Lower Bounds on the Efficiency of Encryption and Digital Signature Schemes

Rosario Gennaro
IBM T.J. Watson Research Center
Yorktown Heights, NY
rosario@watson.ibm.com

Yael Gertner
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA
ygertner@saul.cis.upenn.edu

Jonathan Katz*
Dept. of Computer Science
University of Maryland
College Park, MD
jkatz@cs.umd.edu

ABSTRACT

A central focus of modern cryptography is to investigate the weakest possible assumptions under which various cryptographic algorithms exist. Typically, a proof that a "weak" primitive (e.g., a one-way function) implies the existence of a "strong" algorithm (e.g., a private-key encryption scheme) proceeds by giving an explicit construction of the latter from the former. In addition to showing the *existence* of such a construction, an equally important research direction is to explore the *efficiency* of such constructions.

Among the most fundamental cryptographic algorithms are digital signature schemes and schemes for public- or private-key encryption. Here, we show the first lower bounds on the efficiency of any encryption or signature construction based on black-box access to one-way or trapdoor one-way permutations. If S is the assumed security of the permutation π (i.e., no adversary of size S can invert π on a fraction larger than $1/S$ of its inputs), our results show that:

- Any public-key encryption scheme for m-bit messages must query π at least $\Omega(m/\log S)$ times.
- Any private-key encryption scheme for m-bit messages (with k-bit keys) must query π at least $\Omega(\frac{m-k}{\log S})$ times.
- Any signature verification algorithm for m-bit messages must query π at least $\Omega(m/\log S)$ times.

Our bounds match known upper bounds for the case of encryption.

We prove our results in an extension of the Impagliazzo-Rudich model. That is, we show that any black-box construction beating our lower bounds would imply the unconditional existence of a one-way function.

*A portion of this work was done while the author was visiting DIMACS.

STOC'03, June 9–11, 2003, San Diego, California, USA.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public-Key Cryptosystems; F.2 [**Analysis of Algorithms and Problem Complexity**]: Miscellaneous

General Terms

Security, Theory

Keywords

Lower Bounds, Digital Signatures, Encryption, Black-Box

1. INTRODUCTION

The aim of modern cryptography is to construct protocols that are both secure and efficient. Generally, work proceeds along two lines: (1) constructing protocols based on the weakest possible assumptions; and (2) improving the efficiency of known constructions, based on (perhaps) stronger assumptions. The first approach seeks to understand the assumptions necessary for the existence of a given cryptographic primitive, while the focus of the second is to construct efficient protocols based on any (reasonable) assumption; many times these goals are considered independently. However, it is also of interest to *combine* these approaches in order to determine the best achievable efficiency (for a given task) based on a particular assumption. Indeed, this approach is even more important now that minimal assumptions are known for most of the basic tools of cryptography.

Typically, improved efficiency is achievable *only* by making a stronger assumption; in many cases, a substantial gap exists between the efficiency of schemes based on a weak assumption and those based on stronger assumptions. As an example, it is known that public-key encryption may be based on any trapdoor permutation; the best known constructions, however, require $\Theta(m/\log k)$ invocations of the trapdoor permutation to encrypt m bits with security parameter k [3, 2, 11, 25]. On the other hand, the El Gamal public-key encryption scheme [7] requires only two function invocations to encrypt $\Theta(k)$ bits with security parameter k; security of this encryption scheme, however, is based on the decisional Diffie-Hellman assumption [4].

Much effort has been expended to bridge this gap, and a long line of research has focused on improving the efficiency of cryptographic protocols based on general assumptions. Recently, researchers have begun to consider the possibility that the efficiency of known constructions based on general assumptions might be the best achievable. Kim, Simon, and Tetali [17] initiated work in this direction by focusing on the construction of universal one-way hash functions (UOWHFs) from one-way permutations, and showing (non-tight) lower bounds on the efficiency of constructions of the former from the latter. Extending these results, Gennaro and Trevisan [8] showed that the known constructions of UOWHFs based on one-way permutations are in fact *optimal*. They also showed a similar result for the case of pseudorandom generators (PRGs). A consequence is that the efficiency of these primitives can be improved only by relying on stronger assumptions or non-black-box techniques.

Previous work in this area (focusing on UOWHFs and PRGs) has been limited to 1-party, or "stand-alone", cryptographic primitives for which there is no inherent notion of "interaction". We may also notice that UOWHFs and PRGs are typically used for "private-key" (or "shared-key") applications. The case of 2-party protocols, especially those executed in a "public-key" setting (such as public-key encryption and digital signatures), has been more difficult to analyze. In particular, Gennaro and Trevisan [8] state as an open question whether their techniques can be extended to show efficiency bounds for public-key encryption schemes based on trapdoor one-way permutations.

Note that this open question can be answered immediately for *specific constructions* of encryption schemes: namely, encryption schemes following the *hard-core bit paradigm* (as in [3, 2, 11, 25]). A corollary of [8] is that a generic one-way permutation with security S cannot have more than $\Theta(\log S)$ hard-core bits. This immediately implies a lower bound of $\Omega(m/\log S)$ invocations of the permutation for any encryption scheme following the hard-core bit paradigm. Yet, this idea does not seem to extend to give a lower bound that holds for *any* encryption scheme which might be constructed in an arbitrary way.

In this paper, we prove lower bounds on the efficiency of 2-party protocols in both the public- and private-key settings. We give results (summarized in Section 1.1) for both public- and private-key encryption as well as for digital signature schemes; our bounds match the known upper bounds for the case of encryption. The key new ideas are highlighted in Section 1.2.

1.1 Our Results

Informally, we say that a one-way (trapdoor) permutation $\pi : \{0,1\}^n \rightarrow \{0,1\}^n$ has security S if any circuit A of size smaller than 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 stated as follows.

Encryption. A public key encryption (PKE) scheme for m-bit messages is semantically-secure [12] 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. More formal definitions appear in Sections 2.3 and 2.4.

We show that *any* black-box construction of a public-key encryption scheme based on trapdoor permutations *must* query the oracle for the permutation $\Omega(m/\log S)$ times. Our lower bound applies to the number of invocations made by the encryption algorithm (i.e., *in addition to* any invocations made during the key generation phase) and is therefore tight. Our proof proceeds by translating the public-key encryption scheme to the *private-key* setting, and showing that any public-key scheme beating the above bound can be used to construct a private-key scheme in which the message is longer than the shared key (and π is not used at all). A result of Impagliazzo and Luby [14] then shows that this implies the existence of a one-way function (we give a simpler proof of this result in Section 3.1).

Using related techniques, we also show that any construction of a private-key encryption scheme based on one-way permutations which securely encrypts messages of length m using a shared key of length k (with $k < m$) must query the oracle for the permutation $\Omega(\frac{m-k}{\log S})$ times. Any construction using fewer than this many invocations of the permutation can be used to construct a one-way function (and hence an unconditionally-secure private-key encryption scheme in which the message is longer than the key). Again, this bound is tight as it matches known constructions.

Signature schemes. A *one-time signature scheme* is said to be secure if no efficient algorithm can forge a valid signature on a new message after seeing a signature on a single message of its choice. In this work, we prove lower bounds on the efficiency of the verification algorithm for one-time signature schemes (where, furthermore, the observed signature is on a randomly-chosen message); our results thus immediately extend to schemes satisfying stronger notions of security (e.g., [13]). We prove that, for any construction of a one-time signature scheme for messages of length m, the verification algorithm must query the permutation oracle $\Omega(m/\log S)$ times. Here, we directly show that any construction beating this bound can be used to give an explicit construction of a one-way function.

We observe the following upper bounds for the complexity of verification in one-time signature schemes: The basic Lamport scheme [18] requires m invocations of the permutation. An alternate approach is to use the scheme suggested in [1]; instantiating their method using a UOWHF with keys of length $O(n \log m)$ [23] and the Lamport scheme yields verification complexity $O(n \log m + m/\log S)$. Finally, we mention an approach which (to the best of our knowledge) has not appeared previously: let the public key be (PK_1, PK_2, H_1, H_2) where PK_1, PK_2 are two independent public keys for the Lamport scheme (with associated secret keys SK_1, SK_2) and H_1, H_2 are two UOWHFs mapping m bits to $O(n)$ bits. To sign message M, choose random $r \in \{0,1\}^m$ and sign $H_1(r)$ using SK_1 and $H_2(r \oplus M)$ using SK_2; the resulting signature includes both partial signatures. This achieves verification complexity $O(n + m/\log S)$.

The best constructions available for schemes secure in the stronger sense of [13] are less efficient still.

1.2 Our Techniques

To show their lower bounds, Gennaro and Trevisan use two main observations: (1) a *random* permutation is (with very high probability) hard to invert; and (2) a bounded amount of oracle queries to a random permutation can be replaced by generating random answers, as needed. In the case of, e.g., a PRG constructed from a one-way permutation, these random bits specifying the value of the permutation can be included with the seed; this modified construction remains a PRG as long as the output remains longer than the seed. If the number of queries is "small", the seed-length does not grow "too much" and hence remains shorter than the output-length. The proof of their lower bound concludes by noticing that this modified PRG does not use the oracle at all, and is thus unconditionally secure. To summarize, PRGs making "few" black-box oracle queries exist only if PRGs exist unconditionally (i.e., without making any oracle calls) something we do not currently know how to prove (it would imply in particular that $P \neq NP$).

Unfortunately, in a 2-party "public-key" setting we cannot use this same elegant argument in a straightforward manner. Specifically, in the case of 1-party primitives like PRGs and UOWHFs the randomness needed to simulate black-box access to the permutation can be given as input to (or can be generated by) the algorithm itself. In the 2-party setting, however, we must be careful to ensure consistency between the various parties. Although in the private-key case this is — in principle — possible (by sharing this randomness in advance as part of the secret key), it becomes much harder to ensure consistency in the "public-key" setting where no pre-shared information exists. Note that placing the necessary randomness in the public key (so that all parties know it) may destroy the security of the protocol altogether!

The proofs of our lower bounds follow a 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 shared as part of the secret key without compromising the functionality or the security of the scheme. Note, however, that our final result is somewhat weaker than the analogous result of [8]: namely, we show that a public-key encryption scheme making "few" black-box oracle queries exists only if a secure *private-key* encryption scheme exists unconditionally; although this is weaker than showing the unconditional existence of a secure public-key encryption scheme, it suffices to imply $P \neq NP$.

1.3 Oracle Separations

Our results hold in the oracle separation model of Impagliazzo and Rudich [15] (see [8, 10] for interesting discussions about this model) which has been used to separate primitives under black-box reductions. These are an important subclass of reductions, since many reductions in cryptography are black-box.[1]

A black-box reduction of B to A is a construction of B out of A that ignores the internal structure of the circuit implementing A. Furthermore, the security of the resulting implementation of B is based solely on the security of A. Black-box reductions work relative to any oracle; therefore, one way to prove that a black-box reduction of B to A is impossible is to prove the existence of an oracle relative to which A exists whereas B does not. Since the initial work of Impagliazzo and Rudich [15], many separation results [22, 21, 24, 9, 16, 10, 6] in this model have appeared. Note that these works do not address the *efficiency* of constructions under a given assumption; rather, they address the *existence* of constructions under a given assumption.

Here we use a related technique used previously in [17, 8]. To show a lower bound on the efficiency of constructions of B from A, we show an oracle for A relative to which a construction of B beating the bound would yield the unconditional existence of something beyond current techniques; e.g., a one-way function.

1.4 Extensions and Open Problems

Our lower bound for public-key encryption matches the known upper bound for schemes semantically secure against chosen-plaintext attacks. For "long" messages M, our bound asymptotically matches the efficiency of the following scheme achieving non-malleability under chosen-plaintext attacks (NM-CPA): encrypt a "short", random key K using the (black-box) NM-CPA scheme of [5], and then use an NM-CPA *private-key* scheme (and key K) to encrypt M. Can one show a tight lower bound for NM-CPA public-key encryption of "short" messages (say, for a single bit)? For schemes secure against chosen-ciphertext attacks, the situation is even more complex since no black-box constructions are known. A black-box construction of IND-CCA public-key encryption — or a proof that such a construction is impossible — would be an interesting result. Finally, can one show any lower bounds on *non*-black-box constructions?

For signature schemes, our bound on the efficiency of verification is not tight even for one-time signature schemes; the gap widens for schemes secure when multiple messages are signed. Finally, it would be interesting to bound the efficiency of the key-generation and signing algorithms. (The Lamport scheme [18] requires no oracle calls when signing but $O(m)$ oracle calls for key generation; thus, it seems there is a trade-off between the complexity of the two.)

2. PRELIMINARIES

2.1 One-Way Permutations

Following prior work, we model black-box access to a one-way permutation by giving algorithms access to an oracle π for a *random* permutation, which is one-way with high probability. Formally, a function $f : \{0,1\}^n \to \{0,1\}^n$ is (S, ε)-one way if for every circuit A of size $\leq S$ we have

$$\mathbf{Pr}_x[A^f(f(x)) \in f^{-1}(f(x))] \leq \varepsilon.$$

(To reduce the number of parameters, we will also say that a function is *S-hard* if it is $(S, 1/S)$-one way.) Denote by Π_t the set of all permutations over $\{0,1\}^t$. The following result then holds:

THEOREM 1 ([8]). *For sufficiently large t, a random $\pi \in \Pi_t$ is $2^{t/5}$-hard with probability at least $1 - 2^{-2^{t/2}}$.*

For $t < n$, denote by $\Pi_{t,n}$ the subset of Π_n such that $\pi \in \Pi_{t,n}$ iff $\pi(a, b) = (\hat{\pi}(a), b)$ for some $\hat{\pi} \in \Pi_t$. A corollary to Theorem 1 is that if $t = 5 \log S_p$, then for $n > t$ a randomly chosen $\pi \in \Pi_{t,n}$ is S_p-hard with high probability.

COROLLARY 1. *For sufficiently large t and $n > t$, a random $\pi \in \Pi_{t,n}$ is $2^{t/5}$-hard with probability at least $1 - 2^{-2^{t/2}}$.*

[1] We stress, however, that a number of non-black-box cryptographic constructions based on zero-knowledge techniques are known (e.g., all known constructions of IND-CCA public-key encryption from trapdoor permutations).

2.2 Trapdoor Permutations

Our model for *trapdoor* permutations is somewhat more involved and has not previously appeared. We represent a family of trapdoor permutations as a tri-partite oracle $\tau = (G, F, F^{-1})$. Informally, G will be the *key generation* oracle which when queried on a string tk (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 $G(tk) = k$ and $F(k, x) = y$, then $F^{-1}(tk, y) = x$.

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

- $G \in \Pi_n$ is an oracle for a random permutation over $\{0,1\}^n$.
- $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is an oracle such that, for each $k \in \{0,1\}^n$, $F(k, \cdot)$ is an independent random permutation on $\{0,1\}^n$.
- $F^{-1} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is an oracle defined as follows: $F^{-1}(tk, y)$ returns the unique x such that $G(tk) = k$ and $F(k, x) = y$.

We say that trapdoor permutation family $\tau = (G, F, F^{-1})$ is (S, ε)-trapdoor one way if for every circuit A of size $\leq S$ we have

$$\Pr_{x,tk}[k := G(tk) \ : \ A^\tau(k, F(k, x)) = x] \leq \varepsilon.$$

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*.

We now show an analogue of Theorem 1 for trapdoor permutations. The basic idea is to notice that, for any k, $F(k, \cdot)$ is a random permutation so Theorem 1 applies except that we need to take into account the possibility that the adversary is able to invert G. But since G is also a random permutation, all we need to do is apply Theorem 1 twice.

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

PROOF. We actually show the stronger result that τ is $(2^{t/5}, 2^{1-t/5})$-trapdoor one-way. The simpler expression $2^{t/6}$ follows by trivial upper bounding.

Let A be a circuit of size $S = 2^{t/5}$. 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 $p = 1 - 2^{-2^{t/2}}$ the permutation $F(k, \cdot)$ is one-way with security $2^{t/5}$. This holds as long as A^τ never queries the oracle F^{-1} on tk such that $G(tk) = k$. 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 adversary A and permutation family τ, let Q_{tk} denote the event that A^τ queries $F^{-1}(tk, y)$ for some y. We have:

$$\Pr_{x,tk}[k := G(tk) \ : \ A^\tau(k, F(k, x)) = x] \leq$$
$$\Pr_{x,tk}[k := G(tk) \ : \ A^\tau(k, F(k, x)) = x \mid \overline{Q_{tk}}] \qquad (1)$$
$$+ \Pr_{x,tk}[k := G(tk) \ ; \ x \leftarrow A^\tau(k, F(k, x)) \ : \ Q_{tk}]. \qquad (2)$$

Now, (1) can be upper bounded by $2^{-t/5}$ since we are conditioning on the event that A^τ never queries F^{-1} on the correct tk. Similarly, (2) can be upper bounded by $2^{-t/5}$ since A^τ can be used as a circuit inverting G on input k. Summing gives the desired result. □

Consider an oracle $\hat{\tau} = (\hat{G}, \hat{F}, \hat{F}^{-1}) \in T_t$. From $\hat{\tau}$ we can construct an oracle $\tau = (G, F, F^{-1}) \in T_n$ as:[2]

- $G(tk) = G(tk_a, tk_b) = (\hat{G}(tk_a), tk_b)$. That is, $G \in \Pi_{t,n}$ is a permutation acting only on the first t bits of its input.
- $F(k, x) = F((k_a, k_b), (x_a, x_b)) = (\hat{F}(k_a, x_a), x_b)$. That is, $F(k, \cdot) \in \Pi_{t,n}$ is a permutation acting only on the first t bits of its input. Moreover, this permutation is uniquely determined by the first t bits of k.
- $F^{-1}(tk, y)$ returns the unique x such that $G(tk) = k$ and $F(k, x) = y$.

For $t < n$, we define $T_{t,n} \subset T_n$ as the family of oracles so constructed, for all $\hat{\tau} \in T_t$. It is not hard to see that "inverting" an oracle $\tau \in T_{t,n}$ requires "inverting" the underlying $\hat{\tau} \in T_t$. Thus, we obtain the following:

COROLLARY 2. *For sufficiently large t, for any $n > t$, if we pick at random an oracle $\tau \in T_{t,n}$, then with probability at least $1 - 2^{1-2^{t/2}}$ the oracle is $2^{t/6}$-trapdoor one-way.*

2.3 Public-Key Encryption

A construction of a public-key encryption scheme for m-bit messages is a procedure $\mathcal{PKE}^{(\cdot)}$ that is given access to an oracle $\tau \in T_n$ for a (family of) trapdoor permutations. $\mathcal{PKE}^{(\cdot)}$ consists of the following algorithms:

- *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.
- *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.
- *Decryption algorithm* $\mathsf{Dec}^{(\cdot)}$ is a deterministic algorithm which, on input the secret key sk and a ciphertext C, outputs either a message $M \in \{0,1\}^m$ or $\perp$.

We assume that $\mathcal{PKE}^{(\cdot)}$ is *correct*; that is, for all $\tau \in T_n$, for all (pk, sk) output by Gen^τ, for all $M \in \{0,1\}^m$, and for all C output by $\mathsf{Enc}^\tau(pk, M)$ we have $\mathsf{Dec}^\tau(sk, C) = M$. Our results can be modified appropriately if this is not the case.

We now define the security of a given construction $\mathcal{PKE}^{(\cdot)}$. We say random variables $\mathcal{X}^\tau, \mathcal{Y}^\tau$ which depend upon an oracle τ are (S, ε)-indistinguishable (we will write $\mathcal{X}^\tau \overset{(S,\varepsilon)}{\approx} \mathcal{Y}^\tau$), if for every circuit Dist of size $\leq S$ we have

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

For a given τ and any message $M \in \{0,1\}^m$ we denote by $\mathcal{PKE}^\tau(M)$ the probability distribution over the *entire*

[2]We use the following notation: if $|x| = n$, we write $x = (x_a, x_b)$ with $|x_a| = t$ and $|x_b| = n - t$.

view of an adversary eavesdropping on communication to the receiver; i.e.,

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

We say a public-key encryption scheme $\mathcal{PKE}^\tau$ is (S_e, ε)-secure if for all $M_0, M_1 \in \{0,1\}^m$ we have:

$$\mathcal{PKE}^\tau(M_0) \stackrel{(S_e,\varepsilon)}{\approx} \mathcal{PKE}^\tau(M_1).$$

Finally, we say that $\mathcal{PKE}^{(\cdot)}$ is an $(S_{tp}, S_e, \varepsilon)$-secure PKE construction if for every oracle $\tau \in T_n$ that is S_{tp}-trapdoor one way, $\mathcal{PKE}^\tau$ is (S_e, ε)-secure.

2.4 Private-Key Encryption

The case of private-key encryption is handled via appropriate modification of the above. A construction of a private-key encryption scheme for m-bit messages using k-bit keys is a procedure $\mathcal{SKE}^{(\cdot)}$ that is given access to an oracle $\pi \in \Pi_n$ for a one-way permutation. In contrast to the public-key setting, $\mathcal{SKE}^{(\cdot)}$ consists only of algorithms $\mathsf{Enc}^{(\cdot)}$ and $\mathsf{Dec}^{(\cdot)}$. The shared key sk is uniformly distributed in $\{0,1\}^k$. Algorithm $\mathsf{Enc}^{(\cdot)}$, which may be probabilistic, takes as input 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 sk and 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 $\pi \in \Pi_n$, for all $sk \in \{0,1\}^k$, for all $M \in \{0,1\}^m$, and for all C output by $\mathsf{Enc}^\pi(sk, M)$ we have $\mathsf{Dec}^\pi(sk, C) = M$.

For a given π and any message $M \in \{0,1\}^m$ we denote by $\mathcal{SKE}^\pi(M)$ the probability distribution over the view of an adversary eavesdropping on the communication; i.e.,

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

This definition can be extended for the case of an interactive encryption scheme in the obvious way (namely, the adversary is given the entire transcript of the communication).

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

$$\mathcal{SKE}^\tau(M_0) \stackrel{(S_e,\varepsilon)}{\approx} \mathcal{SKE}^\tau(M_1).$$

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

2.5 Signature Schemes

A construction of a digital signature scheme for m-bit messages is a procedure $\mathcal{SIG}^{(\cdot)}$ that is given access to an oracle $\pi \in \Pi_n$ for a one-way permutation. $\mathcal{SIG}^{(\cdot)}$ consists of:

- *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.
- *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.
- *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 $\pi \in \Pi_n$, for all (PK, SK) output by Gen^π, for all $M \in \{0,1\}^m$, and for 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 very weak: we only require the scheme to be secure against an adversary who sees 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. [13]) trivially achieves this "weak" level of security, our results immediately imply a bound for the more general case.

We say that $\mathcal{SIG}^{(\cdot)}$ is an $(S_p, S_\Sigma, \varepsilon)$-secure signature construction if for every $\pi \in \Pi_n$ that is S_p-hard and for all circuits A of size $\leq S_\Sigma$ we have

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

3. LOWER BOUNDS

3.1 Public-Key Encryption

Let $\mathcal{PKE}^{(\cdot)} = (\mathsf{Gen}^{(\cdot)}, \mathsf{Enc}^{(\cdot)}, \mathsf{Dec}^{(\cdot)})$ be a public-key encryption scheme for messages of length m that expects as an oracle a (family of) trapdoor permutations τ, and suppose $\mathcal{PKE}^\tau$ is secure whenever τ is S_{tp}-hard. We prove that unless Enc^τ queries τ at least $\Omega(m/\log S_{tp})$ times, an unconditional one-way function exists. In more detail, we first show 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. A previous result of Impagliazzo and Luby [14] (see also Theorem 3) shows that this implies the existence of a one-way function.

Our construction of a private-key encryption scheme proceeds by first noticing that τ is S_{tp}-hard even when $G : \{0,1\}^n \to \{0,1\}^n$ randomly permutes the first $t = 6 \log S_{tp}$ bits of its input and leaves the remaining bits fixed, and furthermore $F(k, \cdot) : \{0,1\}^n \to \{0,1\}^n$ (for each $k \in \{0,1\}^n$) randomly permutes the first t bits of its input and leaves the remaining bits fixed (i.e., $\tau \in T_{t,n}$; see Corollary 2). We then observe that each query to τ can be "simulated" (i.e., queries by $\mathsf{Gen}, \mathsf{Enc}$, and Dec to τ can be "answered") by generating t bits of randomness to prepare the answers as needed, as in [8]; a more detailed description of the simulation of τ is given in the proof of Theorem 4. This suggests a 2-round unconditional encryption scheme for m-bit messages in which the receiver first runs Gen with a simulation of τ as described to generate (pk, sk) and sends pk to the sender. The sender computes $C \leftarrow \mathsf{Enc}(pk, M)$ with a simulation of τ and sends C back to the receiver. The receiver computes M via $\mathsf{Dec}(sk, C)$.

While this encryption scheme is "secure", it may not be correct! Indeed, to ensure correct decryption the parties need to make sure that their simulated calls to τ are answered consistently. This can be achieved by having each party send, along with his message, a list of t-prefixes (i.e., the first t bits) for the queries and answers generated thus far. Now, however, security is no longer guaranteed. But this can be easily remedied in the private-key setting: the parties simply share a random 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 $\Theta(t \cdot (q_g + q_e))$ to encrypt an m-bit message. Recalling the result of Impagliazzo and Luby [14], this construction requires that $q_g + q_e = \Omega(m/t)$ since the opposite would imply the unconditional existence of a one-way function. This already gives a "weak" lower bound. To obtain the better lower bound $q_e = \Omega(m/t)$, additional work is needed; details are given in the proof of Theorem 4.

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 appears in [14], we give a much simpler proof here which may be of independent interest. Note that the theorem applies even to an interactive encryption protocol.

THEOREM 3 ([14]). *Let $(\mathsf{Enc}, \mathsf{Dec})$ be an (S, δ)-secure private-key encryption scheme in which the shared key is of length k and the message is of length $m > k$. Let S_e be the size of the circuit for Enc. Then for any $\ell \in \mathbb{N}$ there exists an $(S - 2\ell S_e, \ell\delta + 2^{-\ell(m-k)})$-one-way function.*

PROOF. We prove the theorem for non-interactive encryption schemes; extension to the case of interactive encryption is straightforward. First note that 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 $(\mathsf{Enc}_\ell, \mathsf{Dec}_\ell)$ 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 such that $\mathsf{Succ}_{B,f} \stackrel{\text{def}}{=} \Pr[sk \leftarrow \{0,1\}^{\ell k}; M \leftarrow \{0,1\}^{\ell m}; C \leftarrow \mathsf{Enc}_\ell(sk, M) : B(C\|M) \in f^{-1}(C\|M)] > \delta'$ and $|B| \leq S'$.

We use B to construct an algorithm A breaking the security of $(\mathsf{Enc}_\ell, \mathsf{Dec}_\ell)$. Let M_0, M_1 be two messages chosen randomly from $\{0,1\}^{\ell m}$. On input C (representing either an encryption of M_0 or M_1), algorithm A computes $sk'\|M'\|\omega' := B(C\|M_0)$ and checks whether $f(sk', M', \omega') \stackrel{?}{=} C\|M_0$. If not, then A outputs 0. If so, then A outputs 1. Note that $|A| = |B| + \ell S_e \leq S - \ell S_e$.

First note that $\Pr_{C \in \mathcal{SKE}(M_0)}[A(C) = 1] = \mathsf{Succ}_{B,f} > \delta'$. Secondly, we have

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

But then

$$
\left| \Pr_{C \in \mathcal{SKE}(M_0)}[A(C) = 1] - \Pr_{C \in \mathcal{SKE}(M_1)}[A(C) = 1] \right| > \delta' - 2^{\ell(k-m)} \geq \ell\delta,
$$

a contradiction. □

Our main result of this section is the following:

THEOREM 4. *Let $\mathcal{PKE}^{(\cdot)}$ be an $(S_{tp}, S_e, \varepsilon)$-secure PKE construction for messages of length m, and let $t = 6 \log S_{tp}$. If Gen makes q_g queries to its oracle τ, Enc makes $q_e < m/5t$ queries to its oracle τ, and*

$$
\varepsilon < (1/2 - 2^{1-S_{tp}}) \cdot (2 \cdot \lceil 5tq_g/(m - 5tq_e) \rceil)^{-1},
$$

then there exists a one-way function (without access to τ).

PROOF. Let $\mathcal{PKE}^{(\cdot)} = (\mathsf{Gen}^{(\cdot)}, \mathsf{Enc}^{(\cdot)}, \mathsf{Dec}^{(\cdot)})$. As in the previous proof, for any $\ell \in \mathbb{N}$ we may define public-key encryption scheme $\mathcal{PKE}_\ell^{(\cdot)} = (\mathsf{Gen}^{(\cdot)}, \mathsf{Enc}_\ell^{(\cdot)}, \mathsf{Dec}_\ell^{(\cdot)})$ for ℓm-bit messages in the natural way; note that the key generation algorithm only needs to be run once since we are now in the public-key setting. It is easy to see (via hybrid argument) that $\mathcal{PKE}_\ell^{(\cdot)}$ is an $(S_{tp}, S_e - \ell S_{\mathsf{Enc}}, \ell\varepsilon)$-secure PKE construction, where S_{Enc} is the size of the circuit for Enc. Also, the number of queries made by Enc_ℓ to its oracle is at most ℓq_e.

Set $\ell = 2 \cdot \lceil 5t\, q_g/(m - 5t\, q_e) \rceil$, and let $S' = S_e - \ell S_{\mathsf{Enc}}$ and $\delta = \ell\varepsilon + 2^{1-S_{tp}}$. 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. The value for ℓ chosen above ensures that $5t \cdot (q_g + \ell q_e) < \ell m$ and $\delta < 1/2$; thus, application of Theorem 3 (with $\ell = 1$ there) yields the statement of the present theorem.

Since $\mathcal{PKE}_\ell^{(\cdot)}$ is an $(S_{tp}, S', \ell\varepsilon)$-secure public-key encryption scheme, it means that if τ is S_{tp}-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_{tp}-hard except with probability at most $2^{1-S_{tp}}$. A straightforward averaging argument 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

$$
\begin{aligned}
&\left| \Pr_{\substack{\tau \in T_{t,n} \\ v \in \mathcal{PKE}_\ell^\tau(M_0)}}[B^\tau(v) = 1] - \Pr_{\substack{\tau \in T_{t,n} \\ v \in \mathcal{PKE}_\ell^\tau(M_1)}}[B^\tau(v) = 1] \right| \\
&\qquad \leq \ell\varepsilon + 2^{1-S_{tp}} \\
&\qquad = \delta
\end{aligned}
\tag{3}
$$

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. We first describe the details of this simulation. The simulation procedure $\mathcal{SIM}$ must 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 $tk\|b$ to G was answered by $k\|b$ (where $|tk| = |k| = t$), then subsequent query $tk'\|b'$ to G must be answered by $k'\|b'$ where $k' = k$ iff $tk' = tk$. A more involved procedure is needed to answer queries to F and F^{-1}.

PROCEDURE 1. $\mathcal{SIM}(L)$

- *On query $tk\|b$ to G (where $|tk| = t$): if $\exists k$ such that $(tk, k) \in L$, return $k\|b$. Otherwise pick random $k \in$*

$\{0,1\}^t$ such that $\forall tk' : (tk', k) \notin L$, return $k\|b$, and store (tk, k) in L.

- *On query $(k\|b, x\|b')$ to F (where $|k| = |x| = t$):*
  1. *if $\exists y$ s.t. $(k, x, y) \in L$, return $y\|b'$.*
  2. *Otherwise, if $\exists tk$ s.t. $(tk, k) \in L$, 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 $tk \in \{0,1\}^t$ such that $\forall k' : (tk, k') \notin L$, choose random $y \in \{0,1\}^t$, return $y\|b'$, and store (k, tk) and (k, x, y) in L.*
- *On query $(tk\|b, y\|b')$ to F^{-1} (where $|tk| = |y| = t$):*
  1. *if $\exists k, x$ s.t. $(tk, k), (k, x, y) \in L$, return $x\|b'$*
  2. *Otherwise, if $\exists k$ s.t. $(tk, k) \in L$, 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 $\forall tk' : (tk', k) \notin L$, choose random $x \in \{0,1\}^t$, return $x\|b'$, and store (tk, k) and (k, x, y) in L*

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

We 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 Enc'. 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 $M := \mathsf{Dec}_\ell^{\mathcal{SIM}(L)}(sk, C_i)$.

Because $\mathcal{SIM}$ perfectly simulates a random $\tau \in T_{t,n}$, 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 breaking the security of $\mathcal{PKE}_\ell$ 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 $A(pk, \hat{s}_1, C, \hat{s}_2)$. Note that B makes no use of the oracle τ. Because the keys s_1, s_2 of $\mathcal{SKE}$ are used as a one-time pad, it is easy to see that:

$$\Pr_{\substack{\tau \in T_{t,n} \\ (pk, C) \in \mathcal{PKE}_\ell^\tau(M_0)}}[B(pk, C) = 1] = \Pr_{v \in \mathcal{SKE}(M_0)}[A(v) = 1]$$

and, similarly,

$$\Pr_{\substack{\tau \in T_{t,n} \\ (pk, C) \in \mathcal{PKE}_\ell^\tau(M_1)}}[B(pk, C) = 1] = \Pr_{v \in \mathcal{SKE}(M_1)}[A(v) = 1].$$

Thus, the advantage of B is equal to the advantage of A (which is greater than δ), contradicting (3). $\square$

3.2 Private-Key Encryption

The techniques of the previous section can be adapted to show a similar lower bound for private-key encryption schemes. 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 4, 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 4 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, this requires that $k + 2tq_e \geq m$ since the opposite implies the unconditional existence of a one-way function (cf. Theorem 3).

Theorem 5. *Let $(\mathsf{Enc}^{(\cdot)}, \mathsf{Dec}^{(\cdot)})$ be an (S_p, S_e, ε)-secure SKE construction for messages of length m using a key of length k in which Enc makes $q_e < \frac{m-k}{10 \log S_p}$ queries to its oracle π, and $\varepsilon < 1/2 - 2^{-S_p}$. Then there exists a one-way function (without access to π).*

Note that the theorem implies the existence of a secure (non-trivial) private-key encryption scheme (i.e., without access to π), since a one-way function suffices for this purpose.

3.3 Signature Schemes

We now show a lower bound on the efficiency of black-box constructions for signature schemes based on one-way permutations (we no longer consider trapdoor one-way permutations since one-way permutations suffice to construct signature schemes [18, 19]; indeed, one-way functions are sufficient [20]). 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 π, 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 $\mathcal{SIG}$ an unconditional one-way function. Note that this one-way function could then be used to construct an unconditionally-secure signature scheme (i.e., without any access to π).

We start with an informal overview of our proof technique (formalized in the proof of Theorem 6). 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 amounts to signature forgery. As presently defined, however, evaluating F_1 requires calls to π and is therefore not unconditionally one-way (recall, our goal is to construct a function which does not use π). As in Section 2.1, however, one may observe that π is S_p-hard (and thus $\mathcal{SIG}$ is secure) when π randomly permutes the first $t = 5 \log S_p$ bits of its input while leaving the remaining bits fixed (cf. Corollary 1). If Vrfy makes q queries to its oracle for π, then specifying

qt bits as the answers to these queries removes any need to query the oracle. Based on this approach, one might consider the function

$$F_2(PK, y_1, \ldots, y_q, M, \sigma) = PK\|y_1\| \cdots \|y_q\|x_1\| \cdots \|x_q\|\mathsf{Vrfy}^{y_1,\ldots,y_q}(PK, M, \sigma), \quad (4)$$

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). 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 1). 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 forged signature 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 in this case, 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 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 $|\vec{x}| = |x_1| + \cdots + |x_q| < |M|$, 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.

THEOREM 6. *Let $\mathcal{SIG}^{(\cdot)} = (\mathsf{Gen}^{(\cdot)}, \mathsf{Sign}^{(\cdot)}, \mathsf{Vrfy}^{(\cdot)})$ be an $(S_p, S_\Sigma, \varepsilon)$-secure signature construction for m-bit messages in which $\varepsilon < 1/2 - S_p^{-1}$ and Vrfy makes $q < m/5 \log S_p$ queries to its oracle π. Then there exists a one-way function (without access to π).*

PROOF. Before giving the details of the proof, we 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 standard definition.

LEMMA 1. *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 f' that is $(S - S_\mathcal{D}, \delta)$-one way.

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

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

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

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

a contradiction. □

Returning to the proof of the theorem, let S_{Vrfy} be the size of the circuit for Vrfy. Our proof 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:

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

Lemma 1 then yields the desired result.

If $\mathcal{SIG}^{(\cdot)}$ is an $(S_p, S_\Sigma, \varepsilon)$-secure signature construction, it means that if $\pi : \{0,1\}^n \to \{0,1\}^n$ is S_p-hard then, for any circuit B of size $\leq S_\Sigma$ we have $\Pr[\mathsf{Forge}_{\pi,B}] \leq \varepsilon$ where

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

Let $t = 5 \log S_p$. Corollary 1 indicates that a random permutation $\pi \in \Pi_{t,n}$ is S_p-hard except with probability at most $2^{-2^{t/2}} < 2^{-S_p}$. An averaging argument then implies that for any circuit B of size $\leq S_\Sigma$ we have

$$\Pr[\pi \leftarrow \Pi_{t,n} : \mathsf{Forge}_{\pi,B}] < \varepsilon + 2^{-S_p}. \quad (5)$$

Our one-way function F is defined as in (4) which we repeat 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 $y_i\|b_i$. Define distribution $\mathcal{D}$ by the output of the following efficient procedure which takes uniformly distributed coins $r_g, r_s, r_y \in \{0,1\}^*$ and $r_M \in \{0,1\}^m$:

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

where the random coins r_y are used to simulate[3] a random $\pi \in \Pi_{t,n}$, and $\vec{y}$ represents the (ordered) t-prefixes of the answers given in response to the queries of Vrfy.

Assume toward a contradiction that there exists a circuit A of size $\leq S_\Sigma - S_{\mathsf{Vrfy}}$ for which $\Pr[X \leftarrow \mathcal{D} : A(F(X)) \in F^{-1}(F(X))] > \varepsilon + 2^{-S_p} + 1/2$. We use A to construct an algorithm B that, given PK and a signature σ of a random message M, forges a signature on a new message M'. $B^\pi(PK, M, \sigma)$ first runs $\mathsf{Vrfy}(PK, M, \sigma)$, answering the queries of Vrfy by forwarding them to π. Let $\vec{x}$ be

[3]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, $|x| = |x'| = |y| = |y'| = t$).

the t-prefixes of the queries made by Vrfy and let $\vec{y}$ be the t-prefixes of the corresponding answers. Define $X = PK\|\vec{y}\|M\|\sigma$ and $Y = PK\|\vec{y}\|\vec{x}\|1 = F(X)$. 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 that specified by $\mathcal{D}$. Furthermore, B outputs a successful forgery whenever both $F(PK'\|\vec{y}\,'\|M'\|\sigma') = Y$ and $M' \neq M$ hold. To see this, note that the first condition implies that $\mathsf{Vrfy}^{\vec{y}}(PK, M', \sigma') = 1$ and furthermore Vrfy makes queries with t-prefixes $\vec{x}$. But this then means that $\mathsf{Vrfy}^{\pi}(PK, M', \sigma') = 1$, since we have already determined that $\pi(x_1\|b) = y_1\|b, \ldots, \pi(x_q\|b) = y_q\|b$ (for arbitrary b). If furthermore $M' \neq M$, this implies that (M', σ') is a successful forgery. Finally, note that $|B| \leq S_\Sigma$.

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

$$
\begin{aligned}
&\Pr[\pi \leftarrow \Pi_{t,n} : \mathsf{Forge}_{\pi,B}] \\
&\quad\geq \Pr_{X \leftarrow \mathcal{D}}[Y = F(X); X' = A(Y) : X' \in F^{-1}(Y) \wedge \overline{\mathsf{Eq}}] \\
&\quad= \Pr_{X \leftarrow \mathcal{D}}[Y = F(X); X' = A(Y) : X' \in F^{-1}(Y)] \\
&\qquad - \Pr_{X \leftarrow \mathcal{D}}[Y = F(X); X' = A(Y) : X' \in F^{-1}(Y) \wedge \mathsf{Eq}] \\
&\quad> \varepsilon + 2^{-S_p} + 1/2 \\
&\qquad - \Pr[X \leftarrow \mathcal{D};\ PK\|\vec{y}\|\vec{x}\|1 = F(X); \\
&\qquad\qquad PK'\|\vec{y}\,'\|M'\|\sigma' = A(PK\|\vec{y}\|\vec{x}\|1) : M' = M] \\
&\quad= \varepsilon + 2^{-S_p} + 1/2 \\
&\qquad - \textstyle\sum_{\vec{z}} \Pr[X \leftarrow \mathcal{D};\ PK\|\vec{y}\|\vec{x}\|1 = F(X); \\
&\qquad\qquad PK'\|\vec{y}\,'\|M'\|\sigma' = A(PK\|\vec{y}\|\vec{x}\|1) : \\
&\qquad\qquad M' = M \wedge \vec{x} = \vec{z}] \\
&\quad\geq \varepsilon + 2^{-S_p} + 1/2 \\
&\qquad - \sum_{\vec{z}} \Pr[X \leftarrow \mathcal{D}; PK\|\vec{y}\|\vec{x}\|1 = F(X); \\
&\qquad\qquad PK'\|\vec{y}'\|M'\|\sigma' = A(PK\|\vec{y}\|\vec{z}\|1) : M' = M] \\
&\quad= \varepsilon + 2^{-S_p} + 1/2 - \sum_{\vec{z}} 2^{-m}, \qquad (6)
\end{aligned}
$$

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 the number of terms in the sum of (6) is exactly $2^{|\vec{z}|} \leq 2^{m-1}$, we have:

$$\Pr[\pi \leftarrow \Pi_{t,n} : \mathsf{Forge}_{\pi,B}] > \varepsilon + 2^{-S_p},$$

contradicting (5). $\square$

4. REFERENCES

[1] M. Bellare and P. Rogaway. Collision-resistant hashing: toward making UOWHFs practical. Crypto '97.

[2] M. Blum and S. Goldwasser. An efficient probabilistic encryption scheme which hides all partial information. Crypto '84.

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

[4] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6): 644–654 (1976).

[5] C. Dwork. The non-malleability lectures. Available from the author

[6] M. Fischlin. On the impossibility of constructing non-interactive statistically-secret protocols from any trapdoor one-way function. CT-RSA 2002.

[7] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4): 469–472 (1985).

[8] R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. FOCS 2000.

[9] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public-key encryption and oblivious transfer. FOCS 2000.

[10] Y. Gertner, T. Malkin, and O. Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. FOCS 2001.

[11] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. STOC '89.

[12] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comp. System Sciences*, 28(2): 270–299 (1984).

[13] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen message attacks. *SIAM J. Comp.*, 17(2): 281–308 (1988).

[14] R. Impagliazzo and M. Luby. One-way functions are essential for complexity-based cryptography. FOCS '89.

[15] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. STOC '89.

[16] J. Kahn, M. Saks, and C. Smyth. A dual version of Reimer's inequality and a proof of Rudich's conjecture. Computational Complexity 2000.

[17] J. H. Kim, D. Simon, and P. Tetali. Limits on the efficiency of one-way permutation-based hash functions. FOCS '99.

[18] L. Lamport. Constructing digital signatures from a one-way function. Technical Report CSL-98, SRI International, 1979.

[19] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. STOC '89.

[20] J. Rompel. One-way functions are necessary and sufficient for digital signatures. STOC '90.

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

[22] S. Rudich. The use of interaction in public cryptosystems. Crypto '91.

[23] V. Shoup. A composition theorem for universal one-way hash functions. Eurocrypt 2000.

[24] D. Simon. Finding collisions on a one-way street: can secure hash functions be based on general assumptions? Eurocrypt '98.

[25] A. C. Yao. Theory and applications of trapdoor functions. FOCS '82.