Lower Bounds on the Efficiency of Generic Cryptographic Constructions

Rosario Gennaro
IBM T.J.Watson Research Center
rosario@watson.ibm.com

Luca Trevisan
UC Berkeley
luca@cs.berkeley.edu

Abstract

We present lower bounds on the efficiency of constructions for Pseudo-Random Generators (PRGs) and Universal One-Way Hash Functions (UOWHFs) based on black-box access to one-way permutations. Our lower bounds are tight as they match the efficiency of known constructions.

A PRG (resp. UOWHF) construction based on black-box access is a machine that is given oracle access to a permutation. Whenever the permutation is hard to invert, the construction is hard to break. In this paper we give lower bounds on the number of invocations to the oracle by the construction.

If S is the assumed security of the oracle permutation π (i.e. no adversary of size S can invert π on a fraction larger than $1/S$ of its inputs) then a PRG (resp. UOWHF) construction that stretches (resp. compresses) its input by k bits must query π in $q = \Omega(k/\log S)$ points. This matches known constructions.

Our results are given in an extension of the Impagliazzo-Rudich model. That is, we prove that a proof of the existence of PRG (resp. UOWHF) black-box constructions that beat our lower bound would imply a proof of the unconditional existence of such construction (which would also imply $P \neq NP$).

1 Introduction

Since the seminal paper by Diffie and Hellman [DH78] modern cryptography has been based on the concept of *one-way functions*. Informally a function $f : A \to B$ is one-way if given $y = f(x)$ for x chosen at random in A it is hard to compute any preimage of y. We do not know if one-way functions exists (their existence would imply that $P \neq NP$) but there are some candidate functions based on number-theoretic problems (like factoring and the discrete logarithm) which are widely believed to be one-way.

In the twenty-plus years since [DH78], one major direction of research in cryptography has been to try to construct cryptographic primitives based on the weakest possible computational assumption. Under the existence of one-way functions we know how to prove the existence of universal one-way hash functions and digital signatures [NY89, Rom90], pseudo-random generators [HILL99], pseudo-random function ensembles [GGM86, HILL99] and commitment schemes [Nao91, HILL99].

These constructions are very important from a theoretical point of view because they are based on the minimal complexity assumption required to have cryptography. On the other hand, however, their practical impact is very limited because of their inefficiency. In practice, constructions based on stronger assumptions (such as the hardness of a specific number-theoretic problem) might be much more attractive from an efficiency point of view.

This trade-off between the efficiency of a cryptographic construction and the strength of the complexity assumption on which it relies is one of the most interesting features in modern cryptographic research. Attempts of improving the efficiency of known constructions based on general assumptions have mostly failed. It is thus an interesting question to ask: *How efficient can cryptographic constructions be when based on general assumptions?*

In this paper we focus on constructions for universal one-way hash functions and pseudo-random bit generation (UOWHF and PRG for short in the following). For these primitives we provide lower-bounds on the efficiency of general constructions that match the efficiency of known schemes. Our lower bounds are expressed in the number of required invocations of a one-way *permutation* (since one-way permutations are *a fortiori* also one-way functions our lower bounds are stronger and clearly hold for functions as well).

1.1 Our Results

Informally, we say that a one-way permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ has security S if any circuit A of size smaller than S inverts π with probability less than $1/S$ (for concreteness, one can think of S as a slightly super-polynomial function of n, such as $n^{\log n}$, but our results hold for any choice of S). For an integer l, we also denote by U_l

the uniform distribution over $\{0,1\}^l$.

PSEUDO-RANDOM GENERATORS. A PRG is a deterministic length-increasing function $G : \{0,1\}^m \to \{0,1\}^{m+k}$ such that $G(U_m)$ is computationally indistinguishable from U_{m+k}. PRG's were introduced by Blum and Micali [BM84] and Yao [Yao82]. They proved that PRG's can be constructed based on one-way permutations. This construction, using a later improvement by Goldreich and Levin [GL89], requires $O(k/\log S)$ invocations (see e.g. [Gol95, Section 2.5.3] for more details), which is the best known bound for generic constructions.

We prove that this is essentially the best that can be achieved. That is, we prove that any construction of PRG's that stretches its input by k bits and is limited to black-box access to a one-way permutation π with security S must invoke it $\Omega(k/\log S)$ times.

UNIVERSAL ONE-WAY HASH FUNCTIONS. A UOWHF is a family of length-decreasing functions such that for any input x it is hard to find a *collision* with x for a function chosen randomly from the family. UOWHF's were introduced by Naor and Yung in [NY89] where they showed that they are sufficient to construct digital signature algorithm. In [NY89] it is shown how to construct UOWHF's from any one-way permutation. Later this was improved by Rompel in [Rom90] to one-way functions.

Regarding efficiency, the constructions in [NY89, Rom90] require at least one invocation of the one-way permutation/function for every bit of length decrease. That is, if we have a one-way permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ and we want to build a UOWHF family $\{h\}$ where each $h : \{0,1\}^{m+k} \to \{0,1\}^m$ then the construction requires k invocation to π. This can be easily improved to $O(k/\log S)$ invocations.

Here too, we prove that this is essentially the best that can be done. That is, we prove that any construction of UOWHF's that compresses its input by k bits and is limited to black-box access to a one-way permutation π with security S must invoke it $\Omega(k/\log S)$ times.

DISCUSSION AND REMAINING OPEN PROBLEMS. Our results indicate that assuming the mere existence of one-way functions, or even permutations, is too weak of a computational hypothesis to obtain *efficient* cryptographic primitives. As it will be evident from our proof techniques, the limitation stems from the fact that a permutation π may still be one-way with security S even if it hides only very few, say $O(\log S)$, bits of its input (actually we use such "pathological" functions to prove our lower bounds).

Thus when designing new schemes with an eye out to efficiency, it is important to use stronger computational assumptions that provide us with many more "secure" bits for each invocation of the one-way function. An interesting research direction would be to try to find the most general assumption which still allows for efficient schemes.

For the case of PRG's, for example, we know that we can have efficient constructions if we assume the existence of a one-way function with $\Omega(n)$ hard-core bits (an example of such a function can be found in [HSS93]). Similarly, we know that semantically secure encryption can be implemented efficiently with a trapdoor permutation that hides many bits (none of the "classic" trapdoor permutations has this property, although recently [CG00] present some candidates based on non-standard number theoretic problems). But if we look at encryption schemes secure against active attacks, we only know how to construct an efficient scheme based on a specific number-theoretic assumption (the Cramer-Shoup scheme [CS98] which is based on the so-called Decisional Diffie-Hellman assumption).

An interesting question is to try to come up with an efficient encryption scheme secure against active attacks based on a "generic" assumption on trapdoor functions (say a trapdoor permutation that hides $\Omega(n)$ bits). Another interesting question would be to determine a lower bound on the number of invocations to any trapdoor permutation in order to achieve even simple semantic security.

1.2 Overview of our techniques

We prove our results in an extension of the model of Impagliazzo and Rudich [IR89]. Informally (see Section 2 for a more detailed discussion on the models) Impagliazzo and Rudich proved that a construction of secure key exchange based solely on one-way functions must "contain" a proof that $P \neq NP$.

Similarly, we show that a secure construction of PRG (or UOWHF) that makes less than a certain number of queries to a one-way permutation black box, must contain a proof that $P \neq NP$. In fact we prove an even stronger consequence: if a secure construction of PRG (resp. UOWHF) makes less than the required number of queries, then PRG (resp. UOWHF) *exists unconditionally*, i.e., can be constructed without accessing a one-way function or permutation.

The proof hinges on a technical lemma stating that a random permutation mapping t bits into t bits is, with high probability, one-way with security $2^{-\Omega(t)}$, even against non-uniform adversaries. For the related case of random functions, such a result has been proved by Impagliazzo and Rudich [IR89] for the (much simpler) uniform case, and by Impagliazzo [Imp96] in the non-uniform case.[1]

Then we start from a secure construction G of a PRG (the case of UOWHF is similar although technically more complicated) with oracle access to a one-way permutation. We

[1] One could derive our result from Impagliazzo's proof and from the fact that a random function is indistinguishable from a random permutation. Anyway, our proof is quite different from Impagliazzo's, and a bit simpler.

run G with an oracle permutation that leaves $n - O(\log S)$ of its input bits unchanged, and it is a random permutation on the remaining $O(\log S)$ bits. According to the above lemma, a permutation chosen according to this distribution is, with high probability, one-way with security S, and thus G is also secure.

We show that if the number of queries q to this oracle is "small" (i.e. less than $k/O(\log S)$, where k is the stretch of the generator G), then we can construct a different PRG G', that takes as input the original seed s and q (distinct) random points in $\{0,1\}^{O(\log S)}$ and simulates G by using the q points to answer G's queries to the oracle. G' is a generator, because if q is "small", then its input is shorter than its output. G' is secure because its output is the same as G and thus indistinguishable from random.

Notice that G' is unconditional, i.e., it does not need to access any one-way permutation. Thus we prove that if G makes a small number of queries, then we have a proof of the unconditional existence of PRG's (a corollary of which is that we have a proof that $P \neq NP$).

1.3 Prior Related Work

This research was motivated and inspired by recent work of Kim, Simon and Tetali [KST99], who essentially initiated the study of efficiency limitations for cryptographic constructions.[2]

Our lower bounds on the complexity of UOWHF constructions improves on [KST99], where a lower bound of $\Omega(\sqrt{k}/\log S)$ on the number of invocations of a one-way permutation is proven. Our result is also qualitatively better, since it holds in a more general model (see Section 2 below for a discussion about the models in which such results can be stated). We do not know of any similar work for PRG's.

Previous negative result had focused more on impossibility results for the *security* of certain constructions rather than for their efficiency. Impagliazzo and Rudich in [IR89] give strong evidence that black-box access to one-way permutations cannot yield secure key exchange. In [Sim98], Simon proves that one-way permutations are not sufficient to construct collision-resistant hash functions (which is a stronger primitive than UOWHF's). Finally a very recent result [KSS00] shows that there is no construction of one-way permutations based on one-way functions.

[2] A somewhat different notion of efficiency was considered earlier by Rudich [Rud91], who proved that for every k, there exists an oracle relative to which secret key exchange can be done in k rounds but not in $k-1$ rounds.

2 The Models

2.1 Impagliazzo-Rudich, Black Boxes and Oracles

The fundamental paper about impossibility for cryptographic constructions is [IR89], and it is useful to start from there to motivate our definitions. The purpose of [IR89] was to prove that a certain kind of cryptographic construction was impossible. In this paper we are concerned with cryptographic constructions that are possible, and we are interested in their efficiency, but the difficulties in formalizing the question are similar.

More specifically, [IR89] was concerned with the question of whether key-exchange protocols based only on one-way functions exists. The difficulty in addressing this question is in the way of formalizing the notion of "being based on one-way functions." Intuitively, this should be formalized as the key exchange protocol being an oracle procedure that is given oracle access to a function. If the function is one-way then the protocol is secure. However if key exchange protocols exist, then there are key exchange protocols "based on one-way functions," that simply ignore the function given as an oracle (however, in order to prove the security of such a construction, one has to prove the possibility of key-exchange from scratch, which is beyond what we are able to prove with current techniques). So if one wants to prove that there is no key-exchange protocol based on one-way functions, one has to give a more restrictive definition, or to show that (as in the case above) the only way to make such a construction is by proving something that is beyond our current techniques.

The Impagliazzo-Rudich Approach. Impagliazzo and Rudich first restrict to "black-box" constructions that are secure whenever the function given as an oracle is hard to invert (even if it is also hard to compute, and so does not satisfy the definition of being one-way). Then, they assume that P=NP. Under these assumptions, they prove that when a random function is used as an oracle in any key exchange protocol, then the protocol can be broken, even though a random function is (with high probability) hard to invert. It then follows that a proof of security of a "black-box" construction of a key-exchange protocol based on one-way functions must also contain a proof that P $\neq$ NP, and so is beyond the reach of current techniques.

As we briefly mentioned in the introduction, we extend this model. But the proof methodology is basically identical. We show that a "black-box" construction of PRG which queries a random permutation oracle in too few places, may be transformed in a constructions that *never queries* the oracle at all. This yields the unconditional existence of PRG's,

which is a result beyond the reach of current techniques (since it also implies P $\neq$ NP).

Relativizations. In computational complexity theory there is a canonical way of showing that a certain result is seemingly beyond reach of current techniques, namely to show that the opposite result holds relative to an oracle. Impagliazzo and Rudich observe that their result can also be interpreted in this setting.

Comparison. The Impagliazzo-Rudich approach provides a black-box one-way function (in their case, a random function) and an adversary that breaks the construction when it uses the primitive. The adversary is implementable in polynomial time if P=NP. If one uses an oracle relative to which P=NP (e.g. an oracle for a PSPACE-complete problem), and one augments this oracle with the above black-box one-way function, one gets an oracle relative to which a one-way function exist, yet the construction can be broken. Typically, then, an impossibility result in the Impagliazzo-Rudich setting implies a relativized impossibility result, so that the Impagliazzo-Rudich setting is more general.

Among previous impossibility results, a very recent result showing that there is no construction of one-way permutations based on one-way functions [Rud88, KSS00] is in the Impagliazzo-Rudich model, while other results are based on relativizations [Rud91, Sim98, KST99].

2.2 Model for Pseudorandom Generator Constructions

We say that a permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ is (S,ϵ)-one way if for every circuit A of size $\leq S$ we have

$$\mathop{\mathbf{Pr}}_{x}[A(\pi(x)) = x] \leq \epsilon$$

To reduce the number of parameters, we will also say that a function is S-one way (or *one-way with security* S) if it is $(S, 1/S)$-one way.

We say that a random variable X_m ranging over $\{0,1\}^m$ is (S,ϵ)-indistinguishable from uniform if for every circuit T of size $\leq S$ we have

$$|\mathop{\mathbf{Pr}}_{x\in U_m}[T(x)=1] - \mathop{\mathbf{Pr}}_{x\in X_m}[T(x)=1]| \leq \epsilon$$

A pseudorandom generator construction is an oracle procedure $G^{(\cdot)} : \{0,1\}^m \to \{0,1\}^{m+k}$ that expects as an oracle a permutation $\pi : \{0,1\}^n \to \{0,1\}^n$. We are interested in constructions where G is computable in time polynomial in m, n, k and where the output of the generator is indistinguishable from uniform whenever the oracle permutation is a fixed one-way permutation, and the input of the generator is randomly generated.

In particular, we will say that G is a (S_p, S_g, ϵ) pseudorandom generator construction if for every permutation π that is S_p-one way we have that $G^\pi(U_m)$ is (S_g,ϵ)-indistinguishable from uniform.

The counterpositive is that G is a (S_p, S_g, ϵ)-generator if for every π such that there exists a circuit T of size $\leq S_g$ satisfying

$$|\mathop{\mathbf{Pr}}_{x}[T^\pi(x)=1] - \mathop{\mathbf{Pr}}_{s}[T^\pi(G^\pi(s))=1]| > \epsilon$$

there exists an oracle circuit A of size $\leq S_p$ satisfying

$$\mathop{\mathbf{Pr}}_{x}[A^\pi(\pi(x)) = x] > 1/S_p$$

In Section 4 we show that if there is a (S_p, S_g, ϵ) pseudorandom generator construction that uses less than $k/5\log S_p$ accesses into the permutation, then it is possible to construct unconditionally a polynomial-time computable pseudorandom generator whose output is (S_g,ϵ)-indistinguishable from uniform. This means that, in particular, P$\neq$NP (as in the Impagliazzo-Rudich setting).

It should be noted that our consequence is not only stronger than P$\neq$NP, but it is a "tight" consequence: it says that the only way to construct a pseudorandom generator based on a generic one-way permutation and that makes $o(k/\log S)$ accesses into the permutation is to prove unconditionally that a pseudorandom generator can be constructed, thus dispensing with the use of one-way permutations altogether. It is as Impagliazzo and Rudich had proved that if there is a key-agreement protocol that uses a one-way function, then there is an unconditionally secure key-agreement protocol.

2.3 Model for Universal One-Way Hash Function Constructions

Roughly speaking, a family of universal one way hash functions (abbreviated UOWHF) is a family $\mathcal{H}$ of functions having the same range and the same domain (the domain being smaller than the range) such that when we pick at random a function h from $\mathcal{H}$ and a point x from the domain, it is hard, given h and x, to find a point $x' \neq x$ such that $h(x) = h(x')$.

Formally, a family $\{h_s\}_{s\in\{0,1\}^r}$ of functions $h_s : \{0,1\}^{m+k} \to \{0,1\}^m$ is (S,ϵ)-universal one way hash (abbreviated $(S,\epsilon)-$UOWH) if for every circuit A of size $\leq S$ we have

$$\mathop{\mathbf{Pr}}_{s,x}[A(x,s,h_s(x)) = x' : x \neq x' \text{ and } h_s(x') = h_s(x)] \leq \epsilon$$

It will be convenient to represent such a family as a single function $H : \{0,1\}^r \times \{0,1\}^{m+k} \to \{0,1\}^m$ where $H(s,x) = h_s(x)$.

A construction of UOWHF from one-way permutations is an oracle procedure $H^{(\cdot)}(\cdot,\cdot)$ such that H expects as an oracle a permutation $\pi : \{0,1\}^n \to \{0,1\}^n$, and is given an input seed $s \in \{0,1\}^r$ and an input string $x \in \{0,1\}^{m+k}$. The output is $H^\pi(s,x) \in \{0,1\}^m$. Such a construction is (S_p, S_h, ϵ)-UOWH if for every π that is S_p-one way we have that $H^\pi(\cdot,\cdot)$ is (S_h,ϵ)-UOWH (even when the adversary is given oracle access to the permutation π). We show in section 5 that if there is a (S_p, S_h, ϵ)-UOWH construction that makes less than $k/5 \log S_p$ invocations to the permutation, then there exists, unconditionally, a polynomial time construction that is (S_h,ϵ)-UOWH. As a consequence, P$\neq$NP. Once more, our result falls in the Impagliazzo-Rudich setting, but it has a stronger, and "tight," consequence.

3 The Hardness of Inverting Random Permutations

In this section we prove that a random permutation is, with high probability, one-way with exponential security, even against non-uniform circuits.

Lemma 1 *Let A be an oracle procedure that makes at most q queries into the oracle. Let $\pi : [N] \to [N]$ such that*

$$\mathop{\mathbf{Pr}}_x[A^\pi(x) = \pi^{-1}(x)] \geq \epsilon$$

Then π can be described using $2(\log \binom{N}{a}) + \log((N-a)!)$ bits of information, given A, where $a = \epsilon N/(q+1)$.

PROOF: Consider the set I of ϵN points on which A is able to invert π, after making at most q queries into π. We want to to argue that there exists a subset $S \subseteq I$ such that $|S| \geq \epsilon N/(q+1)$ and such that the value of π^{-1} in all the points of S is totally determined once we are given A, the sets S and $\pi^{-1}(S)$ and the value of π^{-1} in all the points in $[N] - S$.

We define S by the following process. Initially S is empty, and all elements of I are candidates for being an element of S. We take the lexicographically first element x out of I, and we put it into S. We simulate the computation $A^\pi(x)$, and let us call $x_1, \ldots, x_q$ the queries made by A (we assume wlog that they are different), and $y_1, \ldots, y_q$ the answers (i.e. $y_i = \pi(x_i)$). If x is none of the answers, then we remove $y_1, \ldots, y_q$ from I. If x is one of the answers, say y_i, then we remove $y_1, \ldots, y_{i-1}$ from I. Then we take the lexicographically smallest of the remaining elements of I, we put it into S, etc. At any step of the construction of S, we add one element to S and we remove at most q elements from I. Since I has initially ϵN elements, in the end S has at least $\epsilon N/(q+1)$ elements.

(Note: a way to picture the previous argument is to draw a directed graph with [N] nodes, where there is an edge (x,y) if $A^\pi(x)$ makes a query x' such that $\pi(x') = y$. In this graph, every vertex has out-degree at most q. We mark all vertices corresponding to elements of I. We want to find a subset S of I such that the only edges among elements of S go from nodes of higher lex order to nodes of lower lex order. A greedy algorithm will find an S such that $|S| \geq |I|/q$.)

We now claim that given descriptions of the sets S and $\pi^{-1}(S)$, and given the values of π on $[N] - \pi^{-1}(S)$, and given A, it is possible to compute (or invert) π everywhere. It is enough to show that it is possible to invert π everywhere. The values of $\pi^{-1}(x)$ for $x \notin S$ are explicitly given. The values of $\pi^{-1}(x)$ for $x \in S$ can be reconstructed as follows (we should do the following reconstruction sequentially, for all $x \in S$ in lexicographic order). We simulate the computation of $A^\pi(x)$. By construction of S, $A^\pi(x)$ will make queries either in points not in $\pi^{-1}(S)$, or it will query $\pi^{-1}(x')$ where $x' \in S$ but x' precedes x in lexicographic order, or, otherwise, A is querying $\pi^{-1}(x)$ itself. In the first two cases, we have enough information to continue the simulation. In the last case, it means that the current query is $\pi^{-1}(x)$. In all possible cases, we have enough information to reconstruct $\pi^{-1}(x)$.

In order to describe S, $\pi^{-1}(S)$, and π restricted to $[N] - \pi^{-1}(S)$, we need $\log \binom{N}{a} + \log \binom{N}{a} + \log(N-a)!$ bits, where $a = |S|$. This completes the proof. □

As a consequence, we have

Theorem 2 *For sufficiently large t, if we pick at random a permutation $\pi : \{0,1\}^t \to \{0,1\}^t$, there is a probability at least $1 - 2^{-2^{t/2}}$ that the permutation is one-way with security $2^{t/5}$.*

We note that it would be possible to prove that with comparably high probability the permutation has security about $(2.34) \cdot 2^{t/4}$. The weaker expression $2^{t/5}$ is easier to use in our application, so we did not try to optimize.

PROOF: Let A be an oracle circuit of size $S = 2^{t/5}$. The circuit will not access the oracle permutation more than $q = 2^{t/5}$ times. Let us call $N = 2^t$. From Lemma 1, we have that the fraction of permutations $\pi : \{0,1\}^t \to \{0,1\}^t$ such that

$$\mathop{\mathbf{Pr}}_x[A^\pi(\pi(x)) = x] > 2^{-t/5}$$

is at most

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

where $a = 2^{-t/5}N/(q+1) = N/N^{1/5}(N^{1/5}+1) > N^{3/5}/2$. The above expression can be simplified to

$$\frac{\binom{N}{a}}{a!}$$

and using the inequalities $a! < (a/e)^a$ and $\binom{N}{a} > (eN/a)^a$, the expression is upper bounded by

$$\left(\frac{e^2N}{a^2}\right)^a < \left(\frac{4e^2}{N^{1/5}}\right)^a < 2^{-a} < 2^{-N^{3/5}/2}$$

for sufficiently large N.

There are at most $2^{St\log S} = 2^{\frac{1}{5}N^{1/5}(\log N)^2}$ oracle circuits of size $S = 2^{t/5}$, and so, by a union bound, the probability over the choice of a random π that there is one such circuit such that

$$\mathop{\mathbf{Pr}}_{x}[A^{\pi}(\pi(x)) = x] > 2^{-t/5}$$

is at most $2^{\frac{1}{5}N^{1/5}(\log N)^2 - N^{3/5}/2} < 2^{-N^{1/2}}$ for sufficiently large N. □

For each parameter $t < n$ we denote with $\Pi_{t,n}$ the following subset of the family of permutations over n-bits:

$$\begin{aligned}\pi \in \Pi_{t,n} \text{ iff}\\ \pi(a,b) = (\hat{\pi}(a), b) \text{ for some}\\ \hat{\pi} : \{0,1\}^t \to \{0,1\}^t\end{aligned}$$

A corollary to Theorem 2 is that if $t = 5\log S_p$, then for any $n > t$, $\pi \in_R \Pi_{t,n}$ is one-way with security S_p with very high probability.

Corollary 3 *For sufficiently large t, for any $n > t$, if we pick a random permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ in $\Pi_{t,n}$, then with probability bigger than $1 - 2^{-2^{t/2}}$, the permutation π is one-way with security $2^{t/5}$.*

4 Lower Bound for Pseudo-Random Generators

In this section we show our lower bound for PRG constructions. We start from such a PRG construction $G^{(\cdot)}(\cdot)$ that expects as an oracle a permutation $\pi : \{0,1\}^n \to \{0,1\}^n$. We assume that G stretches an m-bit seed into an $(m+k)$-bit output.

Suppose that G is a provable construction such that if π is S_p-hard then G is secure; we prove that unless G queries π in at least $\Omega(k/\log S_p)$ places, it is possible to derive from G an unconditional pseudorandom generator.

The basic idea of the proof is as follows. If G queries only few points in the oracle we can encode the answers in the seed of a new PRG, $G' : \{0,1\}^{m'} \to \{0,1\}^{m+k}$, which will be able to "simulate" a computation of G when it is fed with a random permutation oracle. Notice that G' does not use any oracle at all. We then use Theorem 2 to claim that a random permutation oracle is indeed hard to invert by a circuit of size S_p even when its range is $t = O(\log S_p)$. Thus with this oracle the outputs of G and consequently G' are indistinguishable from random. The desired bound comes from the fact that G' is still a "stretching" generator provided that m' is smaller than $m + k$. But $m' < qt + m$ since qt bounds the number of bits needed to encode the t-bits answers to G's q questions, plus one needs m bits to encode the original seed of G.

Theorem 4 *Let $G^{(\cdot)} : \{0,1\}^m \to \{0,1\}^{m+k}$ be a (S_p, S_g, ϵ) PRG construction that makes q queries into the permutation oracle and suppose $\epsilon > 2^{-S_p}$. If $q < k/5\log S_p$ then there is a $(S_g, 2\epsilon)$ PRG $G' : \{0,1\}^{m'} \to \{0,1\}^{m+k}$ (with $m' < m + k$) without access to a permutation oracle.*

PROOF: If $G^{(\cdot)}$ is a (S_p, S_g, ϵ) PRG, it means that if $\pi : \{0,1\}^n \to \{0,1\}^n$ is S_p-hard then, for any statistical test T of size $\leq S_g$ we have that

$$|\mathop{\mathbf{Pr}}_{x}[T(x) = 1] - \mathop{\mathbf{Pr}}_{s}[T(G^{\pi}(s)) = 1]| < \epsilon$$

Fix a test T of size S_g. Let $t = 5\log S_p$. From Corollary 3 we know that a random permutation $\pi \in_R \Pi_{t,n}$ is S_p-hard with probability larger than $1 - 2^{-2^{t/2}}$. Recall that π operates only on the first t input bits, i.e. π is defined as $\pi(a,b) = (\hat{\pi}(a), b)$ where $\hat{\pi}$ is a random permutation over $\{0,1\}^t$. Thus in other words

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\pi\in\Pi_{t,n}}[|\mathop{\mathbf{Pr}}_{x}[T(x) = 1] - \mathop{\mathbf{Pr}}_{s}[T(G^{\pi}(s)) = 1]| < \epsilon] >\\ > 1 - 2^{-2^{t/2}} > 1 - 2^{-S_p} > 1 - \epsilon\end{aligned}$$

By an averaging argument this yields that

$$|\mathop{\mathbf{Pr}}_{x}[T(x) = 1] - \mathop{\mathbf{Pr}}_{\pi\in\Pi_{t,n};s}[T(G^{\pi}(s)) = 1]| < 2\epsilon$$

Without loss of generality we may assume now that G always queries $\pi \in \Pi_{t,n}$ with strings that have distinct t-prefixes. Indeed notice that queries (a,b) to π are answered by $(\hat{\pi}(a), b)$. Thus for each "generic" G that asks arbitrary queries, one can construct a $\hat{G}$ with essentially the same running time as G, such that whenever G asks (a,b) where a was the prefix of a query asked before, it will "skip" the query to π and take as answer $(\hat{\pi}(a), b)$. In general the behavior of $\hat{G}$ is much different than G's, but when we restrict $\pi \in \Pi_{t,n}$ they are equivalent. So assume w.l.o.g. that G always make queries with distinct prefixes.

Now we use the fact that $G^{(\cdot)}$ queries its oracle only $q < k/t$ times. Consider the following PRG G'. It takes as input a seed s' of length $m' = \log\binom{2^t}{q} + s < qt + m < m + k$. It uses the first $\log\binom{2^t}{q}$ bits to select q distinct elements $y_1, \ldots, y_q$ in $\{0,1\}^t$ and then define

$$G'(y_1, \ldots, y_q, s) = G^{y_1,\ldots,y_q}(s)$$

where with the notation $G^{y_1,\ldots,y_q}(s)$ we mean the computation of G on input s and when its q oracle queries are answered using the t-prefixes $y_1,\ldots,y_q$. More precisely, if the i^{th} query by G is $x_i = (a_i, b_i)$ where $|a_i| = t$, then G' answers it with (y_i, b_i) (here is where we use the fact that w.l.o.g all the a_i's are distinct).

Clearly the distribution of $\{G'(s')\}_{s'}$ is identically distributed to $\{G^{\hat{\pi}}(s)\}_{\hat{\pi}\in P i_{t,n};s}$ thus:

$$|\mathop{\mathbf{Pr}}_{x}[T(x)=1] - \mathop{\mathbf{Pr}}_{s'}[T(G'(s'))=1]| < 2\epsilon$$

which ends the proof. □

5 Lower Bound for One Way Hash Functions

In this section we show our lower bound for UOWHF constructions. The proof outline is similar to the one for the case of PRG's. We start from a UOWHF construction $H^{(\cdot)}(\cdot,\cdot)$ that expects as an oracle a permutation $\pi : \{0,1\}^n \to \{0,1\}^n$. We assume that H takes as input a r-bit key and compresses $m+k$-bit inputs into m-bit outputs.

If the construction makes a number q of accesses into the permutation such that $q < k/5\log S$ (where S is the security of the permutation), then we show that it is possible to derive an unconditionally secure construction of UOWHF.

As in the case of PRG, we observe that H is unconditionally secure when the permutation π is chosen so that it randomly permutes the first $5\log S$ bits of the inputs, while leaving the remaining bits unchanged. We then show that this idealized setting can be realized by putting answers for the q queries into the key (one needs about $5q\log S$ bits to specify such answers, since only the relevant $5\log S$ bits of each answer have to be specified). Furthermore, we put in the *output* of our new hash function the q queries (or rather, the first $5\log S$ bits of each such query) done during the computation. If $q < k/5\log S$ we are still getting a length-decreasing function, and we are able to show that if an adversary can find collision in this new construction, then there is an adversary that finds collisions in the idealized setting, which we know is impossible. So we get an unconditionally secure construction of UOWHF.

Theorem 5 *Let $H^{(\cdot)} : \{0,1\}^r \times \{0,1\}^{m+k} \to \{0,1\}^m$ be a (S_p, S_h, ϵ) UOWHF construction that makes q queries into the permutation oracle, suppose $\epsilon > 2^{-S_p}$. If $q < k/5\log S_p$ then there is a $(S_h, 2\epsilon)$ UOWH $H' : \{0,1\}^{r'} \times \{0,1\}^{m+k} \to \{0,1\}^{m'}$ (with $m' < m+k$) without access to a permutation oracle.*

PROOF: If $H^{(\cdot)}(\cdot,\cdot)$ is (S_p, S_h, ϵ)-UOWH, it means that if $\pi : \{0,1\}^n \to \{0,1\}^n$ is S_p-hard then, for any collision finding adversary A of size S_h we have that

$$\mathop{\mathbf{Pr}}_{s,z}[A^\pi(z,s,H^\pi(s,z)) = z' : z \neq z' \text{ and } H^\pi(s,z) = H^\pi(s,z')] \leq \epsilon$$

Let us define $t = 5\log S_p$. For an element $x \in \{0,1\}^n$, we call the string made up by the first t bits of x the *t-prefix* of x.

We will now define a stronger notion of collision for H. We say that z' is a *strong collision* for z with seed s and permutation π if $z' \neq z$, $H^\pi(s,z) = H^\pi(s,z')$ and the t-prefixes of the q queries made during the computation of $H^\pi(s,z)$ are the same as the t-prefixes of the q queries made during the computation of $H^\pi(s,z')$. Clearly, for any adversary A of size $\leq S_h$, and any permutation π of security S_p we have

$$\mathop{\mathbf{Pr}}_{s,z}[A^\pi(z,s,H^\pi(s,z)) = z' : z' \text{ is strong collision for } z \text{ with respect to } s \text{ and } \pi] \leq \epsilon$$

After making a restriction on the definition of success, we now also make a restriction on the class of adversaries: instead of considering an adversary A that can access π arbitrarily, we consider adversaries that do not have oracle access to π, but are given the t-prefixes of the queries and answers during the computation of $H^\pi(s,z)$. Since such restricted adversaries can be simulated by general adversaries with no overhead, we also have that for every permutation π of security S_p and every A of size $\leq S_h$

$$\mathop{\mathbf{Pr}}_{s,z}[A(z,s,H^\pi(s,z),x_1,\ldots,x_q,y_1,\ldots,y_q) = z' : z' \text{ is strong collision for } z \text{ with respect to } s \text{ and } \pi] \leq \epsilon$$

where $x_1,\ldots,x_q$ are the t-prefixes of the q queries made to π during the computation of $H^\pi(s,z)$, and $y_1,\ldots,y_q$ are the t-prefixes of the respective answers.

Let now A be a fixed circuit of size $\leq S_h$, and let us sample a random permutation π from the set $\Pi_{t,n}$ (which we shorten with Π in the following). With high probability, π is one-way with security S_p (see Corollary 3), and in fact we have:

$$\mathop{\mathbf{Pr}}_{\pi\in\Pi}[\mathop{\mathbf{Pr}}_{s,z}[A(z,s,H^\pi(s,z),x_1,\ldots,x_q,y_1,\ldots,y_q) = z' : z' \text{ is strong collision for } z] > \epsilon\,] < \epsilon$$

which implies

$$\mathop{\mathbf{Pr}}_{\pi\in\Pi,s,z}[A(z,s,H^\pi(s,z),x_1,\ldots,x_q,y_1,\ldots,y_q) = z' : z' \text{ is strong collision for } z] < 2\epsilon$$

By the same argument in the proof of Theorem 4 we can assume w.l.o.g. that H queries π only on points with distinct t-prefixes.

Consider now the hash function: $H' : \{0,1\}^{r'} \times \{0,1\}^{m+k} \to \{0,1\}^{m'}$ where $r' = r + qt$ and $m' = m + qt$ defined as follows[3]:

$$H'((s,y_1,\ldots,y_q),z) = (H^{y_1,\ldots,y_q}(s,z),x_1,\ldots,x_q)$$

[3] As in the case of the PRG less than qt random bits are actually used to sample q distinct random elements $y_1 \neq \ldots \neq y_q$ in $\{0,1\}^t$

where by $H^{y_1,\ldots,y_q}(s,z)$ we mean the computation of $H^{(\cdot)}(s,z)$ when the t-prefix of the answer to the i-th query is y_i (and the remaining bits of the answer are equal to the remaining bits of the query); and where we denote by $x_1,\ldots,x_q$ the t-prefixes of the q queries made during such computation.

Notice that if $q < k/t$ then $m' < m + k$, i.e. this is a length-decreasing function. Now clearly for any adversary A of size S_h we have that

$$\begin{aligned}\mathop{\mathbf{Pr}}_{s',x}[&A(s',z,H'(s',z)) = z' : z \neq z' \text{ and} \\ &H'(s',z) = H'(s',z')] = \mathop{\mathbf{Pr}}_{\pi\in\Pi,s,z}[\\ &A^{\pi}(z,s,H^{\pi}(s,z),x_1,\ldots,x_q,y_1,\ldots,y_q) = z' : \\ &z' \text{ is strong collision for } z] < 2\epsilon\end{aligned}$$

which means that H' is $(S_h, 2\epsilon)$-UOWH. Notice lastly that H' does not invoke any oracle. □

References

[BM84] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, 1984. Preliminary version in *Proc. of FOCS'82*.

[CG00] D. Catalano and R. Gennaro. Trapdoor permutations with lots of hard bits: or efficient secure encryption without random oracle or decisional assumptions. Manuscript, 2000.

[CS98] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Proceedings of CRYPTO'98*, pages 13–25. LNCS 1462, 1998.

[DH78] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6), 1978.

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

[GL89] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st ACM Symposium on Theory of Computing*, pages 25–32, 1989.

[GMW91] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity. *Journal of the ACM*, 38(3), 1991.

[Gol95] O. Goldreich. Foundations of cryptography — fragments of a book. Unpublished Monograph, 1995.

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

[HSS93] J. Håstad, A. Schrift, and A. Shamir. The discrete logarithm modulo a composite hides $O(n)$ bits. *Journal of Computer and System Sciences*, 47:376–404, 1993.

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

[IR89] R. Impagliazzo and S. 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.

[Nao91] M. Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991.

[NY89] M. Naor and M. Yung. Universal hash functions and their cryptographic applications. In *Proceedings of the 21st ACM Symposium on Theory of Computing*, 1989.

[Rom90] J. Rompel. One-way functions are necessary and sufficient for digital signatures. In *Proceedings of the 22nd ACM Symposium on Theory of Computing*, 1990.

[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.

[Sim98] D. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Proceedings of EUROCRYPT'98*, 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.