new function is polynomially larger than the input length of the original function. (In a security-preserving construction, the input length of the new function would be linear in the input length of the original one.) See for example [Lub96] for a discussion of "security preserving" reductions and the importance, in a cryptographic reduction, of not increasing the input length of the new primitive by too much.

For one-way permutations, we do have a security-preserving construction due to Goldreich et al. [GIL$^+$90] based on random walks on expanders. We stress that our results do not rule out fully black-box security-preserving hardness amplification for one-way functions, and we hope that the connections presented in this paper will help resolve this open problem.

1.4 Our Results

We say that a polynomial time computable oracle function $F^{(\cdot)}$ is a fully black-box construction of $\epsilon(n)$-secure functions from $(1-\delta(n))$-secure functions if there is a probabilistic polynomial time oracle algorithm $R^{(\cdot,\cdot)}$ such that for every function $f : \{0,1\}^n \to \{0,1\}^n$ and adversary $A()$ with the property that $A()$ inverts F^f on a $\geq \epsilon(n)$ fraction of inputs, then $R^{A,f}()$ inverts $f()$ on a $\geq 1-\delta(n)$ fraction of inputs. From this definition it is immediate to see that if $f()$ is polynomial time computable and no polynomial time adversary can invert it on more than a $1-\delta(n)$ fraction of inputs, then it follows that F^f is polynomial time computable and no polynomial time adversary can invert it on more than a $\epsilon(n)$ fraction of inputs. The definition, however, requires the reduction R to transform an adversary $A()$ of arbitrary complexity that inverts $F^f()$ into an adversary that inverts $f()$ in polynomial time given oracle access to $A()$ and $f()$.

A polynomial time computable oracle function $F^{(\cdot)}$ is a weak black-box construction of $\epsilon(n)$-secure functions from $(1-\delta(n))$-secure functions if for every function $f : \{0,1\}^n \to \{0,1\}^n$ and polynomial time adversary $A()$ with the property that $A()$ inverts F^f on a $\geq \epsilon(n)$ fraction of inputs, then there is a polynomial time oracle adversary $R^{(\cdot)}$ such that R^f inverts $f()$ on a $\geq 1-\delta(n)$ fraction of inputs.

Impossibility of Fully Black-Box Constructions Our first main result is as follows.

[2] This approach is typically credited to [Yao82].

Theorem 1. *Let $F^{(\cdot)}$ be a fully black-box construction of $\epsilon(n)$-secure functions from $(1-\delta(n))$-secure permutations, let ℓ be the input length of $F^{()}$, n the length of inputs of the oracle function, and q be the number of oracle queries.*

Then $q \geq \Omega(\frac{1}{\delta}\log\frac{1}{\epsilon})$ and $\ell \geq n - O(\log q) + \Omega(\log\frac{1}{\epsilon})$.

In comparison, the direct product construction has $q = O(\frac{1}{\delta}\log\frac{1}{\epsilon})$, which is tight, but $\ell = nq$. The construction of [GIL+90], that only works if the oracle is a permutation, has $q = O(\frac{1}{\delta}\log\frac{1}{\epsilon})$, which is tight, and $\ell = O(n + \log\frac{1}{\epsilon})$ for $\delta = n^{-O(1)}$, which is nearly tight.

It should be noted that our result applies even to constructions that require the oracle to be a permutation and that do not guarantee the new function to be a permutation. In particular, it applies as a special case to constructions that map permutations into permutation and functions into functions.

To prove Theorem 1, we first show that a fully black-box reduction of strong to weak one-way functions or permutations implies the existence of a disperser, or a "hitter" in the terminology of [Gol97] with efficiency parameters that depend on the efficiency of the reduction. A hitter is a randomized algorithm that outputs a small number of strings in $\{0,1\}^n$ such that for every sufficiently dense subset of $\{0,1\}^n$, the output of the hitter hits the set (that is, at least one of these strings is contained in the set) with high probability. In a hitter, we would like to use a small number of random bits, to generate a small number of strings, and we would like the density of the sets to be low and the probability of hitting them to be high. Various impossibility results are known for hitters and, in particular, if ℓ is the number of random bits, q is the number of strings, δ is the density of the sets and $1-\epsilon$ is the hitting probability, then it is known that $q \geq \Omega(\frac{1}{\delta}\cdot\log\frac{1}{\epsilon})$ and $\ell \geq n - \log q + \Omega(\log\frac{1}{\epsilon})$. Our negative results for fully black-box constructions will follow from the connection between such constructions and hitters and from the above negative results for hitters.

The intuition for the connection is, with some imprecision, as follows: a function $f : \{0,1\}^n \to \{0,1\}^n$ may be $(1-\delta)$-secure and still be extremely easy to invert on a $1-\delta$ fraction of inputs, while only a subset H of density δ of inputs is very hard to invert. If the computation of $F^f(z)$ involves only oracle queries to $f()$ on inputs outside H, then $F^f()$ is not "using" the hardness of $f()$, and $F^f(z)$ will be "easy" to invert. Considering that at most an ϵ fraction of inputs of $F^f()$ can be easy to invert, it follows that, for at least a $1-\epsilon$ fraction of the choices of z, the oracle queries of the computation $F^f(z)$ hit the set H. In conclusion, using ℓ random bits (to choose z) we have constructed q strings in $\{0,1\}^n$ (the oracle queries in the computation $F^f(z)$) such that a set of density δ (the set H) is hit with probability at least $1-\epsilon$. Of course none of this is technically correct, but the above outline captures the main intuition.

Moving on to a more precise description of our proof, we show that if $F^{(\cdot)}$ is a fully black-box construction of ϵ-secure functions from $(1-\delta)$-secure ones, where ℓ is input length of F, n the input length of the oracle, and q the number of oracle queries, then we can derive a hitter that uses randomness ℓ, produces q strings, and has hitting probability $1-\sqrt{\epsilon}$ for sets of density 2δ. The hitter, given an ℓ-bit random string z, simply outputs the oracle queries in the computation of $F^{\mathsf{id}}(z)$, where id is the identity permutation.

Suppose that the construction is not a hitter as promised, then there is a set H of density 2δ such that $F^{\mathsf{id}}(z)$ avoids querying elements of H for a $\sqrt{\epsilon}$ fraction of the z. Let $\pi : \{0,1\}^n \to \{0,1\}^n$ be a permutation that is the identity on elements not in H and that is a random permutation on H. Then F^{id} and F^{π} agree on at least a $\sqrt{\epsilon}$ fraction of inputs. We can also show that if A is a uniform (possibly, exponential time) algorithm that inverts F^{id} everywhere then A inverts F^{π} on at least a fraction ϵ of the inputs. (We would not lose this quadratic factor if we insisted that F^{id} and F^{π} be permutations.) In fact, such an A exists as long as F is polynomial time computable. Now we have that $R^{A,\pi}$ is a uniform algorithm that inverts π on at least a $1-\delta$ fraction of inputs, using polynomial number of oracle queries into π. Restricting ourselves to H, we get that $R^{A,\pi}$ inverts π on at least $1/2$ of the elements of H, which is impossible because π is a random permutation over H, and it cannot be inverted on many inputs by a uniform procedure (regardless of running time) that makes a polynomial number of oracle queries [IR89,Imp96,GT00]. We have reached a contradiction, and so our construction was indeed a hitter as promised.

Impossibility of Weakly Black-Box Constructions For weakly black-box constructions, we show that improving beyond our lower bounds is possible only by constructing strong one-way functions from scratch.

Theorem 2. *Let $F^{(\cdot)}$ be a weakly black-box construction of $\epsilon(n)$-secure permutations from $(1-\delta(n))$-secure permutations, let ℓ be the input length of $F^{()}$, n the length of inputs of the oracle function, and q be the number of oracle queries.*

There are constants c_1, c_2, c_3 such that if $q \leq c_1 \frac{1}{\delta} \log \frac{1}{\epsilon}$ or $\ell \leq n - c_2 \log q + c_3 \log \frac{1}{\epsilon}$, then one-way permutations exist unconditionally and, in particular, F^{id} is a $(1-\epsilon(n))$-secure permutation.

The proof is similar to the one of Theorem 1. We define a hitter based on the computation of F^{id} as before. If q and ℓ are too small, then the hitter must fail, and there must be some set H of density 2δ that is avoided with probability at least 2ϵ. Then we define a permutation π that is random on H and the identity elsewhere, and we note that F^{id} and F^{π} have agreement at least 2ϵ. If there were a polynomial time algorithm that inverts F^{id} on a $1-\epsilon$ fraction of inputs, the same algorithm would invert F^{π} on a ϵ fraction of inputs. This would yield a polynomial time oracle algorithm that given oracle access to π, inverts π on a $1-\delta$ fraction of inputs, which is again a contradiction. Therefore F^{id} is a (weak) one-way permutation.

A Reverse Connection We also point out a reverse connection, namely that special types of hitters yield fully black-box amplification of hardness (F,R). Specifically, we require that the F satisfy two additional properties (apart from computing a hitter). Suppose F^f on input z queries f on $x_1, \ldots, x_q$. The first property tells us that inverting F^f on $F^f(z)$ is at least hard as inverting f on all of $f(x_1), \dots, f(x_q)$. Next, given a challenge $f(x)$, the second property allows us to sample a challenge $F^f(z)$ (with the appropriate distribution) by substituting $f(x)$ for one of $f(x_1), \dots, f(x_q)$. We may view both hardness amplification via direct product and via random walks on expanders [GIL+90] in this framework, which yields a more modular and arguably simpler presentation of both results.

1.5 Perspective

The new connection between fully black-box hardness amplification and hitters makes explicit the construction of hitters in previous results on hardness amplification (namely a hitter from independent sampling in amplification via direct product and from random walks on expanders in [GIL+90]) and shows that such a construction is in fact necessary. In addition, we see from [GIL+90] in order to address the major open problem in this area of research - whether we can achieve security-preserving hardness amplification for one-way functions, it would be sufficient to give a hardness amplification procedure based on (δ, ϵ)-hitters with randomness complexity $O(n + 1/\delta \log 1/\epsilon)$ (which is optimal up to constant factors for constant δ but not sub-constant δ). There are simple and direct constructions of hitters achieving such parameters, and reviewing these constructions may prove to be a fruitful starting point for resolving this open problem.

2 Preliminaries

2.1 Notation

We use U_n to denote the uniform distribution over $\{0,1\}^n$. Given a function $G : \{0,1\}^m \to (\{0,1\}^n)^k$, $G_i : \{0,1\}^m \to \{0,1\}^n$, for $i = 1, 2, \dots, k$, is the function that on input z, outputs the i'th block of $G(z)$. In probability expressions that involve a probabilistic computation (of a probabilistic algorithm, say), the probability is also taken over the internal coin tosses of the underlying computation.

2.2 Notions of reducibility

Here, we are only interested in hardness amplification wherein the construction of the strong one-way function f' uses black-box access to a weak one-way permutation f. However, we distinguish between fully black-box and weakly black-box constructions, following the work of [RTV04], depending on whether the proof of security is black-box.

Definition 1 (Fully Black-Box Amplification of Hardness). *A fully black-box construction of $\epsilon(n)$-secure functions from $(1 - \delta(n))$-secure permutations is a pair of polynomial time computable oracle procedures F and R (where F is* *deterministic whereas R may be randomized) such that, for every permutation $f : \{0,1\}^n \to \{0,1\}^n$, F^f is a function mapping $\ell(n)$ bits into $\ell(n)$ bits, and for every function $A : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell(n)}$, if*

$$\Pr_{z \sim U_{\ell(n)}}[A(F^f(z)) = z' : F^f(z') = F^f(z)] \geq \epsilon(n)$$

then

$$\Pr_{x \sim U_n}[R^{A,f}(f(x)) = x] \geq 1 - \delta(n) \ .$$

By requiring that F and R be polynomial time computable, we guarantee that if f and A are polynomial time computable, then F^f and $R^{A,f}$ are also polynomial time computable. However, (F, R) must also satisfy the stated property even when given oracle access access to some function f and A that may not be polynomial time computable.

Definition 2 (Weakly Black-Box Amplification of Hardness). *A weakly black-box construction of $\epsilon(n)$-secure functions from $(1 - \delta(n))$-secure permutations is a (deterministic) polynomial time computable oracle procedure F such that, for every permutation $f : \{0,1\}^n \to \{0,1\}^n$, F^f is a function mapping $\ell(n)$ bits into $\ell(n)$ bits, and if there is a probabilistic polynomial time algorithm A such that*

$$\Pr_{z \sim U_{\ell(n)}}[A(F^f(z)) = z' : F^f(z') = F^f(z)] \geq \epsilon(n)$$

then there is a probabilistic polynomial time oracle algorithm I such that

$$\Pr_{x \sim U_n}[I^f(f(x)) = x] \geq 1 - \delta(n) \ .$$

Remark 1. In both definitions, the new function defined by the construction is $\epsilon(n)$-secure on inputs of length $N = \ell(n)$, and so, according to our definition, it would be more precise to call it a $\epsilon(\ell^{(-1)}(N))$-secure function.

2.3 Hitters

Definition 3 (Hitter [Gol97]). *A function $G : \{0,1\}^m \to (\{0,1\}^n)^k$ is a* (δ, ϵ)-hitter *if all for sets $H \subseteq \{0,1\}^n$ of density at least δ,*

$$\Pr_{z \sim U_l}[\forall\, i = 1, 2, \ldots, k, G_i(z) \notin H] \leq \epsilon$$

We refer to m and k as the randomness complexity and sample complexity of G respectively.

An equivalent, and more common, notion is that of a *disperser*. Using a notation consistent with the one above, a function $D : \{0,1\}^m \times \{0,1\}^\kappa \to \{0,1\}^n$ is a (b, δ)-disperser if for every distribution X over $\{0,1\}^m$ of min-entropy at least b, and for every set $H \subseteq \{0,1\}^n$ of density at least δ, there is a non-zero probability that $D(X, U_\kappa)$ hits H. Such an object is easily seen to be equivalent to a $(2^{b-n}, \delta)$-hitter with $k = 2^\kappa$. We will use the hitter notation because it is more convenient for our purposes. The following lower bounds for hitters are proved in [Gol97,RTS97].

Theorem 3 (Lower Bounds for Hitters). *If $G : \{0,1\}^m \rightarrow (\{0,1\}^n)^k$ is a (δ, ϵ)-hitter, then:*

$$\begin{array}{lll} \textit{(sample complexity)} & k \geq \frac{1}{2\delta} \ln \frac{1}{2\epsilon} & \textit{provided } \epsilon \leq 1/8 \\ \textit{(randomness complexity)} & m > n - \log k + \log \frac{1}{\epsilon} + \log\log \frac{1}{\delta} & \end{array}$$

Efficient constructions of hitters are known that match these lower bounds up to constant factors.

Theorem 4. *[Gol97] There exists a polynomial time computable (δ, ϵ)-hitter with sample complexity $O(\frac{1}{\delta} \log \frac{1}{\epsilon})$ and randomness complexity $2n + O(\log \frac{1}{\epsilon})$.*

The construction of dispersers of Ta-Shma [TS98] give even tighter bounds.

2.4 Hardness of inverting random permutations

We begin by establishing that a permutation that is a random permutation on a subset of $\{0,1\}^n$ of density 2δ and is the identity everywhere else is $(1-\delta)$-secure. We will be using this permutation as a weak one-way function for establishing lower bounds for black-box hardness amplification.

Lemma 1. *Fix $T(n) = n^{\log n}$. For all sufficiently large n, for all $\delta > \frac{1}{T(n)}$, for all sets $H \subseteq \{0,1\}^n$ of density 2δ, there exists a permutation π_H on $\{0,1\}^n$ such that π_H is the identity on $\{0,1\}^n - H$, and for all oracle Turing machines M with description at most $\log n$ bits that makes at most $T(n)$ oracle queries,*

$$\Pr_{x \sim U_n}[M^{\pi_H}(\pi_H(x)) = x] < 1 - \delta$$

Proof. Let Π_H denote the set of permutations that is the identity on $\{0,1\}^n - H$. Fix an oracle Turing machine M. Then,

$$\mathrm{E}_{\pi \sim \Pi_H}[\#\{y \in H : M^{\pi}(y) = \pi^{(-1)}(y)\}] \leq 2\delta \cdot 2^n \left(\frac{n^{\log n} + 1}{2\delta \cdot 2^n - n^{\log n}} \right) < \frac{\delta}{4n} \cdot 2^n$$

Hence,

$$\Pr_{\pi \sim \Pi_H}[\#\{x \in H : M^{\pi}(\pi(x)) = x\} \geq \delta 2^n] \leq \frac{1}{4n}$$

This allows us to take a union bound over all oracle Turing machines M with description at most $\log n$ bits. □

Note that we could also derive a non-uniform analogue of this lemma using the counting argument of [GT00]:

Lemma 2. *Fix $T(n) = n^{\log n}$. For all sufficiently large n, for all $\delta > \frac{1}{T(n)}$, for all sets $H \subseteq \{0,1\}^n$ of density 2δ, there exists a permutation π_H on $\{0,1\}^n$ such that π_H is the identity on $\{0,1\}^n - H$, and for all probabilistic oracle Turing machines M with description at most $\log n$ bits that makes at most $T(n)$ oracle queries and uses at most $T(n)$ bits of non-uniformity and at most $T(n)$ random coin tosses,*

$$\Pr_{x \sim U_n}[M^{\pi_H}(\pi_H(x)) = x] < 1 - \delta$$

Remark 2. We stress that in both Lemma 1 and Lemma 2, we allow the machine M to have arbitrary (possibly exponential) running time, but we require that M has bounded non-uniformity, makes a bounded number of oracle queries and uses a bounded number of random coins; in this sense, M still has "low complexity".

3 Fully black-box hardness amplification

We use id to denote the identity function on $\{0,1\}^n$. If $F^{()}$ is an oracle procedure from ℓ bits to ℓ bits that makes at most q queries, we use $G^f : \{0,1\}^\ell \to (\{0,1\}^n)^q$ to denote the function computing the sequence of oracle queries (possibly adaptive) that F^f makes.

Theorem 1 follows from the following lemma and from the lower bounds for hitters of Lemma 3.

Lemma 3 (Fully BB versus Hitters). *Let (F, R) be a fully black-box construction of ϵ-secure functions f from $(1-\delta)$-secure permutations. Then, $G^{\mathsf{id}} : \{0,1\}^\ell \to (\{0,1\}^n)^q$ is a (polynomial time computable) $(2\delta, \sqrt{\epsilon})$-hitter.*

Before proving Lemma 3, we first prove a technical result that will be useful later. The point of the result is that if f and g have a noticeable agreement, and we are able to invert f in a strong sense (namely, uniformly sample pre-images), then we are also able to invert g on a noticeable fraction of inputs.

Lemma 4. *Let $f, g : \{0,1\}^n \to \{0,1\}^n$ be functions with agreement ϵ, and let $A()$ be a probabilistic procedure such that, for every $y \in \{0,1\}^n$, $A(y)$ outputs $\perp$ if $f^{(-1)}(y) = \emptyset$, and the output of $A(y)$ is uniform over $f^{(-1)}(y)$ otherwise. Then, the probability that $A(g(x)) \in g^{(-1)}(g(x))$ is at least ϵ^2, when taken over the uniform choice of $x \in \{0,1\}^n$ and over the internal coin tosses of A.*

Furthermore, if f, g are permutations with agreement ϵ and if A is such that $A(f(x)) = x$ for every x, then A inverts g on at least an ϵ fraction of inputs.

Proof. Given f, g with agreement ϵ, we define for each $s \in \{0,1\}^n$:

$$\sigma_s = \Pr_{x \sim U_n}[f(x) = s]$$
$$\tau_s = \Pr_{x \sim U_n}[f(x) = g(x) = s]$$

Clearly, $\sum_{s\in\{0,1\}^n} \sigma_s = 1$ and $\sum_{s\in\{0,1\}^n} \tau_s = \epsilon$. Observe that:

$$\begin{aligned}&\Pr_{x\sim U_n}[A(g(x)) \in g^{(-1)}(g(x))]\\ \geq &\Pr_{x\sim U_n}[f(x) = g(x) \text{ and } g(A(f(x))) = f(x)]\end{aligned}$$

We may rewrite the expression on the right-hand-side of the inequality as:

$$\sum_{s\in f(\{0,1\}^n)} \tau_s \cdot \frac{\tau_s}{\sigma_s} = \Big(\sum_{s\in f(\{0,1\}^n)} \frac{\tau_s^2}{\sigma_s}\Big)\Big(\sum_{s\in f(\{0,1\}^n)} \sigma_s\Big) \geq \Big(\sum_{s\in f(\{0,1\}^n)} \tau_s\Big)^2 = \epsilon^2$$

where the inequality follows from Cauchy-Schwartz. The case where f and g are permutations is trivial. □

We can now give the proof of Lemma 3.

Proof (Of Lemma 3). Suppose G^{id} is not a $(2\delta, \sqrt{\epsilon})$-hitter. Then, there exists a set $H \subseteq \{0,1\}^n$ of density 2δ such that

$$\Pr_{z\sim U_l}[\forall\, i = 1, 2, \ldots, q, G_i^{\mathsf{id}}(z) \notin H] > \sqrt{\epsilon}$$

Let A denote the uniform algorithm that inverts F^{id} everywhere using brute force; that is, A on input $F^{\mathsf{id}}(z) \in \{0,1\}^\ell$ computes F^{id} on all $z' \in \{0,1\}^\ell$, and outputs a randomly chosen z' such that $F^{\mathsf{id}}(z') = F^{\mathsf{id}}(z)$, and $\perp$ if no such z' exists. Let π_H denote the permutation guaranteed by Lemma 2.

Observe that for each $z \in \{0,1\}^\ell$ such that $G_i^{\mathsf{id}}(z) \notin H$ for all $i = 1, 2, \ldots, q$, it must be the case that $F^{\pi_H}(z) = F^{\mathsf{id}}(z)$, and, in particular, we have that F^{id} and F^{π_H} have agreement at least $\sqrt{\epsilon}$. From Lemma 4 we have that A inverts F^{π_H} with probability at least ϵ, and so R^{A,π_H} inverts π_H on a $1-\delta$ fraction of inputs. By incorporating A into R, we have a probabilistic oracle Turing machine M (with exponential running time) that given oracle access to just π_H makes at most a polynomial number of queries (and so less than $n^{\log n}$) into π_H, flips a polynomial number of random coins and inverts π_H on a $1-\delta$ fraction of inputs, a contradiction to Lemma 2. □

Remark 3. To be more precise, we should say, fix $\delta, \epsilon : \mathbb{N} \to (0, 1/2)$. Then, for all sufficiently large n, G^{id} is a $(2\delta, \sqrt{\epsilon})$-hitter.

4 Weakly black-box hardness amplification

Theorem 2, our negative result for weakly black-box constructions, follows from the result below and from Lemma 3, the negative results about hitters.

Lemma 5 (Weakly BB versus Hitters). *Suppose there exists a weakly black-box construction of a ϵ-secure permutation $F^{()} : \{0,1\}^\ell \to \{0,1\}^\ell$ from a $(1-\delta)$-secure permutation $f : \{0,1\}^n \to \{0,1\}^n$, which makes at most q queries to f. Then, one of the following is true:*

1. $G^{\mathsf{id}} : \{0,1\}^\ell \to (\{0,1\}^n)^q$ *is a polynomial time computable* $(2\delta, 2\epsilon)$*-hitter;*
2. F^{id} *is a* $(1-\epsilon)$*-secure permutation.*

Proof. Suppose neither statement is true. Then, there exists a set $H \subseteq \{0,1\}^n$ of density 2δ such that

$$\Pr_{z \sim U_l}[\forall\, i = 1, 2, \ldots, q, G_i^{\mathsf{id}}(z) \notin H] > 2\epsilon$$

In addition, there exists an efficient algorithm A that inverts F^{id} on a $1-\epsilon$ fraction of inputs. Again, let π_H denote the permutation guaranteed by Lemma 2. By the "furthermore" part of Lemma 4 we have $\Pr[F^{\mathsf{id}}(z) = F^{\pi_H}(z)] > 2\epsilon$, so it follows that

$$\Pr_{z \sim U_\ell}[A(F^{\pi_H}(z)) = z] > \epsilon$$

By the weakly black-box property of $F^{()}$, there exists an efficient oracle algorithm B such that

$$\Pr_{x \sim U_n}[B^{\pi_H}(\pi_H(x)) = x] \geq 1 - \delta$$

a contradiction. □

5 Revisiting the Direct Product Construction and [GIL+90]

We present a simple, modular and unified view of the analysis for previous results for fully black-box hardness amplification. We stress that the analysis is not novel, and is based largely on the exposition of [GIL+90] in [Gol01, Sec 2.6].

Theorem 5. *Let F be a (deterministic) polynomial time computable oracle procedure such that for every function (resp permutation) $f : \{0,1\}^n \to \{0,1\}^n$, F^f is a function mapping $\ell(n)$ bits to $\ell(n)$ bits and makes at most $q(n)$ oracle queries. Let $G^{()} : \{0,1\}^\ell \to (\{0,1\}^n)^q$ be the function that computes the sequence of q oracle queries that F makes. Suppose F also satisfies the following properties for every function (resp permutation) $f : \{0,1\}^n \to \{0,1\}^n$:*

1. *(consistent) for any $z, z' \in \{0,1\}^\ell$ such that $F^f(z') = F^f(z)$, we have $f(G_i^f(z')) = f(G_i^f(z))$ for all i.*
2. *(restrictable) there exists a polynomial time oracle algorithm that given oracle access to f, and given input $i \in [k]$ and $y \in f(\{0,1\}^n)$, outputs a random sample from the distribution[3] $\{F^f(U_l) \mid f(G_i^f(U_l)) = y\}$.*
3. *(hitting) G^f is a $(\delta/2, \epsilon/2)$-hitter.*

[3] The distribution may be described more precisely by the following two-step experiment: pick z uniformly at random from $\{z \in \{0,1\}^\ell : f(G_i^f(z)) = y\}$ and output $F^f(z)$. We stress that the sampling algorithm may not compute z explicitly.

Then, there exists a probabilistic polynomial time oracle procedure R such that (F, R) constitute a fully black-box construction of a ϵ-secure function from a $(1-\delta)$-secure function (resp permutation). In addition, $R^{A,f}$ makes $O(\frac{q^2}{\epsilon}\log\frac{1}{\delta})$ oracle queries to A.

Consider what happens in a black-box reduction for a proof of security of hardness amplification. We are given oracle access to an algorithm A that inverts F^f on a ϵ fraction of input and a challenge $f(x)$. The "consistent" property tells us (informally) that inverting F^f on $F^f(z)$ is at least as hard as inverting f on all of the $f(G_i^f(z))$'s ($i = 1, 2, \ldots, q$), and the "restrictable" property allows us to construct from $f(x)$ a challenge $F^f(z)$ for A by substituting $f(x)$ for one of the $f(G_i^f(z))$'s. Note that the "consistent" property is trivially satisfied if F^f is injective. We also do not need to make any assumptions about the distributions $G_i^f(U_\ell)$, $i = 1, 2, \ldots, q$.

Proof. Let A be a function that inverts F^f on an ϵ fraction of input. Now, consider an oracle procedure I that given oracle access to A, f and on input $y \in \{0,1\}^n$, does the following: for each $i = 1, 2, \ldots, q$,

1. samples $y^{(i)}$ from $\{F^f(U_l) \mid f(G_i^f(U_l)) = y\}$, and computes $z^{(i)} = A(y^{(i)})$;
2. checks whether $f(G_i^f(z^{(i)})) = y$, and if so, outputs $G_i^f(z^{(i)})$.

Define the set H (for "hard") by:

$$H = \{x \in \{0,1\}^n \mid \Pr[I^{A,f}(f(x)) \in f^{(-1)}(f(x))] < \epsilon/2q\}$$

It is easy to see that $x \in H$ iff $f(x) \in f(H)$.

Claim. $|H| < \delta/2 \cdot 2^n$

Proof. (of claim) Suppose otherwise. Then,

$$\begin{aligned}
&\Pr_{z\sim U_\ell}[A \text{ inverts } F^f(z)] \\
&\le \Pr_{z\sim U_\ell}[\forall i, G_i^f(z) \notin H] + \sum_{i=1}^{q} \Pr_{z\sim U_\ell}[A \text{ inverts } F^f(z) \text{ and } G_i^f(z) \in H] \\
&\le \epsilon/2 + \sum_{i=1}^{q} \Pr_{z\sim U_\ell}[A \text{ inverts } F^f(z) \text{ and } f(G_i^f(z)) \in f(H)] \quad \text{(by "hitting")} \\
&\le \epsilon/2 + \sum_{i=1}^{q} \max_{y\in f(H)} \Pr_{z\sim U_\ell}[A \text{ inverts } F^f(z) \mid f(G_i^f(z)) = y] \\
&\le \epsilon/2 + \sum_{i=1}^{q} \max_{y\in f(H)} \Pr_{z\sim U_\ell}[I^{A,f} \text{ inverts } y] \qquad \text{(by "consistent")} \\
&< \epsilon/2 + q \cdot \epsilon/2q \le \epsilon
\end{aligned}$$

a contradiction. ∎

Consider the oracle procedure R that given oracle access to A and f, runs $I^{A,f}$ $O(\frac{q}{\epsilon}\log\frac{1}{\delta})$ times. This allows us to amplify the success probability of inverting values not in H to $1-\delta/2$. Hence,

$$\begin{aligned}&\Pr_{x\in U_n}[R^{A,f}(f(x))\notin f^{(-1)}(f(x))]\\ &\le \Pr_{x\in U_n}[x\in H]+\Pr_{x\in U_n}[R^{A,f}(f(x))\notin f^{(-1)}(f(x))\mid x\notin H]<\delta\end{aligned}$$

The result follows. □

Next, we review previous results on hardness amplification in our framework:

Direct product. [Yao82] Here, we start with a $(1-\delta)$-secure function f, and we define $F^f : (\{0,1\}^n)^q \to (\{0,1\}^n)^q$ is given by $F(x_1,\ldots,x_q) = (f(x_1),\ldots,f(x_q))$, where $q = O(1/\delta\log 1/\epsilon)$. $G^f : \{0,1\}^{nq} \to (\{0,1\}^n)^q$ is then the identity function for all f. It is easy to check that F satisfies all of the 3 properties, from which hardness amplification via direct product follows.

Random walk on expanders. [GIL+90] Here, we start with a $(1-\delta)$-secure permutation π and a family of d-regular explicitly constructible expanders $\{\Gamma_n\}$ with vertex set $\{0,1\}^n$, where d is a constant. We define $G^\pi : \{0,1\}^n \times [d]^t \to (\{0,1\}^n)^{t+1}$ as follows:

$$\begin{aligned}&G_1^\pi(x,\sigma_1,\ldots,\sigma_t) = x\\ &G_{i+1}^\pi(x,\sigma_1,\ldots,\sigma_t) = g_{\sigma_i}(\pi(G_i^\pi(x,\sigma_1,\ldots,\sigma_t))) \quad i=1,2,\ldots,t\end{aligned}$$

where $g_\sigma(x)$ for $x\in\{0,1\}^n$ and $\sigma\in[d]$ denotes the σ'th neighbor of vertex x in Γ_n. Note that the output of G is the set of vertices visited in a random walk on G started at x along the path $\sigma_1,\ldots,\sigma_t$, interspersed with an application of π before each step. Since π is a permutation, applying π does not affect the mixing properties of the random walk, and therefore if we take $t = O(1/\delta\log 1/\epsilon)$, then G^π yields a $(\delta/2,\epsilon/2)$-hitter. The new function $F^\pi : \{0,1\}^n\times[d]^t \to \{0,1\}^n\times[d]^t$ is given by

$$F^\pi(x,\sigma_1,\ldots,\sigma_t) = (G_{t+1}^\pi(x,\sigma_1,\ldots,\sigma_t),\sigma_1,\ldots,\sigma_t)$$

It is easy to see that F^π is injective, and thus F is "consistent" and F^π is a permutation. The "restrictable" property is satisfied using the following algorithm: given $i\in[t+1]$ and $y\in\{0,1\}^n$, pick $\sigma_1,\ldots,\sigma_t$ independently at random from $[d]$, and output $(\pi(g_{\sigma_t}(\ldots g_{\sigma_i}(y)\ldots)),\sigma_1,\ldots,\sigma_t)$. This constitutes the basic building block: a fully black-box construction of ϵ-secure permutations from $(1-\delta)$-secure permutations with $\ell = n+O(1/\delta\log 1/\epsilon)$ and $q = O(1/\delta\log 1/\epsilon)$.

To obtain a security-preserving construction of an ϵ-secure permutation on $\{0,1\}^{O(cn+\log 1/\epsilon)}$ from a $(1-1/n^c)$-secure permutation on $\{0,1\}^n$, we compose the basic building block $c+1$ times as follows: we first construct a $(1-1/2n^{c-1})$-secure permutation, then a $(1-1/2n^{c-2})$-secure one, and right up to $1/2$-secure permutation. In the last composition, we construct a ϵ-secure permutation from a $1/2$-secure one.

6 Conclusion

Our negative result for weakly black-box constructions is less general than the one for fully black-box constructions: in the former case we restrict ourselves to constructions that define a permutation if the original primitive is a permutation. It should be noted that both the direct product construction and the construction of [GIL$^+$90] satisfy this property. It would be possible to strengthen Lemma 5 to hold under the assumption that $F^{()}$ is a construction of ϵ-secure *functions*, and with the conclusion that either G^{id} is a $(2\delta, 2\epsilon)$-hitter or that one-way functions exist uncondtionally. The proof would have followed along the lines of the proof of Lemma 3, using a result of Impagliazzo and Luby [IL89] to construct a polynomial time algorithm that approximates algorithm A in the proof of Lemma 3 assuming that one-way functions do not exist. We will give more details in the full version of this paper.

The main open problem that is still unresolved is whether there is a fully black-box security-preserving hardness amplification for one-way functions. From the work of [GIL$^+$90], we know that it would suffice to construct a "restrictable" and "consistent" hitter (see the statement of Theorem 5 for the terminology) with randomness complexity $O(n + 1/\delta \log 1/\epsilon)$.

References

[GGK03] Rosario Gennaro, Yael Gertner, and Jonathan Katz. Lower bounds on the efficiency of encryption and digital signature schemes. In *Proceedings of the 35th ACM Symposium on Theory of Computing*, pages 417–425, 2003.

[GIL$^+$90] Oded Goldreich, Russell Impagliazzo, Leonid Levin, Ramarathnam Venkatesan, and David Zuckerman. Security preserving amplification of hardness. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science*, pages 318–326, 1990.

[GKM$^+$00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 325–335, 2000.

[GMR01] Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 126–135, 2001.

[Gol97] Oded Goldreich. A sample of samplers - a computational perspective on sampling. Technical Report TR97-020, Electronic Colloquium on Computational Complexity, 1997.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1.* Cambridge University Press, 2001.

[GT00] Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 305–313, 2000.

[HILL99] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pages 230–235, 1989.

[Imp96] Russell Impagliazzo. Very strong one-way functions and pseudo-random generators exist relative to a random oracle. Unpublished manuscript, 1996.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21st ACM Symposium on Theory of Computing*, pages 44–61, 1989.

[KSS00] J. Kahn, M. Saks, and C. Smyth. A dual version of Reimer's inequality and a proof of rudich's conjecture. In *Proceedings of the 15th IEEE Conference on Computational Complexity*, 2000.

[KST99] J.H. Kim, D.R. Simon, and P. Tetali. Limits on the efficiency of one-way permutations-based hash functions. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pages 535–542, 1999.

[Lub96] M. Luby. *Pseudorandomness and Cryptographic Applications*. Princeton University Press, 1996.

[RTS97] J. Radhakrishnan and A. Ta-Shma. Tight bounds for depth-two superconcentrators. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 585–594, 1997.

[RTV04] Omer Reingold, Luca Trevisan, and Salil Vadhan. Notions of reducibility between cryptographic primitives. In *Proceedings of the 1st Theory of Cryptography Conference*, pages 1–20. LNCS 2951, 2004.

[Rud88] S. Rudich. *Limits on the provable consequences of one-way functions*. PhD thesis, University of California at Berkeley, 1988.

[Rud91] S. Rudich. The use of interaction in public cryptosystems. In *Proceedings of CRYPTO'91*, pages 242–251, 1991.

[TS98] A. Ta-Shma. Almost optimal dispersers. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, 1998.

[Yao82] A.C. Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23th IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.