Bounds on the Efficiency of Generic Cryptographic Constructions[*]

Rosario Gennaro[†] Yael Gertner[‡] Jonathan Katz[§]
Luca Trevisan[¶]

Abstract

A central focus of modern cryptography is the construction of efficient, "high-level" cryptographic tools (e.g., encryption schemes) from weaker, "low-level" cryptographic primitives (e.g., one-way functions). Of interest are both the *existence* of such constructions, and also their *efficiency*.

Here, we show essentially-tight lower bounds on the best possible efficiency that can be achieved by any black-box construction of some fundamental cryptographic tools from the most basic and widely-used cryptographic primitives. Our results concern constructions of pseudorandom generators, universal one-way hash functions, private-key encryption schemes, and digital signatures based on one-way permutations, as well as constructions of public-key encryption schemes based on trapdoor permutations. Our proofs are in the model introduced by Impagliazzo and Rudich: in each case, we show that any black-box construction beating our efficiency bound would yield the unconditional existence of a one-way function and thus, in particular, prove $P \neq NP$.

1 Introduction

A central focus of modern cryptography is the construction of "high-level" cryptographic protocols and tools that are both provably secure and efficient. Generally speaking, work proceeds along two lines: (1) demonstrating the *feasibility* of a particular construction, based on the weakest possible primitive; and (2) improving the *efficiency* of such constructions, either based on the weakest primitive for which a construction is known or perhaps by assuming the existence of a stronger primitive. The first of these approaches has been immensely successful; for example, the existence of one-way functions is known to be sufficient for constructing pseudorandom generators [8, 42, 22, 27], pseudorandom functions [21], universal one-way hash functions and digital signature schemes [36, 38], private-key encryption schemes and message-authentication codes [20], and commitment schemes [35]. In

[*] This work appeared in preliminary form as [15, 14].

[†] rosario@watson.ibm.com. IBM T.J.Watson Research Center.

[‡] ygertner@uiuc.edu. Department of Psychology, University of Illinois at Urbana-Champaign. Work done while at the University of Pennsylvania.

[§] jkatz@cs.umd.edu. Department of Computer Science, University of Maryland. Portions of this work were done while at DIMACS.

[¶] luca@eecs.berkeley.edu. Computer Science Division, UC Berkeley. Work done while at Columbia University. Work supported by NSF grant CCR 9984703.

each of these cases one-way functions are also known to be necessary [30, 38], thus exactly characterizing the feasibility of these constructs.

Unfortunately, progress on the second approach — improving the efficiency of these constructions — has been much less successful. Indeed, while the constructions referenced above are all important from a theoretical point of view, their practical impact has been limited precisely due to their inefficiency. In practice, more efficient constructions based on stronger assumptions (e.g., the hardness of a specific, number-theoretic problem) are used. Furthermore, stronger assumptions currently seem necessary to obtain improved efficiency; for each of the examples listed above, no constructions based on general assumptions are known which improve upon the efficiency of the initial solutions.

This trade-off between the efficiency of a cryptographic construction and the strength of the complexity assumption on which it relies motivates the question: *How efficient can cryptographic constructions be when based on general assumptions?* In this paper, we show that in fact, the efficiency of many of the known constructions based on general assumptions cannot be improved without using non-black-box techniques, or finding an unconditional proof that one-way functions exist (and hence proving $P \neq NP$).

1.1 Our Results

Informally, we say a one-way (trapdoor) permutation $\pi : \{0,1\}^n \rightarrow \{0,1\}^n$ has security S if any circuit of size at most S inverts π with probability less than $1/S$ (one can think of S as a slightly super-polynomial function of n but our results hold for any choice of S). Given this definition, our results may be summarized as follows.

Pseudorandom generators (PRGs). Let U_ℓ denote the uniform distribution over ℓ-bit strings. A PRG is a deterministic, length-increasing function $G : \{0,1\}^\ell \rightarrow \{0,1\}^{\ell+k}$ such that $G(U_\ell)$ is computationally indistinguishable (by poly-time algorithms) from $U_{\ell+k}$. The notion of a PRG was introduced by Blum and Micali [8] and Yao [42], who showed that PRGs can be constructed from any one-way permutation. (This was subsequently improved by Håstad, et al. [27], who show that a PRG can be constructed from any one-way function.) Their construction, using a later improvement of Goldreich and Levin [22] (see also [18, Section 2.5.3]), requires $\Theta(k/\log S)$ invocations of a one-way permutation with security S in order to construct a PRG stretching its input by k bits. This is the best known efficiency for generic constructions (i.e., constructions based on an *arbitrary* one-way permutation).

We show that this is essentially the best efficiency that can be obtained using generic constructions. More formally, we show that any (black-box) generic construction of a PRG that stretches its input by k bits while making fewer than $\Omega(k/\log S)$ invocations of a one-way permutation with security S implies the *unconditional* existence of a PRG (i.e., without any invocations of the one-way permutation). Put another way, the only way to come up with a more efficient construction of a PRG is to design a PRG from scratch! This would in particular imply the unconditional existence of a one-way function, as well as a proof that $P \neq NP$.

(Families of) universal one-way hash functions (UOWHFs). A UOWHF $\mathcal{H} = \{h_s\}$ is a family of length-decreasing functions (all defined over the same domain and range) such that for any input x and random choice of $h_i \in \mathcal{H}$ it is hard to find a *collision* (i.e., a $y \neq x$ such that $h_i(y) = h_i(x)$). UOWHFs were introduced by Naor and Yung [36], who show that UOWHFs suffice to construct secure signature schemes and furthermore show how to construct the former from any one-way permutation. (Rompel [38] later gave a construction of UOWHFs, and hence signature schemes, based on any one-way function.) The Naor-Yung construction requires one invocation of the one-way permutation per bit of compression; that is, if $h_i : \{0,1\}^{\ell+k} \rightarrow \{0,1\}^{\ell}$ (for all $h_i \in \mathcal{H}$), then evaluating h_i requires k invocations of the one-way permutation. This can be improved easily to $\Theta(k/\log S)$ invocations for compression by k bits.

We show that this, too, is essentially optimal. In particular, any (black-box) generic construction of a UOWHF whose hash functions compress their input by k bits yet can be evaluated using fewer than $\Omega(k/\log S)$ invocations of a one-way permutation (with security S) implies the *unconditional* existence of a UOWHF. Since the existence of UOWHFs implies the existence of one-way functions, this consequence would again imply a proof of $P \neq NP$. Our result improves upon a result of Kim, Simon, and Tetali [33], who show a bound of $\Omega(\sqrt{k/\log S})$ invocations.[1]

Encryption schemes. PRGs and UOWHFs may be viewed as 1-party, or "stand-alone", cryptographic primitives for which there is no inherent notion of "interaction". We also explore the efficiency of 2-party protocols, including those used in a "public-key" setting.

A public-key encryption scheme for m-bit messages is semantically-secure [25] if for any two messages $M_0, M_1 \in \{0,1\}^m$ the distribution over encryptions of M_0 is computationally indistinguishable from the distribution over encryptions of M_1, even when given the public key as input. A similar definition (but with no public key) holds for the case of private-key encryption. Public-key encryption schemes constructed using the hard-core bit paradigm [8, 7, 22, 42] require $\Theta(m/\log S)$ invocations of a trapdoor permutation to encrypt an m-bit message. Similarly, private-key encryption schemes constructed using this paradigm require $\Theta(\frac{m-k}{\log S})$ invocations of a one-way permutation, where k is the length of the shared key. (For completeness, we note that while secure private-key encryption with $m > k$ is equivalent to the existence of one-way functions [30, 27], there is evidence that public-key encryption and trapdoor functions are not equivalent with respect to black-box reductions [6, 17].)

We show that the above constructions are essentially the best possible (at least, for the notions of security considered above). For the case of public-key encryption, we show that an encryption scheme for m-bit messages requires $\Omega(m/\log S)$ invocations of the trapdoor permutation for the encryption algorithm alone (i.e., in addition to any invocations made during the key-generation phase). Using related techniques, we also show that any construction of a private-key encryption scheme which securely encrypts m-bit messages using a k-bit key must query the one-way permutation $\Omega(\frac{m-k}{\log S})$ times. In each case, we show that any black-box construction beating our bound would imply the unconditional existence of a one-way function (from which a secure private-key encryption scheme, with no reliance on an oracle, can be derived), and hence a proof that $P \neq NP$.

Signature schemes. A one-time signature scheme is secure if no efficient algorithm can forge a valid signature on a new message after seeing a signature on a single, random message (any scheme satisfying this definition may be easily converted — with only a factor of two loss in efficiency — to one which is secure when an adversary sees a signature on a single,

[1] Besides being quantitatively better, our result is also qualitatively better since it holds in a more general model; see the discussion in Section 1.3 about different models for proving black-box lower bounds.

chosen message; see Section 2.5). Of course, lower bounds on one-time schemes immediately extend to schemes satisfying stronger definitions of security [26]. For any one-time signature scheme for messages of length m, we show that the verification algorithm must evaluate the one-way permutation $\Omega(m/\log S)$ times. As before, any black-box construction beating our bound implies the unconditional existence of a one-way function (from which a secure signature scheme requiring no oracle access can be constructed).

We are not aware of any work explicitly concerned with upper bounds on the efficiency of signature verification. However, we observe that there exist schemes essentially meeting our lower bound; see Section 3.5 for further discussion.

1.2 Overview of Techniques

We prove our results in an extension of the model of Impagliazzo and Rudich [31]. Informally (see Section 1.3 for a more detailed discussion of models for black-box impossibility results), Impagliazzo and Rudich proved that a black-box construction of a secure key-exchange protocol based on an arbitrary one-way permutation would inherently yield a proof that $P \neq NP$. Here, we show that a similar consequence (i.e., a proof that $P \neq NP$) results from any black-box construction beating our efficiency bounds. In fact, we prove an even stronger statement: any such construction would prove the unconditional existence of a one-way function; this is stronger both because the existence of a one-way function is not known to be implied by $P \neq NP$, and also because (in all but one case) a one-way function suffices to give an *unconditional* construction of the object under consideration.

Each of our proofs hinges on a technical lemma which, though perhaps "folklore", has not previously been stated or proved explicitly to the best of our knowledge: a random permutation on t-bit strings is, with high probability, one-way with security $2^{-\Omega(t)}$ even against non-uniform adversaries. For the related case of random functions, a similar result has been proven by Impagliazzo and Rudich [31] in the (much simpler) uniform case, and by Impagliazzo [29] in the non-uniform case.[2]

Using this lemma, we now describe the intuition behind our lower bound in the case of PRGs. Given a secure construction G of a PRG with oracle access to a one-way permutation over n-bit strings, we run G with a permutation oracle that randomly permutes its first $t = \mathcal{O}(\log S)$ bits while leaving the remaining $n - t$ bits unchanged. It follows from the technical lemma above that with high probability such a permutation is one-way with security S, and hence G is secure when run with a permutation chosen from this distribution.

Let q be the number of queries made by G to its oracle. The key point of our proof is to notice that the answers to these q oracle queries can be "simulated" by a deterministic function G' *itself* (i.e., without access to any oracle) if $q \cdot t$ random bits, representing the t-prefixes of the q answers to G's oracles queries, are included as part of the input (or seed) of G'. The distribution on the output of G' (over random choice of seed) is identical to that of the output of G (over random seed, and random choice of oracle as above), and is thus indistinguishable from uniform. Finally, if q is "small" then the seed-length does not grow "too much" and the input of G' remains shorter than its output. But this means that G' is an *unconditional* PRG, which does not require any oracle access (a corollary of which is a

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

proof that $P \neq NP$).

Additional technical work is needed to prove our bounds on UOWHFs, public-key encryption schemes, and digital signature schemes. In the latter two cases in particular, which are in a "public-key" setting, there is no "seed" as part of which to include the necessary randomness for answering oracle queries, and thus no immediate way to apply the above technique. Although randomized algorithms are allowed in these settings (in contrast to the case of PRGs and UOWHFs, which must be deterministic), consistency must be ensured between the two interacting parties if we are to transform, say, an encryption scheme which uses an oracle to one which does not. It should be obvious that including the necessary randomness in the public key may ruin the security of the protocol.

The proofs of our lower bounds in these cases follow a slightly different approach. In the case of public-key encryption, for example, we show that a scheme making fewer than the prescribed number of queries can be used to construct (unconditionally) a secure *private-key* encryption scheme in which the key is shorter than the message. Moving from the public-key to the private-key setting circumvents the issues above, and enables the necessary randomness to be included as part of the shared key without compromising the functionality or the security of the scheme. Unfortunately, our result in this case is somewhat weaker than what we obtain in all other cases; namely, we show that a *public*-key encryption scheme making "few" black-box oracle queries exists only if a secure *private*-key encryption scheme (or, equivalently, a one-way function) exists unconditionally. This is, of course, weaker than showing the unconditional existence of a secure public-key encryption scheme. (We stress that for the case of PRGs, UOWHFs, private-key encryption schemes, and signature schemes, constructions beating our bounds imply the existence of an unconditional construction for each of these tasks.)

1.3 Black-Box Lower Bounds and Impossibility Results

Our results hold in a generalization of the Impagliazzo-Rudich model [31, 39] (described below), which has been used to separate primitives under so-called *black-box reductions*. Informally, a black-box reduction of Q to P is a construction of Q out of P that ignores the internal structure of the circuit implementing P. Furthermore, the security of the resulting implementation of Q is based solely on the security of P. The fact that Q depends only the input/output characteristics of P (and not its internal structure) means that an oracle can be substituted for P in constructions of Q.

Impagliazzo and Rudich [31, 39] were concerned with the question of whether key-exchange protocols could be constructed based on one-way functions. They focus on "black-box" constructions, as discussed above. More precisely, Impagliazzo and Rudich define a black-box key-exchange protocol based on one-way functions as a pair of oracle procedures $(A^{()}, B^{()})$ for the two parties Alice and Bob such that the following holds: If f is a one-way function in the oracle sense (that is, if there is no efficient oracle procedure $I^{()}$ such that I^f inverts f), then (A^f, B^f) is "secure" (in the appropriate attack model) against every efficient adversary E^f given oracle access to f. This formalization of a black-box reduction is called *semi-black-box* in [37]. Note that a construction in which A and B use the *code* of the one-way function f would not fall into this framework. In fact, even a construction in which f is used only as an oracle, but in which the *security reduction* uses the *code of the* *adversary* would not fall into this framework since, in the above definition, the construction is required to be secure even against adversaries given oracle access to f.

Impagliazzo and Rudich prove that if $P = NP$ and if f is a random function then every semi-black-box key-agreement protocol based on f can be broken by an adversary that has access to f. Furthermore, a random function is one-way with very high probability. This implies that if $P = NP$ then there cannot be any semi-black-box key-agreement protocol based on one-way functions. Alternately, if one could construct a semi-black-box key-agreement protocol based on one-way functions then a proof that $P \neq NP$ would follow. Reingold, et al. [37] have strengthened this result and proved *unconditionally* that there can be no semi-black-box construction of a key-agreement protocol based on one-way functions. The Impagliazzo-Rudich result may also be viewed as a proof that any construction of secure key-exchange from one-way functions does not relativize (see [37]): their proof shows that any construction of key-exchange from one-way functions fails relative to an oracle for which $P = NP$ (e.g., a PSPACE oracle) augmented by an oracle for a random function.

As we have briefly mentioned earlier, we extend the Impagliazzo-Rudich model in two ways. First, we rule out a larger class of black-box constructions which are called *weak black-box* in [37]. Weak black-box constructions formalize the notion of a construction of a primitive Q from a primitive P that uses P as a black box in the construction but allows for an arbitrary proof of security and, in particular, may use the code of the adversary in the security proof. For example, a weak black-box key-exchange protocol based on one-way functions is a pair of oracle procedures $(A^{()}, B^{()})$ for the two parties Alice and Bob such that the following holds: If f is a one-way function in the oracle sense (as defined above), then (A^f, B^f) is a secure key-exchange protocol against every efficient adversary E (note that E here is not given oracle access to f). The proof of security for a weak black-box reduction may use the code of the adversary, since the previous statement refers only to adversaries given no oracle access. The Impagliazzo-Rudich result does not rule out the existence of such a construction and, indeed, Reingold, et al. [37] prove that if the statement "one-way functions imply key agreement" is true, then a weak black-box construction of key agreement (for a single-bit key) from one-way functions exists. Our results here rule out even weak black-box constructions.

A second way in which our setting is stronger than that of Impagliazzo-Rudich is that we show that the existence of constructions contradicting our bounds imply that one-way functions exist, while Impagliazzo and Rudich derived the weaker conclusion that $P \neq NP$. (As noted above, Reingold, et al. [37] derive an unconditional contradiction instead of $P \neq NP$, but their technique only applies to semi black-box reductions. Furthermore, our work was completed before [37]. We note that using the techniques of [37], we can also show that semi black-box constructions beating our lower bounds are unconditionally impossible; however, we focus our presentation on weak black-box reductions in this paper.)

Finally, we note that Impagliazzo and Rudich were concerned with the question of *feasibility*, while we are concerned with questions of *efficiency*.

Previous impossibility results. Following the work of Impagliazzo and Rudich, a number of additional black-box impossibility results for cryptographic constructions have appeared [40, 41, 33, 16, 32, 17, 13]. In particular the work of Kim, Simon, and Tetali [33] initiated the study of efficiency limitations for cryptographic constructions, and provided the original inspiration for our research.

Non-black-box constructions. Black-box constructions form an important subclass, since most cryptographic constructions are black-box. We stress, however, that a number of non-black-box cryptographic constructions are known.[3] The examples of which we are aware occur in two ways: due to the use of zero-knowledge proofs (of knowledge) [23, 12, 4] or due to the use of protocols for secure computation [43, 24]. To illustrate, for af function, let L_f denote the image of f; i.e., $L_f \stackrel{\text{def}}{=} \{y|y = f(x)\}$. A cryptographic protocol which utilizes a zero-knowledge proof that $y \in L_f$ (for f a one-way function, say) requires the parties to agree on a circuit computing f and is thus inherently non-black-box.[4] Examples of non-black-box constructions where zero-knowledge proofs of this sort are used include a construction of an identification protocol based on one-way functions [12], a signature scheme based on non-interactive zero-knowledge [5], and *all* known constructions of chosen-ciphertext-secure encryption schemes from trapdoor permutations (e.g., [11]). Furthermore, protocols for distributed computation (without honest majority) tolerating computationally-bounded, malicious adversaries [24, 43] are *themselves* non-black-box[5] (this is in contrast to the case of zero-knowledge proofs; cf. footnote 4).

Knowledge of the circuit computing f is also necessary if one wants to evaluate f in a secure, distributed fashion (e.g., if two parties with respective inputs x_1, x_2 want to evaluate $y = f(x_1 \oplus x_2)$ without revealing any more information about their inputs than what is revealed by y itself). Thus, a straightforward construction of a threshold cryptosystem [10] based on a family F of trapdoor permutations (in which the parties share the trapdoor for inverting a single member of this family) would inherently make non-black-box use of the underlying circuit(s) for F. Another example is a result of Beaver [3] which makes non-black-box use of a one-way function f to extend "few" oblivious transfers into "many".

Given the above, a black-box impossibility result cannot be said to rule out the feasibility of a particular construction. Yet, it is unclear how non-black-box techniques can help outside the domains mentioned above (e.g., zero-knowledge proofs or secure computation). Furthermore, a black-box impossibility result is useful insofar as it indicates the type of techniques that will be necessary to achieve a desired result, or, conversely, the type of techniques that are ruled out. Finally, it is fair to say that non-black-box constructions are much less efficient than black-box ones (this is certainly the case for all the examples given above, and we are aware of no exceptions), and thus a black-box impossibility result does seem to rule out constructions likely to be *practical*.

1.4 Future Work and Open Problems

This work suggests a number of intriguing research directions. The results given here indicate that assuming the existence of one-way permutations is too weak of a computational

[3] We focus here on constructions making non-black-box use of some underlying primitive, rather than on constructions whose security analysis makes non-black-box use of the adversary (as in [1, 2]).

[4] Note that constructions of zero-knowledge proofs for NP (e.g., [23]) are *themselves* black-box in their usage of primitives such as one-way functions. The issue is that a proof for the language of interest — e.g., L_f in the example in the text — cannot be given unless a (poly-size) circuit computing f is available.

[5] For example: although the well-known protocol for oblivious transfer secure against *semi-honest* adversaries [19, Sec. 7.3.2] makes black-box use of trapdoor permutations, adapting the protocol (using zero-knowledge proofs) to ensure security against *malicious* adversaries involves *non*-black-box use of the trapdoor permutation.

hypothesis to obtain *efficient* cryptographic constructions. Thus, stronger assumptions are needed to build practical schemes. It is important to explore the "minimal" such assumptions necessary to achieve greater efficiency, as well as to bound the maximum achievable efficiency even when such stronger assumptions are made. For example: what additional efficiency is possible if *homomorphic* one-way permutations are assumed?

In a related vein, it will be interesting to explore number-theoretic assumptions which lead to more efficient schemes. As will be evident from our proof techniques, the efficiency limitations of constructions based on arbitrary (trapdoor) one-way permutations stem from the fact that a one-way permutation may have security S even if it has only $\Theta(\log S)$ "hard" bits. (Actually, we use "pathological" functions of this form to prove our lower bounds.) But specific one-way permutations and trapdoor permutations with $\Theta(n)$ hard-core bits are known under suitable number-theoretic assumptions (e.g., [28, 9]. Given such functions, we know how to construct PRGs and semantically-secure private-/public-key encryption schemes with improved efficiency. It remains open, however, whether such functions can also be used to improve the efficiency of digital signature schemes or (say) public-key encryption schemes achieving chosen-ciphertext security.

The present work also leaves some immediate open questions. First, can bounds on the efficiency of other cryptographic constructions (e.g., commitment schemes) also be given? Additionally, our lower bounds for encryption and signature schemes essentially match the known upper bounds only for schemes achieving relatively weak notions of security: namely, semantic security for encryption of a single message and one-time security, respectively. What can be said about schemes achieving stronger notions of security? Examples of interest include private-key encryption schemes secure when polynomially-many messages are encrypted (this seems intimately related to the efficiency of pseudorandom *functions*, for which a gap remains between known upper and lower bounds), public-key encryption schemes satisfying various flavors of non-malleability/security against chosen-ciphertext attacks, and signature schemes secure when polynomially-many messages are signed.

Furthermore, in the case of signatures our bounds pertain to the efficiency of signature verification. It would be nice to have corresponding bounds for the efficiency of key-generation/signing. (Note that the Lamport scheme [34] requires no oracle calls when signing an m-bit message but $O(m)$ oracle calls for key generation; on the other hand, a scheme which hashes the message before signing requires fewer oracle calls during key generation but more when signing. Thus, it seems there is a trade-off between the complexity of the two.)

2 Definitions and Preliminaries

2.1 One-Way Permutations and Trapdoor Permutations

One-way functions/permutations. We say that a function $\pi : \{0,1\}^n \rightarrow \{0,1\}^n$ is (S, ε)-*one way* if for every circuit A of size $\leq S$ we have $\Pr_x[A(f(x)) \in f^{-1}(f(x))] \leq \varepsilon$. When f is given as an oracle (as will often be the case in this work), we provide A with access to f and write this as A^f. To reduce the number of parameters, we will call a function S-*hard* if it is $(S, 1/S)$-one way.

We now prove that a random permutation is, with high probability, one-way with exponential security even against non-uniform circuits; this extends related results in the context of random *functions* [31, 29] (our proof is also somewhat simpler than these previous proofs). Let Π_t denote the set of all permutations over $\{0,1\}^t$.

Theorem 1 *For all sufficiently large t, a random $\pi \in \Pi_t$ is $2^{t/5}$-hard with probability at least $1 - 2^{-2^{t/2}}$.*

Proof We begin by showing that given any π, A such that A inverts π with "high" probability, the permutation π has a "short" description (given A).

Claim *Let A be a circuit that makes q queries to a permutation $\pi : \{0,1\}^t \to \{0,1\}^t$, and for which $\Pr_y[A^\pi(y) = \pi^{-1}(y)] \geq \varepsilon$. Then π can be described using at most*

$$2 \log \binom{2^t}{a} + \log \left((2^t - a)!\right)$$

bits (given A), where $a \stackrel{\text{def}}{=} \varepsilon 2^t/(q+1)$.

Proof (of claim) Let $N = 2^t$. Consider the set I of εN points on which A is able to invert π, after making q queries to π. We argue that there exists a set $Y \subseteq I$ such that $|Y| \geq a$ and such that the value of π^{-1} on Y is completely determined by A, the sets Y and $X \stackrel{\text{def}}{=} \pi^{-1}(Y)$, and the value of π^{-1} on all points $\{0,1\}^t \setminus Y$.

Define Y via the following process: initially Y is empty, and all elements in I are candidates for inclusion in Y. Take the lexicographically first element y from I, and place it in Y. Next, simulate the computation of $A^\pi(y)$ and let $x_1, \ldots, x_q$ be the queries made by A to π (we assume without loss of generality that they are different), and $y_1, \ldots, y_q$ be the corresponding answers (i.e., $y_i = \pi(x_i)$). If y is none of the answers, then remove $y_1, \ldots, y_q$ from I. If $y = y_i$ for some i, then remove $y_1, \ldots, y_{i-1}$ from I. Then take the lexicographically smallest of the remaining elements of I, put it into Y, etc. At any step of the construction, one element is added to Y and at most q are removed from I. Since I initially contains εN elements, in the end we have $|Y| \geq \lceil \varepsilon N/q \rceil > \varepsilon N/(q+1)$.

(Note: a way to picture the previous argument is to draw a directed graph with 2^t nodes, where there is an edge (y, y') iff $A^\pi(y)$ 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 Y of I such that all edges (y, y') in the subgraph induced by Y satisfy $y \geq_{lex} y'$. A greedy algorithm will find a Y such that $|Y| \geq |I|/q$.)

We claim that given descriptions of the sets Y and X, the values of π on $2^t \setminus X$, and the circuit A, it is possible to compute (or, equivalently, invert) π everywhere. For $y \notin Y$, the value of $\pi^{-1}(y)$ is explicitly given. The values of π^{-1} on Y can be reconstructed sequentially for all $y \in Y$, taken in lexicographic order, as follows: Simulate the computation of $A^\pi(y)$. By construction of Y, during its computation $A^\pi(y)$ will query π either on points not in X, on points $x \in X$ for which $\pi(x) <_{lex} y$, or on the point $x \in X$ for which $\pi(x) = y$. In the first two cases, we have enough information to continue the simulation. In the last case, the query itself gives the desired answer $\pi^{-1}(y)$. In all possible cases, we have enough information to reconstruct $\pi^{-1}(y)$.

Describing Y, X, and the values of π on $2^t \setminus X$ requires $2 \log \binom{N}{|Y|} + \log((N - |Y|)!)$ bits, which is at most the number of bits claimed. □

Given the above claim, we may now easily prove the theorem. Let A be an oracle circuit of size at most $S = 2^{t/5}$. Note that A will make at most $q = 2^{t/5}$ queries to π. Let $N = 2^t$. From the claim, we see that the fraction of permutations $\pi \in \Pi_t$ such that

$$\Pr_x[A^\pi(\pi(x)) = x] \geq 2^{-t/5} \tag{1}$$

is at most

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

where $a \geq 2^{-t/5} 2^t/(2^{t/5}+1) > N^{3/5}/2$. Using the inequalities $a! < (a/e)^a$ and $\binom{N}{a} > (eN/a)^a$, the expression above is upper bounded by

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

for all sufficiently large N.

Since there are at most $2^{St \log S} = 2^{N^{1/5} \log^2 N/5}$ circuits of size S, a union bound shows that the probability over a random choice of $\pi \in \Pi_t$ that there *exists* a circuit of size at most S for which Equation (1) holds is at most

$$2^{N^{1/5} \log^2 N/5} \cdot 2^{-N^{3/5}/2} < 2^{-N^{1/2}}$$

for all sufficiently large N. ■

We did not try to optimize the constants in the above proof, since the statement of the theorem suffices for our applications.

For $t \leq n$, let $\Pi_{t,n}$ denote the subset of Π_n such that $\pi \in \Pi_{t,n}$ iff $\pi(a,b) = (\hat{\pi}(a), b)$ for some $\hat{\pi} \in \Pi_t$ (that is, π permutes the first t bits of its input, while leaving the remaining $n-t$ bits fixed). An immediate corollary of the above theorem is that if $t = 5 \log S$, then for any $n \geq t$ a random $\pi \in \Pi_{t,n}$ is one-way with security S with very high probability.

Corollary 1 *For all sufficiently large t and any $n \geq t$, a random $\pi \in \Pi_{t,n}$ is $2^{t/5}$-hard with probability greater than $1 - 2^{-2^{t/2}}$.*

(One-way) trapdoor permutations. Our model for *trapdoor* permutations is somewhat more involved. We represent a family of trapdoor permutations as a tri-partite oracle $\tau = (G, F, F^{-1})$. Informally, G corresponds to the *key generation* oracle which when queried on a string td (intended as a "trapdoor") produces the corresponding public key k. The oracle F is the actual trapdoor permutation, which will be queried on key k and an input x. The oracle F^{-1} allows inversion of F; i.e., if $G(td) = k$ and $F(k,x) = y$, then $F^{-1}(td, y) = x$.

More formally, consider the class $T_n = \{\tau \mid \tau = (G, F, F^{-1})\}$ where:

- $G \in \Pi_n$ is a permutation over $\{0,1\}^n$. (Allowing G to be a function rather than a permutation does not affect our results.)
- $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is an oracle such that, for each $k \in \{0,1\}^n$, $F(k, \cdot)$ is a permutation on $\{0,1\}^n$.

- $F^{-1} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is an oracle defined as follows: $F^{-1}(td, y)$ returns the unique x such that $G(td) = k$ and $F(k, x) = y$.

A uniformly random $\tau = (G, F, F^{-1}) \in T_n$ is chosen in the natural way: G is chosen at random from Π_n and, for each $k \in \{0,1\}^n$, the permutation $F(k, \cdot)$ is chosen independently at random from Π_n. We say that trapdoor permutation family $\tau = (G, F, F^{-1})$ is (S, ε)-trapdoor one way if for every circuit A of size $\le S$ we have

$$\mathop{\mathbf{Pr}}_{x,td}[k := G(td) \ : \ A^\tau(k, F(k,x)) = x] \le \varepsilon.$$

We say that τ is *S-trapdoor one way* if it is $(S, 1/S)$-trapdoor one way. When clear from the context, we will also say that τ is *S-hard*.

Although technically one must always speak of families of trapdoor permutations, we will often abuse terminology and simply refer to a $\tau \in T_n$ as a trapdoor permutation.

It is not too difficult to show an analogue of Theorem 1 for trapdoor permutations. The basic idea is to notice that $F(k, \cdot)$ is a random permutation for any k so Theorem 1 applies, except that we need to also take into account the possibility that the circuit can invert G. But since G is also a random permutation, all we need to do is apply Theorem 1 twice.

Theorem 2 *For all sufficiently large t, a random $\tau \in T_t$ is $2^{t/6}$-trapdoor one-way with probability greater than $1 - 2^{1-2^{t/2}}$.*

Proof We actually show that τ is $(2^{t/5}, 2^{1-t/5})$-trapdoor one-way; the statement of the Theorem follows (again, we did not try to optimize constants).

From Theorem 1 we know that with probability at least $p = 1 - 2^{-2^{t/2}}$ the permutation G is one-way with security $2^{t/5}$. Since $F(k, \cdot)$ is a random permutation for any k, Theorem 1 also guarantees that with probability at least p the permutation $F(k, \cdot)$ is one-way with security $2^{t/5}$. Thus, with probability at least $p^2 > 1 - 2^{1-2^{t/2}}$ both G and $F(k, \cdot)$ are $2^{t/5}$-one way.

For a given circuit A and permutation family τ, let Q_{td} denote the event that $A^\tau(k, y)$ queries $F^{-1}(td, y')$ where y' is arbitrary and td is such that $G(td) = k$. We have:

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{x,td}[k := G(td) \ : \ A^\tau(k, F(k,x)) = x] \le \\
&\qquad \mathop{\mathbf{Pr}}_{x,td}[k := G(td) \ : \ A^\tau(k, F(k,x)) = x \mid \overline{Q_{td}}] \qquad (2)\\
&\qquad + \mathop{\mathbf{Pr}}_{x,td}[k := G(td) \ ; \ x := A^\tau(k, F(k,x)) \ : \ Q_{td}]. \qquad (3)
\end{aligned}$$

Now, (2) can be upper-bounded by $2^{-t/5}$ since we are conditioning on the event that A does not query F^{-1} using the correct td (and hence F^{-1} does not help A invert F). Similarly, (3) can be upper bounded by $2^{-t/5}$ since A can be used as a circuit to invert G on input k. Summing gives the desired result. ■

For $t \le n$, we let $T_{t,n}$ be the subset of T_n such that $\tau = (G, F, F^{-1}) \in T_{t,n}$ iff:

- $G \in \Pi_{t,n}$, and thus $G(td_a, td_b) = (\hat{G}(td_a), td_b)$ for some $\hat{G} \in \Pi_t$.

- $F((k_a, k_b), (x_a, x_b)) = (\hat{F}(k_a, x_a), x_b)$, where $\hat{F}(k_a, \cdot) \in \Pi_t$. Equivalently, $F(k, \cdot) \in \Pi_{t,n}$ and furthermore this permutation is determined by the first t bits of k.
- As before, $F^{-1}(td, y)$ returns the unique x such that $G(td) = k$ and $F(k, x) = y$.

An immediate corollary of Theorem 2 is that if $t = 6 \log S$, then for any $n \geq t$ a random $\tau \in T_{t,n}$ is S-hard with very high probability.

Corollary 2 *For all sufficiently large t and any $n \geq t$, a random $\tau \in T_{t,n}$ is $2^{t/6}$-hard with probability at least $1 - 2^{1-2^{t/2}}$.*

2.2 Pseudorandom Generators

We say two distributions X, Y are (S, ε)-indistinguishable if for every distinguishing circuit Dist of size at most S we have

$$\left| \Pr_{x \in X}[\mathsf{Dist}(x) = 1] - \Pr_{x \in Y}[\mathsf{Dist}(x) = 1] \right| \leq \varepsilon.$$

We will also write this as $X \overset{(S,\varepsilon)}{\approx} Y$. (In case X and/or Y depend on an oracle τ, then Dist is given access to τ as well.)

A PRG construction (from a one-way permutation) is an oracle procedure $G^{(\cdot)} : \{0,1\}^\ell \to \{0,1\}^{\ell+k}$ that expects as an oracle a permutation $\pi \in \Pi_n$. We are interested in constructions where G is computable in time polynomial in ℓ. We say that G is an (S_g, ε)-secure PRG if $G(U_\ell)$ is (S_g, ε)-indistinguishable from $U_{\ell+k}$, where U_n denotes the uniform distribution over $\{0,1\}^n$. Finally, we say that $G^{(\cdot)}$ is an (S_p, S_g, ε)-PRG construction if for every permutation π that is S_p-hard, G^π is an (S_g, ε)-secure PRG (where indistinguishability holds even for circuits given oracle access to π).

2.3 Universal One-Way Hash Functions

As discussed in the Introduction, a family of universal one-way hash functions (UOWHFs) is a family $\mathcal{H}$ of length-decreasing functions such that, for a random function $h \in \mathcal{H}$ and a random point x in the domain, it is hard (given h, x) to find $x' \neq x$ such that $h(x') = h(x)$. More formally, a family $\mathcal{H} = \{h_s\}_{s \in \{0,1\}^r}$ of functions $h_s : \{0,1\}^{\ell+k} \to \{0,1\}^\ell$ is an (S, ε)-UOWHF if for every circuit A of size at most S we have

$$\Pr_{s,x}[A(s, x, h_s(x)) = x' : x' \neq x \bigwedge h_s(x') = h_s(x)] \leq \varepsilon.$$

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

A construction of a UOWHF from a one-way permutation is an oracle procedure $H^{(\cdot)}(\cdot, \cdot)$ that expects as an oracle a permutation $\pi \in \Pi_n$ and is given inputs $s \in \{0,1\}^r$ and $x \in \{0,1\}^{\ell+k}$. The output is $H^\pi(s, x) \in \{0,1\}^\ell$. We say that H is an (S_p, S_h, ε)-UOWHF construction if for every π that is S_p-hard, H^π is an (S_h, ε)-UOWHF (even for circuits given oracle access to π).

2.4 Public- and Private-Key Encryption

Public-key encryption. A construction of a public-key encryption scheme for m-bit messages (based on trapdoor permutations) is a tuple of oracle procedures $\mathcal{PKE}^{(\cdot)} = (\mathsf{Gen}^{(\cdot)}, \mathsf{Enc}^{(\cdot)}, \mathsf{Dec}^{(\cdot)})$ that expects as an oracle a trapdoor permutation $\tau \in T_n$. These algorithms have the following functionality:

- The *key generation algorithm* $\mathsf{Gen}^{(\cdot)}$ is a probabilistic algorithm which generates a key pair (pk, sk). We say pk is the public key and sk is the secret key.
- The *encryption algorithm* $\mathsf{Enc}^{(\cdot)}$ is a probabilistic algorithm which, on input a public key pk and a message $M \in \{0,1\}^m$, outputs a ciphertext C.
- The *decryption algorithm* $\mathsf{Dec}^{(\cdot)}$ is a deterministic algorithm which, on input a secret key sk and a ciphertext C, outputs either a message $M \in \{0,1\}^m$ or $\perp$.

We assume that $\mathcal{PKE}$ is *correct*; that is, for all $\tau \in T_n$, all (pk, sk) output by Gen^{τ}, all $M \in \{0,1\}^m$, and all C output by $\mathsf{Enc}^{(\cdot)}(pk, M)$ we have $\mathsf{Dec}^{(\cdot)}(sk, C) = M$. Our results can be modified appropriately in case the scheme has a small probability of error.

For $M \in \{0,1\}^m$, let $\mathcal{PKE}(M)$ denote the distribution on the view of an adversary eavesdropping on the encryption of message M; i.e.,

$$\mathcal{PKE}(M) \overset{\text{def}}{=} \{(pk, sk) \leftarrow \mathsf{Gen}; C \leftarrow \mathsf{Enc}(pk, M) : (pk, C)\}.$$

We say that $\mathcal{PKE}$ is (S_e, ε)-secure if for all $M_0, M_1 \in \{0,1\}^m$ we have

$$\mathcal{PKE}(M_0) \overset{(S_e,\varepsilon)}{\approx} \mathcal{PKE}(M_1).$$

(Note that this corresponds to a definition of indistinguishability, or semantic security [25].) Finally, $\mathcal{PKE}^{(\cdot)}$ is an (S_p, S_e, ε)-PKE construction if for every oracle $\tau \in T_n$ that is S_p-trapdoor one way, $\mathcal{PKE}^{\tau}$ is (S_e, ε)-secure.

Private-key encryption. The model for private-key encryption is an appropriate modification of the above. A construction of a private-key encryption scheme for m-bit messages using k-bit keys (and based on trapdoor permutations) is a tuple $\mathcal{SKE}^{(\cdot)}$ that expects as an oracle a permutation $\pi \in \Pi_n$. In contrast to the public-key setting, $\mathcal{SKE}^{(\cdot)}$ consists only of algorithms $\mathsf{Enc}^{(\cdot)}$ and $\mathsf{Dec}^{(\cdot)}$. Algorithm $\mathsf{Enc}^{(\cdot)}$, which may be probabilistic, takes as input a key sk and a message $M \in \{0,1\}^m$ and outputs a ciphertext C. Algorithm $\mathsf{Dec}^{(\cdot)}$ is a deterministic algorithm which takes as input a key sk and a ciphertext C and outputs a message M. As in the public-key case, we assume that $\mathcal{SKE}^{(\cdot)}$ is *correct*; i.e., for all $\pi \in \Pi_n$, all $sk \in \{0,1\}^k$, all $M \in \{0,1\}^m$, and all C output by $\mathsf{Enc}^{\pi}(sk, M)$ we have $\mathsf{Dec}^{\pi}(sk, C) = M$.

For $M \in \{0,1\}^m$, denote by $\mathcal{SKE}^{\pi}(M)$ the probability distribution over the view of an adversary eavesdropping on the communication of message M (where the shared key sk is chosen uniformly at random); i.e.,

$$\mathcal{SKE}(M) \overset{\text{def}}{=} \left\{sk \leftarrow \{0,1\}^k; C \leftarrow \mathsf{Enc}(sk, M) : C\right\}.$$

We say that $\mathcal{SKE}$ is (S_e, ε)-secure if for all $M_0, M_1 \in \{0,1\}^m$ we have:

$$\mathcal{SKE}(M_0) \overset{(S_e,\varepsilon)}{\approx} \mathcal{SKE}(M_1).$$

Finally, $\mathcal{SKE}^{(\cdot)}$ is an (S_p, S_e, ε)-SKE construction if for every oracle $\pi \in \Pi_n$ that is S_p-hard, $\mathcal{SKE}^{\tau}$ is (S_e, ε)-secure.

We remark that the above definition can be extended in the natural way to allow for interactive encryption. Details are omitted.

2.5 Signature Schemes

A construction of a digital signature scheme for m-bit messages (based on one-way permutations) is a tuple of procedures $\mathcal{SIG}^{(\cdot)}$ that expects as an oracle a permutation $\pi \in \Pi_n$. These algorithms have the following functionality:

- The *key generation algorithm* $\mathsf{Gen}^{(\cdot)}$ is a probabilistic algorithm which generates a key pair (PK, SK). We say that PK is the public key and SK is the secret key.

- The *signing algorithm* $\mathsf{Sign}^{(\cdot)}$ is a probabilistic algorithm which, on input SK and a message $M \in \{0,1\}^m$, outputs a signature σ.

- The *verification algorithm* $\mathsf{Vrfy}^{(\cdot)}$ is a deterministic algorithm which, on input PK, a message M, and a signature σ, outputs a single bit.

We assume that $\mathcal{SIG}^{(\cdot)}$ is *correct*; i.e., for all $\pi \in \Pi_n$, all (PK, SK) output by Gen^{π}, all $M \in \{0,1\}^m$, and all σ output by $\mathsf{Sign}^{\pi}(SK, M)$ we have $\mathsf{Vrfy}^{\pi}(PK, M, \sigma) = 1$.

Our definition of a secure signature scheme is relatively weak: we only require the scheme to be secure against existential forgery for an adversary who gets a signature on a single, random message (i.e., we consider a *one-time signature scheme* secure under *random-message attack*). Our lower bounds apply even to "weakly-secure" schemes of this type. Since any signature scheme secure against adaptive chosen-message attack (cf. [26]) trivially achieves this "weak" level of security, our results immediately imply a bound for the more general case. Furthermore, we remark that it is easy to covert any such scheme to a one-time scheme secure against a *chosen-message* attack: run two of the "basic" schemes in parallel to obtain keys (PK_1, SK_1) and (PK_2, SK_2), set $PK = (PK_1, PK_2)$; to sign a message $M \in \{0,1\}^m$, choose random $r \in \{0,1\}^m$, sign r using SK_1 and $r \oplus M$ using SK_2, and output both signatures.

Formally, signature scheme $\mathcal{SIG}$ is (S_Σ, ε)-secure if for all circuits A of size at most S_Σ we have

$$\Pr\left[\begin{array}{l} (PK, SK) \leftarrow \mathsf{Gen}; M \leftarrow \{0,1\}^m; \\ \sigma \leftarrow \mathsf{Sign}(SK, M); (M', \sigma') := A(PK, M, \sigma) : \\ \mathsf{Vrfy}(PK, M', \sigma') = 1 \bigwedge M' \neq M \end{array}\right] \leq \varepsilon.$$

Finally, $\mathcal{SIG}^{(\cdot)}$ is an $(S_p, S_\Sigma, \varepsilon)$-signature construction if for every oracle $\pi \in \Pi_n$ that is S_p-hard, $\mathcal{SIG}^{\pi}$ is (S_Σ, ε)-secure.

3 Lower Bounds

3.1 Pseudorandom Generators

In this section we show our lower bound for PRG constructions. We start from 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 ℓ-bit seed into an $(\ell+k)$-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: let $t = O(\log S_p)$. First, note that if G uses as an oracle a π chosen randomly from $\Pi_{t,n}$, then G is a secure PRG with all but negligible probability (since a random permutation from $\Pi_{t,n}$ is S_p-hard with all but negligible probability). Now, if G queries the oracle at only a few (say, q) points, we can encode the answers to these queries in the seed of the PRG itself. Furthermore, this encoding is "short" since only t bits are needed to answer a query to $\pi \in \Pi_{t,n}$. We thus obtain a PRG $G' : \{0,1\}^{\ell'} \to \{0,1\}^{\ell+k}$ which uses no oracle at all, but which is able to "simulate" the computation of G when using a random permutation oracle. The desired bound comes from the fact that G' still stretches its input provided that ℓ' is smaller than $\ell+k$. But $\ell' \le qt + \ell$ since qt bounds the number of bits needed to encode the t-bit answers to G's q queries, plus one needs ℓ bits to encode the original seed of G.

Theorem 3 *Let $G^{(\cdot)} : \{0,1\}^\ell \to \{0,1\}^{\ell+k}$ be an (S_p, S_g, ε)-PRG construction that makes q queries to an oracle $\pi \in \Pi_n$, and assume (without loss of generality) $\varepsilon > 2^{-S_p}$. If $q < k/(5\log S_p)$ then there exists an $(S_g, 2\varepsilon)$-secure PRG $G' : \{0,1\}^{\ell'} \to \{0,1\}^{\ell+k}$ (with $\ell' < \ell + k$) without access to a permutation oracle.*

Proof Since $G^{(\cdot)}$ is an (S_p, S_g, ε)-PRG construction, this means that if $\pi : \{0,1\}^n \to \{0,1\}^n$ is S_p-hard then for any statistical test (distinguisher) T of size $\le S_g$ we have

$$\left| \mathop{\mathbf{Pr}}_{x \in U_{\ell+k}}[T^\pi(x) = 1] - \mathop{\mathbf{Pr}}_{s \in U_\ell}[T^\pi(G^\pi(s)) = 1] \right| \le \varepsilon.$$

Let $t = 5\log S_p$. From Corollary 1 we know that a random permutation $\pi \in \Pi_{t,n}$ is S_p-hard with probability greater than $1 - 2^{-2^{t/2}}$. An averaging argument thus shows that for any circuit T of size $\le S_g$ we have

$$\left| \Pr_{x \in U_{\ell+k}}[T(x) = 1] - \Pr_{\substack{\pi \in \Pi_{t,n} \\ s \in U_\ell}}[T(G^\pi(s)) = 1] \right| < \varepsilon + 2^{-2^{t/2}} \le 2\varepsilon. \tag{4}$$

(Note that eliminating T's access to π only makes T less powerful.)

Recall that any $\pi \in \Pi_{t,n}$ operates only on its first t input bits; i.e., any such π satisfies $\pi(a,b) = (\hat{\pi}(a), b)$ where $\hat{\pi}$ is a permutation over $\{0,1\}^t$. Without loss of generality, we now assume that G always queries π with strings having distinct t-prefixes. Indeed, for any G asking arbitrary queries, one can construct a $\widehat{G}$ with essentially the same running time, such that if G asks (a,b) with a equal to the t-prefix of a previous query, $\widehat{G}$ simulates the answer without querying π using the previously-obtained value of $\hat{\pi}(a)$. In general the behavior of $\widehat{G}$ is different than that of G, but when we restrict to $\pi \in \Pi_{t,n}$ they are equivalent.

By assumption, G queries π at most $q < k/t$ times. We now construct G' which takes as input a seed s' of length $\ell' \overset{\text{def}}{=} \ell + \log\left(\Pi_{i=0}^{q-1}(2^t - i)\right) < \ell + qt < \ell + k$. The first ℓ bits of s' are used by G' to define a value s, and the remaining bits are used to select q distinct elements $y_1, \ldots, y_q \in \{0,1\}^t$. We then define

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

where the notation $G^{y_1,\ldots,y_q}(s)$ denotes the computation of $G^{(\cdot)}$ on input s when its i^{th} oracle query $x_i = (a_i, b_i)$ (with $|a_i| = t$) is answered with (y_i, b). (Here we use the fact that the t-prefixes of G's queries are distinct.)

G' stretches its input by at least one bit, and requires no oracle access. Furthermore,

$$\{G'(s')\}_{s' \in \{0,1\}^{\ell'}} \equiv \{G^\pi(s)\}_{\pi \in \Pi_{t,n},\, s \in U_\ell}.$$

Equation (4) now immediately implies that G' is an $(S_g, 2\varepsilon)$-secure PRG. ■

3.2 Universal One-Way Hash Functions

Let $H^{(\cdot)} : \{0,1\}^r \times \{0,1\}^{\ell+k} \to \{0,1\}^\ell$ be a UOWHF construction compressing $\ell + k$-bit inputs to ℓ-bit outputs, which expects as an oracle a permutation $\pi \in \Pi_n$. We show that if the construction makes $q < k/(g \log S)$ queries to π (where S is the security of the permutation), then it is possible to derive an unconditionally-secure construction of a UOWHF (i.e., without any access to π).

As in the case of PRGs, we first observe that H is secure when π is chosen uniformly from $\Pi_{t,n}$, for $t = 5 \log S$. We then show that the computation of H can be simulated by an H', without oracle access to π, by including as part of the key the t-prefixes of the answers for the q queries H makes to π. Furthermore, we include in the output of H' the t-prefixes of the q queries themselves. (Recall that when $\pi \in \Pi_{t,n}$, only the t-prefixes of the queries and answers are relevant.) Note that as long as $\ell + qt < \ell + k$, the new function H' is still length-decreasing. The crux of the proof is then to show that if an adversary can find a collision in H', then there exists an adversary that can find a collision in H (which is impossible by assumption). Hence, H' is an unconditionally-secure UOWHF.

Theorem 4 *Let $H^{(\cdot)} : \{0,1\}^r \times \{0,1\}^{\ell+k} \to \{0,1\}^\ell$ be an (S_p, S_h, ε)-UOWHF construction that makes q queries to an oracle $\pi \in \Pi_n$, and assume (without loss of generality) that $\varepsilon > 2^{-S_p}$. If $q < k/(5 \log S_p)$ then there exists an $(S_h - S_H, 2\varepsilon)$-secure UOWHF $H' : \{0,1\}^{r'} \times \{0,1\}^{\ell+k} \to \{0,1\}^{\ell'}$ (with $\ell' < \ell + k$) without access to a permutation oracle, where S_H is the size of the circuit computing H.*

Proof Since $H^{(\cdot)}$ is an (S_p, S_h, ε)-UOWHF construction, this means that if $\pi \in \Pi_n$ is S_p-hard then any circuit A of size $\leq S_h$ finds a collision with probability at most ε; that is

$$\Pr_{\substack{s \in U_r \\ z \in U_{\ell+k}}}\left[A^\pi(s, z, H^\pi(s,z)) = z' : z' \neq z \bigwedge H^\pi(s, z') = H^\pi(s, z)\right] \leq \varepsilon.$$

We say that z' in the above experiment is a *strong collision* if $z' \neq z$, $H^\pi(s, z') = H^\pi(s, z)$, and furthermore the t-prefixes of the q queries made during the computation of $H^\pi(s, z')$

are the same (and occur in the same order) as the t-prefixes of the q queries made during the computation of $H^\pi(s,z)$. Clearly, if π is S_p-hard and A has size at most S_h we have

$$\Pr_{\substack{s\in U_r\\ z\in U_{\ell+k}}}\left[A^\pi(s,z,H^\pi(s,z))=z' : z' \text{ is a strong collision}\right]\leq\varepsilon.$$

We now restrict the class of adversaries A under consideration: we consider adversaries that do not access π arbitrarily, but are instead simply given the t-prefixes of the queries and answers made during the computation of $H^\pi(s,z)$. Since such restricted adversaries can be simulated by general adversaries with overhead S_H (recall that this is the size of the circuit computing H), we have that if π is S_p-hard and A is a circuit of size $\leq S_h - S_H$ then

$$\Pr_{\substack{s\in U_r\\ z\in U_{\ell+k}}}\left[A(s,z,H^\pi(s,z),x_1,\ldots,x_q,y_1,\ldots,y_q)=z' : z' \text{ is a strong collision}\right]\leq\varepsilon,$$

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

Let $t=5\log S_p$. From Corollary 1 we know that a random permutation $\pi\in\Pi_{t,n}$ is S_p-hard with probability greater than $1-2^{-2^{t/2}}>1-\varepsilon$. An averaging argument thus shows that for any circuit A of size $\leq S_h-S_H$ we have

$$\Pr_{\substack{\pi\in\Pi_{t,n}\\ s\in U_r,\, z\in U_{\ell+k}}}\left[A(s,z,H^\pi(s,z),x_1,\ldots,x_q,y_1,\ldots,y_q)=z' : z' \text{ is a strong collision}\right]<2\varepsilon. \quad (5)$$

As in the proof of Theorem 3, we may assume without loss of generality that H queries π on points with distinct t-prefixes. Consider the function $H' : \{0,1\}^{r'}\times\{0,1\}^{\ell+k}\to\{0,1\}^{\ell'}$ defined as follows, where[6] $r'\leq r+qt$ and $\ell'=\ell+qt$:

$$H'\left((s,y_1,\ldots,y_q),z\right)\overset{\text{def}}{=}H^{y_1,\ldots,y_q}(s,z),x_1,\ldots,x_q,$$

where by $H^{y_1,\ldots,y_q}(s,z)$ we mean the computation of $H^{(\cdot)}(s,z)$ when its i^{th} oracle query (a_i,b_i) (with $|a_i|=t$) is answered with (y_i,b_i), and where $x_i,\ldots,x_q$ denote the t-prefixes of the q oracle queries made (i.e., $x_i=a_i$).

Note that any collision in H' gives a strong collision for H; this observation and Equation (5) show that for any adversary A of size $\leq S_h-S_H$ we have

$$\Pr_{\substack{s'\in U_{r'}\\ z\in U_{\ell+k}}}\left[A(s',z,H'(s',z))=z' : z'\neq z\bigwedge H'(s',z')=H'(s',z)\right]<2\varepsilon.$$

Furthermore, if $q<k/t$ then $\ell'<\ell+k$, so $H'(s',\cdot)$ is a length-decreasing function. Since H' does not invoke any oracle, this completes the proof. ■

[6] As in the proof of Theorem 3, at most qt bits are needed to specify q distinct elements in $\{0,1\}^t$.

3.3 Public-Key Encryption

Let $\mathcal{PKE}^{(\cdot)} = (\mathsf{Gen}^{(\cdot)}, \mathsf{Enc}^{(\cdot)}, \mathsf{Dec}^{(\cdot)})$ be an (S_p, S_e, ε)-PKE construction for messages of length m which expects to access an oracle $\tau \in T_n$. We prove that unless Enc^{τ} queries τ at least $\Omega(m/\log S_p)$ times, there exists a one-way function which does not require any oracle access. Our proof proceeds by showing that unless Enc^{τ} makes at least $\Omega(m/\log S_{tp})$ queries to τ, we can explicitly construct an interactive, private-key encryption scheme $(\mathsf{Enc}', \mathsf{Dec}')$ requiring no access to the oracle and in which the encrypted message is longer than the shared key. Using a previous result of Impagliazzo and Luby [30] (see also Lemma 1), this implies the existence of a one-way function.

As in the previous proofs, we first observe that a random $\tau \in T_{t,n}$ is S_p-hard with all but negligible probability when $t = 6 \log S_p$ (cf. Corollary 2). To construct an interactive, private-key encryption scheme without access to an oracle, we would like to have the parties "simulate" a random τ by appropriately choosing random t-prefixes for the answers to their queries, as needed. However, due to the interactive nature of the task, a problem arises. The bits to simulate τ cannot be included in the private key, since the encryption and decryption algorithms may make their queries in different order and, indeed, may make different queries altogether! An additional problem is that the queries made by these algorithms may depend on the message being encrypted, and thus may not be known in advance at the time the key is shared.

A second possibility that comes to mind is to have each party include, along with its message to the other party, a list of t-prefixes for the queries and answers generated thus far in accessing τ. In this case, however, privacy is no longer guaranteed as the queries may reveal information about the message. But this may easily be remedied in the private-key setting: the parties simply share a sufficiently-long one-time pad in advance and then "encrypt" their queries and answers using this pad.

Let q_g be the number of queries made to τ by Gen, and let q_e be the number of queries made by Enc. The private-key encryption scheme outlined above requires a shared key of size roughly $2t \cdot (q_g + q_e))$ to encrypt an m-bit message. Recalling the result of Impagliazzo and Luby [30], if $q_g + q_e < m/2t$ then the key is shorter than the message and a one-way function exists. This already gives a "weak" lower bound. To obtain the better lower bound $q_e < m/t$ (so that we bound the efficiency of encryption alone), additional work is needed; details are given in the proof of Theorem 5.

We begin by showing that the existence of a private-key encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ which securely encrypts messages longer than the shared key implies the existence of a one-way function. Although this result is already known [30] (without the concrete bounds given below), we give a much simpler and more direct proof. We stress that the result applies even to an interactive encryption protocol.

Lemma 1 *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be an* (S, δ)*-secure private-key encryption scheme for messages of length* m *using a shared key of length* $k < m$. *Let* S_e *be the size of the circuit for* Enc. *Then for any* $\ell \in \mathbb{N}$ *there exists a function* f *which is* $(S - 2\ell S_e, \ell\delta + 2^{-\ell(m-k)})$*-one way.*

Proof We prove the theorem for non-interactive encryption schemes; extension to the case of interactive encryption is straightforward. First note that, via standard hybrid argument, running ℓ independent copies of $(\mathsf{Enc}, \mathsf{Dec})$ yields an $(S - \ell S_e, \ell\delta)$-secure private-key encryption scheme for messages of length ℓm in which the shared key has length ℓk. Let $\mathcal{SKE}_\ell = (\mathsf{Enc}_\ell, \mathsf{Dec}_\ell)$ denote this modified scheme.

Define f by $f(sk, M, \omega) = \mathsf{Enc}_\ell(sk, M; \omega) \| M$, where ω represents the random coins used by Enc_ℓ. We claim that this function is (S', δ')-one way, where $S' = S - 2\ell S_e$ and $\delta' = \ell\delta + 2^{-\ell(m-k)}$. Assume the contrary. Then there is an algorithm B of size at most S' for which $\mathsf{Succ}_{B,f} > \delta'$, where

$$\mathsf{Succ}_{B,f} \overset{\text{def}}{=} \Pr\left[sk \leftarrow \{0,1\}^{\ell k}; M \leftarrow \{0,1\}^{\ell m}; C \leftarrow \mathsf{Enc}_\ell(sk, M) : B(C\|M) \in f^{-1}(C\|M)\right].$$

We use B to construct an algorithm A of size at most $S - \ell S_e$ for which $\mathsf{Succ}_{A,\mathcal{SKE}_\ell} > \ell\delta$, where

$$\mathsf{Succ}_{A,\mathcal{SKE}_\ell} \overset{\text{def}}{=} \left| \Pr_{\substack{M_0, M_1 \in \{0,1\}^{\ell m} \\ C \in \mathcal{SKE}_\ell(M_0)}}[A(M_0, M_1, C) = 1] - \Pr_{\substack{M_0, M_1 \in \{0,1\}^{\ell m} \\ C \in \mathcal{SKE}_\ell(M_1)}}[A(M_0, M_1, C) = 1] \right|.$$

This implies that there exist two messages M_0, M_1 for which A can distinguish encryptions of M_0 from encryptions of M_1 with probability better than $\ell\delta$, contradicting the assumed security of $(\mathsf{Enc}_\ell, \mathsf{Dec}_\ell)$. Thus, we are done once we have demonstrated such an A.

Define A as follows: on input (M_0, M_1, C), algorithm A runs $B(C\|M_0)$ to obtain the result $sk'\|M'\|\omega'$. It then checks whether $f(sk', M', \omega') \overset{?}{=} C\|M_0$. If so (i.e., B has succeeded in inverting f), then A outputs 1. Otherwise, A outputs 0. Note that $|A| = |B| + \ell S_e \leq S - \ell S_e$, as required.

First note that

$$\Pr_{\substack{M_0, M_1 \in \{0,1\}^{\ell m} \\ C \in \mathcal{SKE}_\ell(M_0)}}[A(M_0, M_1, C) = 1] = \mathsf{Succ}_{B,f} > \delta'.$$

Secondly, we have

$$\begin{aligned}
&\Pr_{\substack{M_0, M_1 \in \{0,1\}^{\ell m} \\ C \in \mathcal{SKE}_\ell(M_1)}}[A(M_0, M_1, C) = 1] \\
&\qquad \leq \Pr[sk \leftarrow \{0,1\}^{\ell k}; M_0, M_1 \leftarrow \{0,1\}^{\ell m}; \\
&\qquad\qquad C \leftarrow \mathsf{Enc}_\ell(sk, M_1) : \exists sk' \text{ s.t. } \mathsf{Dec}_\ell(sk', C) = M_0] \\
&\qquad \leq \sum_{sk' \in \{0,1\}^{\ell k}} \Pr[sk \leftarrow \{0,1\}^{\ell k}; M_0, M_1 \leftarrow \{0,1\}^{\ell m}; \\
&\qquad\qquad C \leftarrow \mathsf{Enc}_\ell(sk, M_1) : \mathsf{Dec}_\ell(sk', C) = M_0] \\
&\qquad \leq \sum_{sk' \in \{0,1\}^{\ell k}} 2^{-\ell m} \\
&\qquad = 2^{\ell(k-m)}.
\end{aligned}$$

(The third inequality is due to the fact that the distribution on M_0 is independent of $\mathsf{Dec}_\ell(sk', C)$, and hence the probability that these are equal is at most $2^{-\ell m}$.) But then

$$\mathsf{Succ}_{A,\mathcal{SKE}_\ell} > \delta' - 2^{\ell(k-m)} \geq \ell\delta,$$

as desired. ■

Our main result of this section follows.

Theorem 5 *Let $\mathcal{PKE}^{(\cdot)}$ be an (S_p, S_e, ε)-PKE construction for messages of length m, and let $t = 6 \log S_p$. Assume Gen makes q_g queries to an oracle $\tau \in T_n$ and Enc makes q_e queries to τ; set $\ell = 2 \cdot \lceil 5tq_g/(m - 5tq_e) \rceil$. Assume further that $\varepsilon < (1/4 - 2^{1-S_p})/\ell$. If $q_e < m/5t$, then there exists an $(S_e - 3\ell S_{\mathsf{Enc}}, 3/4)$-one-way function (without access to any oracle), where S_{Enc} is the size of the circuit for Enc.*

Proof Note that we did not try to optimize the constants in the proof or the required bound on ε. In applications of cryptographic interest, S_p and S_e are super-polynomial, S_{Enc}, q_g, q_e, and m are (small) polynomials, and ε is negligible; thus, $\varepsilon \ll (1/4 - 2^{1-S_p})/\ell$ and $S_e \gg 3\ell S_{\mathsf{Enc}}$ anyway.

Let $\mathcal{PKE}^{(\cdot)} = (\mathsf{Gen}^{(\cdot)}, \mathsf{Enc}^{(\cdot)}, \mathsf{Dec}^{(\cdot)})$. As in the proof of Lemma 1, for any $\ell \in \mathbb{N}$ we may construct a public-key encryption scheme $\mathcal{PKE}_\ell^{(\cdot)} = (\mathsf{Gen}_\ell^{(\cdot)}, \mathsf{Enc}_\ell^{(\cdot)}, \mathsf{Dec}_\ell^{(\cdot)})$ for ℓm-bit messages in the natural way; furthermore, we may set $\mathsf{Gen}_\ell = \mathsf{Gen}$ since we are now in the public-key setting and so key generation need only be done once. It is easy to see (via hybrid argument) that $\mathcal{PKE}_\ell^{(\cdot)}$ is an $(S_p, S_e - \ell S_{\mathsf{Enc}}, \ell\varepsilon)$-PKE construction, where the number of queries made by Gen_ℓ is q_g and the number of queries made by Enc_ℓ is at most ℓq_e.

Set $\ell = 2 \cdot \lceil 5t\, q_g/(m - 5t\, q_e) \rceil$ as in the statement of the theorem, and let $S' = S_e - \ell S_{\mathsf{Enc}}$ and $\delta = \ell\varepsilon + 2^{1-S_p}$. We use $\mathcal{PKE}_\ell^{(\cdot)}$ to construct an (S', δ)-secure interactive, private-key encryption scheme $\mathcal{SKE} = (\mathsf{Enc}', \mathsf{Dec}')$ for ℓm-bit messages in which the shared key will have length $5t \cdot (q_g + \ell q_e)$. Furthermore, $\mathcal{SKE}$ will require no access to the trapdoor permutation oracle. Finally, we have $5t \cdot (q_g + \ell q_e) < \ell m$ (in fact, $\ell m - 5t \cdot (q_g + \ell q_e) \geq 1$) and $\delta < 1/4$; thus, application of Lemma 1 (with $\ell = 1$ there) yields the desired result.

Since $\mathcal{PKE}_\ell^{(\cdot)}$ is an $(S_p, S', \ell\varepsilon)$-PKE construction, if τ is S_p-hard then, for any circuit B of size $\leq S'$ and for any messages $M_0, M_1 \in \{0,1\}^{\ell m}$ we have

$$\left| \Pr_{v \in \mathcal{PKE}_\ell^\tau(M_0)}[B^\tau(v) = 1] - \Pr_{v \in \mathcal{PKE}_\ell^\tau(M_1)}[B^\tau(v) = 1] \right| \leq \ell\varepsilon.$$

Corollary 2 shows that a random $\tau \in T_{t,n}$ is S_p-hard except with probability less than 2^{1-S_p}. A straightforward averaging argument thus shows that for any circuit B of size $\leq S'$ and for any messages $M_0, M_1 \in \{0,1\}^{\ell m}$ we have

$$\left| \Pr_{\substack{\tau \in T_{t,n} \\ v \in \mathcal{PKE}_\ell^\tau(M_0)}}[B(v) = 1] - \Pr_{\substack{\tau \in T_{t,n} \\ v \in \mathcal{PKE}_\ell^\tau(M_1)}}[B(v) = 1] \right| < \ell\varepsilon + 2^{1-S_p} = \delta. \tag{6}$$

(Removing B's access to τ only makes B weaker.)

As mentioned in the discussion at the beginning of this section, our private-key encryption scheme $\mathcal{SKE}$ will need to "simulate" a random $\tau \in T_{t,n}$ for algorithms $\mathsf{Gen}, \mathsf{Enc}_\ell$, and Dec_ℓ. The simulation procedure $\mathcal{SIM}$ will ensure consistency of the answers to all oracle queries. It does so by storing t-prefixes of all previous oracle queries and answers in a list L. Before answering the current query, $\mathcal{SIM}$ ensures that the current answer will not generate any inconsistencies. For example, if query $td\|b$ to G was answered by $k\|b$ (where $|td| = |k| = t$), then subsequent query $td'\|b'$ to G must be answered by $k'\|b'$ where $k' = k$ iff $td' = td$. A more involved procedure is needed to answer queries to F and F^{-1}. We now describe the details of this simulation.

$\mathcal{SIM}(L)$:

- On query $G(td\|b)$ (where $|td| = t$): if $\exists k$ s.t. $(td, k) \in L$, return $k\|b$. Otherwise pick random $k \in \{0,1\}^t$ such that $\forall td' : (td', k) \notin L$, return $k\|b$, and store (tk, k) in L.

- On query $F(k\|b, x\|b')$ (where $|k| = |x| = t$):

  1. if $\exists y$ s.t. $(k, x, y) \in L$, return $y\|b'$.
  2. Otherwise, if $\exists td$ s.t. $(td, k) \in L$, choose random $y \in \{0,1\}^t$ such that $\forall x' : (k, x', y) \notin L$, return $y\|b'$, and store (k, x, y) in L.
  3. Otherwise, choose random $td \in \{0,1\}^t$ such that $\forall k' : (td, k') \notin L$, choose random $y \in \{0,1\}^t$, return $y\|b'$, and store (k, td) and (k, x, y) in L.

- On query $F^{-1}(td\|b, y\|b')$ (where $|tk| = |y| = t$):

  1. if $\exists k, x$ s.t. $(td, k), (k, x, y) \in L$, return $x\|b'$
  2. Otherwise, if $\exists k$ s.t. $(td, k) \in L$, choose random $x \in \{0,1\}^t$ such that $\forall y' : (k, x, y') \notin L$, return $x\|b'$, and store (k, x, y) in L
  3. Otherwise, choose random $k \in \{0,1\}^t$ such that $\forall td' : (td', k) \notin L$, choose random $x \in \{0,1\}^t$, return $x\|b'$, and store (td, k) and (k, x, y) in L

Note that each time a query is answered, at most $5t$ bits are stored in L.

Construct $\mathcal{SKE}$ as follows. Parse the shared key s as (s_1, s_2) where $|s_1| = 5tq_g$ and $|s_2| = 5t\ell q_e$. To encrypt message M, the receiver Dec' begins by initializing list $L := \emptyset$. The receiver then computes $(pk, sk) \leftarrow \mathsf{Gen}^{\mathcal{SIM}(L)}$ and sends $pk, s_1 \oplus L$ to the sender. The receiver stores sk, L for later use. Upon receiving the first message $pk, \hat{s}_1$, the sender computes $L_1 := s_1 \oplus \hat{s}_1$ and sets $L := L_1$. The sender then computes $C \leftarrow \mathsf{Enc}_\ell^{\mathcal{SIM}(L)}(pk, M)$ and sets $L_2 := L \backslash L_1$. Finally, Enc' sends $C, s_2 \oplus L_2$ to the receiver. Upon receiving message $C, \hat{s}_2$, the receiver decrypts by setting $L_2 := s_2 \oplus \hat{s}_2$ and $L := L \cup L_2$ (here, L is the list stored by the receiver from the first stage). The receiver can then compute $M := \mathsf{Dec}_\ell^{\mathcal{SIM}(L)}(sk, C)$.

It is clear that $\mathcal{SKE}$ has correct decryption. We now show that the scheme is (S', δ)-secure. Assume toward a contradiction that there exists a circuit A of size $\leq S'$ and messages $M_0, M_1 \in \{0,1\}^{\ell m}$ such that

$$\left| \Pr_{v \in \mathcal{SKE}(M_0)}[A(v) = 1] - \Pr_{v \in \mathcal{SKE}(M_1)}[A(v) = 1] \right| > \delta.$$

We construct circuit B attacking $\mathcal{PKE}_\ell$ as follows. On input pk, C, circuit B picks random strings $\hat{s}_1, \hat{s}_2$ where $|\hat{s}_1| = 5tq_g$ and $|\hat{s}_2| = 5t\ell q_e$. Then, B outputs $A(pk, \hat{s}_1, C, \hat{s}_2)$. Because the keys s_1, s_2 of $\mathcal{SKE}$ are used as a one-time pad, it is easy to see that, for $b \in \{0,1\}$:

$$\Pr_{\substack{\tau \in T_{t,n} \\ (pk, C) \in \mathcal{PKE}_\ell^\tau(M_b)}}[B(pk, C) = 1] = \Pr_{v \in \mathcal{SKE}(M_b)}[A(v) = 1].$$

Thus, the advantage of B is equal to the advantage of A (which is greater than δ), contradicting Equation (6). ■

3.4 Private-Key Encryption

The techniques of the previous section can be adapted to show a similar lower bound for private-key encryption schemes based on one-way permutations (this can also be extended for schemes based on trapdoor permutations; however, since one-way permutations are sufficient for secure private-key encryption, we focus on this case). Specifically, let $\mathcal{SKE}^{(\cdot)} = (\mathsf{Enc}^{(\cdot)}, \mathsf{Dec}^{(\cdot)})$ be a private-key encryption scheme for m-bit messages in which the shared key is of length k, and suppose $\mathcal{SKE}^{\pi}$ is secure whenever π is S_p-hard. We show that unless Enc^{π} queries π at least $q_e = \Omega(\frac{m-k}{\log S})$ times, then an unconditional one-way function exists. This matches the known upper bound.

The proof is similar to that of Theorem 5, in that we convert $\mathcal{SKE}^{(\cdot)}$ to a private-key encryption scheme $\mathcal{SKE}'$ that does not access the oracle at all. The only difference between the proof here and the proof of Theorem 5 is that here the parties need to share a k-bit key in addition to the one-time pad used to encrypt their "simulated" queries and answers to the oracle. Set $t = 5 \log S_p$. The resulting $\mathcal{SKE}'$ requires a shared key of length $k + 2tq_e$ and encrypts an m-bit message. As before, then, if $k + 2tq_e < m$ we obtain a private-key encryption scheme (making no oracle queries) in which the message is longer than the key. Furthermore (cf. Lemma 1), this implies the existence of a one-way function.

Theorem 6 *Let* $(\mathsf{Enc}^{(\cdot)}, \mathsf{Dec}^{(\cdot)})$ *be an* (S_p, S_e, ε)*-SKE construction for messages of length* m *using a key of length* k *in which* Enc *makes* q *queries to an oracle* $\pi \in \Pi_n$*, and assume (without loss of generality) that* $\varepsilon > 2^{-S_p}$*. If* $q < \frac{m-k}{10 \log S_p}$ *then there exists an* $(S_e, 2\varepsilon)$*-secure private-key encryption scheme (in which the message is longer than the key) without access to a permutation oracle.*

3.5 Signature Schemes

We now demonstrate a lower bound on the efficiency of signature verification for any signature scheme based on one-way permutations. Let $\mathcal{SIG}^{(\cdot)} = (\mathsf{Gen}^{(\cdot)}, \mathsf{Sign}^{(\cdot)}, \mathsf{Vrfy}^{(\cdot)})$ be a signature scheme for messages of length m that expects as an oracle a permutation $\pi \in \Pi_n$, and suppose that $\mathcal{SIG}^{\pi}$ is secure whenever π is S_p-hard. We prove that unless Vrfy queries π at least $\Omega(m/\log S_p)$ times, then it is possible to construct from $\mathcal{SIG}$ a one-way function which does not access any oracle. Note that this one-way function could then be used to construct an secure signature scheme (which requires no oracle access) [38].

We start with an informal overview of our proof technique. As a first attempt to construct a one-way function from the verification algorithm, one might define

$$F_1(PK, M, \sigma) = PK \| \mathsf{Vrfy}^{(\cdot)}(PK, M, \sigma).$$

Intuitively, this function is difficult to invert on elements of the form $PK\|1$ if PK is a valid and randomly-generated public key, since inverting the function on points of this form is equivalent to signature forgery. As presently defined, however, evaluating F_1 requires calls to π; however, our goal is to construct a function which does not require access to any oracle. As in the previous section, though, one may observe that π is S_p-hard (and thus $\mathcal{SIG}$ is secure) when π is uniformly chosen from $\Pi_{t,n}$ for $t = 5 \log S_p$ (cf. Corollary 1). So, if Vrfy makes q queries to π, then specifying qt bits as the answers to these queries removes any need to query the oracle. Based on this, one might consider the function

$$F_2(PK, y_1, \ldots, y_q, M, \sigma) = PK \| y_1 \| \cdots \| y_q \| x_1 \| \cdots \| x_q \| \mathsf{Vrfy}^{y_1,\ldots,y_q}(PK, M, \sigma), \tag{7}$$

where $|y_1| = \cdots = |y_q| = |x_1| = \cdots = |x_q| = t$ and the i^{th} query $x_i \| b_i$ of Vrfy is answered with $y_i \| b_i$ (we may assume without loss of generality that Vrfy queries π with strings having distinct t-prefixes; furthermore, the $\{y_i\}$ may actually be specified using only $\log \binom{2^t}{q}$ bits). The intuition, as before, is that F_2 is difficult to invert on elements of the form $PK \| \vec{y} \| \vec{x} \| 1$ if $PK, \vec{y}$, and $\vec{x}$ are chosen appropriately.

Now, however, another problem arises. In order for F_2 to qualify as a one-way function, it must be hard to invert $F_2(PK, \vec{y}, M, \sigma)$ when $PK, \vec{y}, M, \sigma$ are sampled from an *efficiently sampleable* distribution (cf. Lemma 2, below). More specifically, a proof of one-wayness will need to show how to efficiently sample $PK, \vec{y}, M, \sigma$ such that inverting the value $F_2(PK, \vec{y}, M, \sigma)$ results in a signature forgery and hence a contradiction. A necessary condition for inversion to result in signature forgery is that $\mathsf{Vrfy}^{\vec{y}}(PK, M, \sigma) = 1$. Generating $PK, \vec{y}, M, \sigma$ such that this holds is easy if we have the secret key SK; but then inverting F_2 and forging a signature does not yield the desired contradiction!

Instead, in the proof we will obtain M, σ from the signer. In this case, however, inverting $F_2(PK, \vec{y}, M, \sigma)$ does not result in a forgery if it results in the same message/signature pair M, σ. We come now to the crux of our proof. If the number of queries is "small", we show that inversion of F_2 yields a different message M' (and thus a successful forgery) with noticeable probability. More precisely, for randomly-generated $PK, \vec{y}, \vec{x}, \sigma$, let

$$Y = PK \| \vec{y} \| \vec{x} \| 1 = F_2(PK, \vec{y}, M, \sigma).$$

If $|\vec{x}| = \sum_{i=1}^{q} |x_i| < |M|$, then (on average) there exists an element $PK \| \vec{y} \| M' \| \sigma' \in F_2^{-1}(Y)$ with $M' \neq M$; hence inverting Y results in a forgery with noticeable probability. This idea is formalized in the proof of the following theorem.

Theorem 7 *Let $\mathcal{SIG}^{(\cdot)}$ be an $(S_p, S_\Sigma, \varepsilon)$-signature construction for messages of length m in which* Vrfy *makes q queries to an oracle $\pi \in \Pi_n$ and $\varepsilon < 1/4 - 2^{-S_p}$. If $q < m/(5 \log S_p)$ then there exists an $(S_\Sigma - S_{\mathsf{Gen}} - S_{\mathsf{Sign}} - 2S_{\mathsf{Vrfy}}, 3/4)$-one-way function (without access to a permutation oracle), where S_{Gen}, S_{Sign},and S_{Vrfy} are the sizes of the circuits for* Gen, Sign, *and* Vrfy*, respectively.*

Proof As in Theorem 5, we did not try to optimize the constants in the proof or the required bound on ε. In applications of cryptographic interest, one will anyway have $S_\Sigma \gg S_{\mathsf{Gen}} + S_{\mathsf{Sign}} + 2S_{\mathsf{Vrfy}}$ and $\varepsilon \ll 1/4 - 2^{-S_p}$.

We first present a technical lemma showing that the existence of a function which is one-way over an efficiently sampleable domain implies the existence of a function which is one-way under the definition of Section 2.1.

Lemma 2 *Let $\mathcal{D}$ be a distribution computable by a circuit of size $S_{\mathcal{D}}$ and let f be a function such that for every circuit A of size $\leq S$ we have:*

$$\Pr[x \leftarrow \mathcal{D} : A(f(x)) \in f^{-1}(f(x))] \leq \delta.$$

Then there exists a function $\hat{f}$ that is $(S - S_{\mathcal{D}}, \delta)$-one way.

Proof (of lemma) Define $\hat{f}(r) \stackrel{\text{def}}{=} f(\mathcal{D}(r))$. We claim that $\hat{f}$ is $(S - S_{\mathcal{D}}, \delta)$-one way. Assume the contrary. Then there exists a circuit $\hat{A}$ of size at most $S - S_{\mathcal{D}}$ for which

$$\Pr_r[\hat{A}(\hat{f}(r)) \in \hat{f}^{-1}(\hat{f}(r))] > \delta.$$

Define circuit A by $A(y) \stackrel{\text{def}}{=} \mathcal{D}(\hat{A}(y))$. Notice that $|A| \leq S$. Furthermore,

$$\begin{aligned}
&\Pr[x \leftarrow \mathcal{D} : A(f(x)) \in f^{-1}(f(x))] \\
&\qquad = \Pr_r[x := \mathcal{D}(r) : f(A(f(x))) = f(x)] \\
&\qquad = \Pr_r[f(A(\hat{f}(r))) = \hat{f}(r)] \\
&\qquad = \Pr_r[\hat{f}(\hat{A}(\hat{f}(r))) = \hat{f}(r)] \\
&\qquad = \Pr_r[\hat{A}(\hat{f}(r)) \in \hat{f}^{-1}(\hat{f}(r))] \ > \ \delta,
\end{aligned}$$

a contradiction. □

The proof of the theorem proceeds by using $\mathcal{SIG}$ to construct a function F along with a distribution $\mathcal{D}$ such that for every circuit A of size $\leq S_{\Sigma} - S_{\mathsf{Vrfy}}$ we have:

$$\Pr[X \leftarrow \mathcal{D} : A(F(X)) \in F^{-1}(F(X))] \leq \varepsilon + 2^{-S_p} + 1/2 < 3/4. \tag{8}$$

Furthermore, $\mathcal{D}$ will be computable by a circuit of size $S_{\mathsf{Gen}} + S_{\mathsf{Sign}} + S_{\mathsf{Vrfy}}$. Applying Lemma 2 then yields the desired result.

Since $\mathcal{SIG}^{(\cdot)}$ is an $(S_p, S_{\Sigma}, \varepsilon)$-signature construction, if π is S_p-hard then, for any circuit B of size $\leq S_{\Sigma}$ we have $\mathsf{Succ}_{\pi,B} \leq \varepsilon$ where

$$\begin{aligned}
&\mathsf{Succ}_{\pi,B} \stackrel{\text{def}}{=} \\
&\quad \Pr\left[(PK, SK) \leftarrow \mathsf{Gen}^{\pi}; M \leftarrow \{0,1\}^m;\ \sigma \leftarrow \mathsf{Sign}^{\pi}(SK, M); (M', \sigma') := B^{\pi}(PK, M, \sigma) :\right. \\
&\qquad \left. \mathsf{Vrfy}^{\pi}(PK, M', \sigma') = 1 \wedge M' \neq M\right].
\end{aligned}$$

Let $t = 5 \log S_p$. Corollary 1 shows that a random $\pi \in \Pi_{t,n}$ is S_p-hard except with probability less than 2^{-S_p}. An averaging argument then implies that for any circuit B of size $\leq S_{\Sigma}$ we have $\mathsf{Succ}^*_B < \varepsilon + 2^{-S_p}$ where Succ^*_B is defined analogously to $\mathsf{Succ}_{\pi,B}$ except that the probability is now taken over random choice of $\pi \in \Pi_{t,n}$ as well.

Define a function F as in Equation (7), repeated here for convenience:

$$F(PK, \vec{y}, M, \sigma) = PK \| \vec{y} \| \vec{x} \| \mathsf{Vrfy}^{\vec{y}}(PK, M, \sigma),$$

where $\vec{y} = (y_1, \ldots, y_q)$, $\vec{x} = (x_1, \ldots, x_q)$, $|y_1| = \cdots = |y_q| = |x_1| = \cdots = |x_q| = t$, and the i^{th} query $x_i \| b_i$ of Vrfy is answered with $y_i \| b_i$. (As in the proof of Theorem 3, we now assume that Vrfy queries its oracle on points having distinct t-prefixes.) We also define distribution $\mathcal{D}$ by the following experiment which depends on uniformly distributed coins $r_g, r_s, r_y \in \{0,1\}^*$, and $M \in \{0,1\}^m$:

$$\left\{ \begin{array}{c} (PK, SK) := \mathsf{Gen}(r_g); \\ \sigma := \mathsf{Sign}(SK, M; r_s); \mathsf{Vrfy}(PK, M, \sigma) \end{array} \ : \ PK \| \vec{y} \| M \| \sigma \right\}.$$

In the above experiment, the coins r_y are used to give a simulation[7] of a random $\pi \in \Pi_{t,n}$ which is consistent across the executions of Gen, Sign, and Vrfy. The component y_i of $\vec{y}$ is the t-prefix of the answer given in response to the i^{th} query of Vrfy. Note that $\mathcal{D}$ is computable by a circuit of size $S_{\mathsf{Gen}} + S_{\mathsf{Sign}} + S_{\mathsf{Vrfy}}$.

We claim that F satisfies the requirement expressed in Equation 8. Assume toward a contradiction that there exists a circuit A of size $\leq S_\Sigma - S_{\mathsf{Vrfy}}$ for which Equation 8 does not hold. We use A to construct an algorithm B that, given PK, a random message M, and a signature σ on M, forges a valid signature on a new message M' with "high" probability. $B^\pi(PK, M, \sigma)$ first runs $\mathsf{Vrfy}(PK, M, \sigma)$, answering the queries of Vrfy by forwarding them to π. Let $\vec{x}$ be the t-prefixes of the queries made by Vrfy, and $\vec{y}$ be the t-prefixes of the corresponding answers. Define $X = PK\|\vec{y}\|M\|\sigma$ and $Y = PK\|\vec{y}\|\vec{x}\|1$; note that $Y = F(X)$. Finally, algorithm B computes $PK'\|\vec{y}\,'\|M'\|\sigma' = A(Y)$ and outputs (M', σ').

The distribution on X (over random choice of $\pi \in \Pi_{t,n}$) is exactly given by $\mathcal{D}$. Furthermore, B outputs a successful forgery if both $F(PK'\|\vec{y}\,'\|M'\|\sigma') = Y$ and $M' \neq M$ hold. To see this, note that the first condition implies that $\vec{y} = \vec{y}\,'$, and hence we have $\mathsf{Vrfy}^{\vec{y}}(PK, M', \sigma') = 1$ and furthermore Vrfy makes queries with t-prefixes $\vec{x}$. But this then means that $\mathsf{Vrfy}^\pi(PK, M', \sigma') = 1$. If furthermore $M' \neq M$, this implies that (M', σ') is a successful forgery. Finally, $|B| \leq S_\Sigma$.

Let Eq be the event that $M' = M$. Then

$$\begin{aligned}
\mathsf{Succ}^*_B &\geq \Pr_{X \leftarrow \mathcal{D}}[Y = F(X); X' = A(Y) : X' \in F^{-1}(Y) \wedge \overline{\mathsf{Eq}}] \\
&= \Pr_{X \leftarrow \mathcal{D}}[Y = F(X); X' = A(Y) : X' \in F^{-1}(Y)] \\
&\quad - \Pr_{X \leftarrow \mathcal{D}}[Y = F(X); X' = A(Y) : X' \in F^{-1}(Y) \wedge \mathsf{Eq}] \\
&> \varepsilon + 2^{-S_p} + 1/2 \\
&\quad - \Pr[X \leftarrow \mathcal{D};\ PK\|\vec{y}\|\vec{x}\|1 = F(X); \\
&\qquad\qquad PK'\|\vec{y}\,'\|M'\|\sigma' = A(PK\|\vec{y}\|\vec{x}\|1) : M' = M] \\
&= \varepsilon + 2^{-S_p} + 1/2 \\
&\quad - \sum_{\vec{z}} \Pr[X \leftarrow \mathcal{D};\ PK\|\vec{y}\|\vec{x}\|1 = F(X); \\
&\qquad\qquad PK'\|\vec{y}\,'\|M'\|\sigma' = A(PK\|\vec{y}\|\vec{x}\|1) : \\
&\qquad\qquad M' = M \wedge \vec{x} = \vec{z}],
\end{aligned}$$

where the sum is over $\vec{z}$ consisting of q distinct t-bit strings. Continuing, we have

$$\begin{aligned}
\mathsf{Succ}^*_B &\geq \varepsilon + 2^{-S_p} + 1/2 \\
&\quad - \sum_{\vec{z}} \Pr[X \leftarrow \mathcal{D};\ PK\|\vec{y}\|\vec{x}\|1 = F(X); \\
&\qquad\qquad PK'\|\vec{y}\,'\|M'\|\sigma' = A(PK\|\vec{y}\|\vec{z}\|1) : M' = M] \\
&= \varepsilon + 2^{-S_p} + 1/2 - \sum_{\vec{z}} 2^{-m}, \qquad (9)
\end{aligned}$$

[7] Simulating a random $\pi \in \Pi_{t,n}$ is done in the natural way: query $x\|b$ is answered with $y\|b$ for random (unused) y, unless a query of the form $x\|b'$ (possibly made by a different algorithm) was previously answered with $y'\|b'$, in which case $y'\|b$ is returned (here, $|x| = |x'| = |y| = |y'| = t$).

where the final equality holds since $M \in \{0,1\}^m$ is chosen uniformly at random independent of PK and $\vec{y}$ (and hence A has no information about M). Noting that there are $2^{qt} \leq 2^{m-1}$ terms in the sum of Equation (9), we derive the contradiction $\mathsf{Succ}^*_B > \varepsilon + 2^{-S_p}$. ■

Upper bounds on the efficiency of signature schemes. As mentioned in the Introduction, our lower bounds focus on the efficiency of signature verification. We briefly observe some upper bounds on the efficiency of verification for one-time signatures on m-bit messages. The Lamport scheme [34] requires m invocations of a one-way permutation to verify a signature. Instead of signing bit-by-bit, the Lamport scheme can be modified to sign block-by-block. When basing the construction on a one-way permutation that is S-hard, it is possible to obtain provable security using blocks of size $\mathcal{O}(\log(S/m))$; this gives a signature scheme requiring only $O(m/\log(S/m))$ invocations for verification. If S is an arbitrary polynomial, this is essentially optimal as far as verification is concerned; yet, the key-generation time and public-key size are prohibitive. (In fact, the resulting scheme does not run in polynomial time unless S is polynomial.) An alternate approach is to include a universal one-way hash function h_s as part of the public key, and to use the (basic) Lamport scheme to sign $h_s(M)$. Verification now requires evaluation of h_s followed by a verification in the underlying Lamport scheme. Since h_s can be used to compress an arbitrary-length message to an n-bit string (when using an S-hard permutation on n bits) [36], we obtain a verification complexity of $\mathcal{O}(n + (m-n)/\log S)$ for $m \geq n$.

References

[1] B. Barak. How to Go Beyond the Black-Box Simulation Barrier. *42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 106–115, 2001.

[2] B. Barak. Constant-Round Coin-Tossing with a Man in the Middle, or Realizing the Shared Random String Model. *43rd IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 345–355, 2002.

[3] D. Beaver. Correlated Pseudorandomness and the Complexity of Private Computations. *28th ACM Symposium on Theory of Computing (STOC)*, ACM, pp. 479–488, 1996.

[4] M. Bellare and O. Goldreich. On Defining Proofs of Knowledge. *Adv. in Cryptology — Crypto 1992*, LNCS vol. 740, Springer-Verlag, pp. 390–420, 1993.

[5] M. Bellare and S. Goldwasser. New Paradigms for Digital Signatures and Message Authentication Based on Non-Interactive Zero-Knowledge Proofs. *Adv. in Cryptology — Crypto 1989*, LNCS vol. 435, Springer-Verlag, pp. 194–211, 1990.

[6] M. Bellare, S. Halevi, A. Sahai, and S. Vadhan. Many-to-one Trapdoor Functions and their Relation to Public-Key Cryptosystems. *Adv. in Cryptology — Crypto 1998*, LNCS vol. 1462, Springer-Verlag, pp. 283–298, 1998.

[7] M. Blum and S. Goldwasser. An Efficient Probabilistic Encryption Scheme Which Hides All Partial Information. *Adv. in Cryptology — Crypto '84*, LNCS vol. 263, Springer-Verlag, pp. 289–302, 1985.

[8] M. Blum and S. Micali. How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits. *SIAM J. Comp.* 13(4): 850–864, 1984.

[9] D. Catalano, R. Gennaro, and N. Howgrave-Graham. Paillier's Trapdoor Function Hides up to $O(n)$ Bits. *J. Crypto.* 15(4): 251–269, 2002.

[10] Y. Desmedt and Y. Frankel. Threshold Cryptosystems. *Adv. in Cryptology — Crypto 1989*, LNCS vol. 435, Springer-Verlag, pp. 307–315, 1990.

[11] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *SIAM J. Computing* 30(2): 391–437, 2000.

[12] U. Feige, A. Fiat, and A. Shamir. Zero-Knowledge Proofs of Identity. *J. Crypto.* 1(2): 77–94, 1988.

[13] M. Fischlin. On the Impossibility of Constructing Non-Interactive Statistically-Secret Protocols From any Trapdoor One-Way Function. *Cryptographers' Track — RSA 2002*, LNCS vol. 2271, pp. 79–95, 2002.

[14] R. Gennaro, Y. Gertner, and J. Katz. Lower Bounds on the Efficiency of Encryption and Digital Signature Schemes. *35th ACM Symposium on Theory of Computing (STOC)*, ACM, pp. 417–425, 2003.

[15] R. Gennaro and L. Trevisan. Lower bounds on the Efficiency of Generic Cryptographic Constructions. *41st IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 305–313, 2000.

[16] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The Relationship between Public-Key Encryption and Oblivious Transfer. *41st IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 325–335, 2000.

[17] Y. Gertner, T. Malkin, and O. Reingold. On the Impossibility of Basing Trapdoor Functions on Trapdoor Predicates. *42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 126–135, 2001.

[18] O. Goldreich. *Foundations of Cryptography, vol. 1: Basic Tools.* Cambridge University Press, Cambridge, UK, 2001.

[19] O. Goldreich. Draft of a Chapter on Cryptographic Protocols. July 2003. Available at http://www.wisdom.weizmann.ac.il/~oded/foc-vol2.html.

[20] O. Goldreich, S. Goldwasser, and S. Micali. On the Cryptographic Applications of Random Functions. *Adv. in Cryptology — Crypto '84*, LNCS vol. 263, Springer-Verlag, pp. 276–288, 1985.

[21] O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions. *J. ACM* 33(4): 792–807, 1986.

[22] O. Goldreich and L. Levin. Hard-Core Predicates for any One-Way Function. *21st ACM Symposium on Theory of Computing (STOC)*, ACM, pp. 25–32, 1989.

[23] O. Goldreich, S. Micali, and A. Wigderson. Proofs that Yield Nothing but Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *J. ACM* 38(3): 691–729, 1991.

[24] O. Goldreich, S. Micali, and A. Wigderson. How to Play Any Mental Game or A Completeness Theorem for Protocols with Honest Majority. *19th ACM Symposium on Theory of Computing (STOC)*, ACM, pp. 218–229, 1987.

[25] S. Goldwasser and S. Micali. Probabilistic Encryption. *J. Computer and System Sciences*, 28(2): 270–299, 1984.

[26] S. Goldwasser, S. Micali, and R. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen Message Attacks. *SIAM J. Computing*, 17(2): 281–308, 1988.

[27] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A Pseudorandom Generator from any One-Way Function. *SIAM J. Computing* 28(4): 1364–1396, 1999.

[28] J. Håstad, A. Schrift, and A. Shamir. The Discrete Logarithm Modulo a Composite Hides $O(n)$ Bits. *J. Computer and System Sciences*, 47(3): 376–404, 1993.

[29] R. Impagliazzo. Very Strong One-Way Functions and Pseudo-random Generators Exist Relative to a Random Oracle. Manuscript, 1996.

[30] R. Impagliazzo and M. Luby. One-Way Functions are Essential for Complexity-Based Cryptography. *30th IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 230–235, 1989.

[31] R. Impagliazzo and S. Rudich. Limits on the Provable Consequences of One-Way Permutations. *21st ACM Symposium on Theory of Computing (STOC)*, ACM, pp. 44–61, 1989.

[32] J. Kahn, M.E. Saks, and C.D. Smyth. A Dual Version of Reimer's Inequality and a Proof of Rudich's Conjecture. *15th IEEE Conference on Computational Complexity*, IEEE, pp. 98–103, 2000.

[33] J.H. Kim, D.R. Simon, and P. Tetali. Limits on the Efficiency of One-Way Permutation-Based Hash Functions. *40th IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 535–542, 1999.

[34] L. Lamport. Constructing Digital Signatures From a One-Way Function. Technical Report CSL-98, SRI International, 1979.

[35] M. Naor. Bit Commitment Using Pseudorandom Generators. *J. Crypto.* 4(2): 151–158, 1991.

[36] M. Naor and M. Yung. Universal One-Way Hash Functions and Their Cryptographic Applications. *21st ACM Symposium on Theory of Computing (STOC)*, ACM, pp. 33–43, 1989.

[37] O. Reingold, L. Trevisan, and S. Vadhan. Notions of Reducibility Between Cryptographic Primitives. *1st Theory of Cryptography Conference*, LNCS vol. 2951, Springer-Verlag, pp. 1–20, 2004.

[38] J. Rompel. One-Way Functions are Necessary and Sufficient for Secure Signatures. *22nd ACM Symposium on Theory of Computing (STOC)*, ACM, pp. 387–394, 1990.

[39] S. Rudich. *Limits on the Provable Consequences of One-Way Functions.* PhD thesis, University of California at Berkeley, 1988.

[40] S. Rudich. The Use of Interaction in Public Cryptosystems. *Adv. in Cryptology — Crypto 1991*, LNCS vol. 576, Springer-Verlag, pp. 242–251, 1992.

[41] D.R. Simon. Finding Collisions on a One-Way Street: Can Secure Hash Functions be Based on General Assumptions? *Adv. in Cryptology — Eurocrypt 1998*, LNCS vol. 1403, Springer-Verlag, pp. 334–345, 1998.

[42] A. C.-C. Yao. Theory and Application of Trapdoor Functions. *23rd IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 80–91, 1982.

[43] A. C.-C. Yao. How to Generate and Exchange Secrets. *27th IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE, pp. 162–167, 1986.