Signature Schemes with Bounded Leakage Resilience

Jonathan Katz[1]* and Vinod Vaikuntanathan[2]

[1] University of Maryland
jkatz@cs.umd.edu
[2] IBM Research
vinodv@alum.mit.edu

Abstract. A *leakage-resilient* cryptosystem remains secure even if arbitrary, but bounded, information about the secret key (and possibly other internal state information) is leaked to an adversary. Denote the length of the secret key by n. We show:

- A full-fledged signature scheme tolerating leakage of $\left(1 - n^{-\epsilon}\right) \cdot n$ bits of information about the secret key (for any constant $\epsilon < 1$), based on general assumptions.
- A one-time signature scheme, based on the minimal assumption of one-way functions, tolerating leakage of $(\frac{1}{4} - \epsilon) \cdot n$ bits of information about the signer's entire state.
- An efficient one-time signature scheme, that can be based on several specific assumptions, tolerating leakage of $(\frac{1}{2} - \epsilon) \cdot n$ bits of information about the signer's entire state.

The latter two constructions extend to give leakage-resilient t-time signature schemes. All the above constructions are in the standard model.

1 Introduction

Proofs of security for cryptographic primitives traditionally treat the primitive as a "black box" which an adversary is able to access in a relatively limited fashion. For example, in the usual model for proving security of signature schemes, an adversary is given the public key and allowed to request signatures on any messages of its choice, but is unable to get any *other* information about the secret key or any internal randomness or state information used during signature generation.

In real-world implementations of cryptographic primitives, on the other hand, an adversary may be able to recover a significant amount of additional information not captured by standard security models. Examples include information leaked by side-channel cryptanalysis [20, 21], fault attacks [5, 3], or timing attacks [4], or even bits of the secret key itself in case this key is improperly stored or erased [17]. Potentially, schemes can also be attacked when they are implemented using poor random number generation [28] (which can be viewed as

* Work done while visiting IBM.

giving the adversary additional information on the internal state, beyond what would be available if the output were truly random), or when the same key is used in multiple contexts (e.g., for decryption and signing).

In the past few years, cryptographers have made tremendous progress toward modeling security in the face of such information leakage [25, 35], and in constructing *leakage-resilient* cryptosystems secure even in case such leakage occurs. (There has also been corresponding work on reducing unwanted leakage by, e.g., building tamper-proof hardware; this is not the focus of our work.) Most relevant to the current work is a recent series of results [11, 1, 31, 9, 10, 26, 2] showing cryptosystems that guarantee security even when *arbitrary* information about the secret key is leaked (under suitable restrictions); we discuss this work, along with other related results, in further detail below. This prior work gives constructions of stream ciphers [11, 31] (and hence stateful symmetric-key encryption and MACs), symmetric-key encryption schemes [9], public-key encryption schemes [1, 10, 26], and signature schemes [2] achieving various notions of leakage resilience.

Most prior work has focused on primitives for ensuring *secrecy*. The only work of which we are aware that deals with *authenticity* is that of Alwen et al. [2] which shows, among other results, leakage-resilient signature schemes based on number-theoretic assumptions in the random oracle model.[1] Here we give constructions of leakage-resilient signature schemes based on *general assumptions* in the *standard model*; our main construction also tolerates more leakage than the schemes of [2]. (In the full version we also show some technical improvements to the results of [2].) We postpone a more thorough discussion of our results until after we define leakage resilience in more detail.

1.1 Modeling Leakage Resilience

We provide a brief overview of the framework in which leakage resilience is defined (specialized to signatures), providing a discussion of previous related work along the way.

At a high level, definitions of leakage resilience take the following form: Begin with a "standard" security notion (e.g., existential unforgeability under adaptive chosen message attacks [15]) and modify this definition by allowing the adversary to (adaptively) specify a series of *leakage functions* $f_1, \ldots$. The adversary, in addition to getting whatever other information is specified by the original security definition, is given the result of applying f_i to the secret key and possibly other internal state of the honest party (e.g., the signer). We then require that the adversary's success probability — in the case of signature schemes, the probability with which it can output a forged signature on a previously unsigned message — remain negligible. It should be clear that this is a general methodology that can be applied to many different primitives. The exact model is then determined by the restrictions placed on the leakage function(s) f_i:

[1] The results of [2] were obtained independently of our own work.

Limited vs. arbitrary information. A first consideration regards whether the $\{f_i\}$ can be arbitrary (polynomial-time computable) functions, or whether they are restricted to be in some more limited class. Early work considered the latter case, for example where the adversary is restricted to learning *specific bits* of the secret key [6], or the values on *specific wires* of the circuit implementing the primitive [19]. More recent work [11, 1, 31, 9, 10, 26, 2] allows arbitrary $\{f_i\}$.

Bounded vs. unbounded information leakage. Let n denote the length of the secret key. If the secret key does not change over time, and the $\{f_i\}$ are allowed to be arbitrary, then security in the traditional sense cannot be achieved once the total length of the *leakage* — that is, the outputs of all the $\{f_i\}$ — is n bits or more. For the case of signatures, the length of the leakage must also be less than the signature length. This inherent restriction is used in [1, 10, 26]. (Alwen et al. [2] do not impose this restriction, but as a consequence can only achieve a weaker notion of security.)

One can avoid this restriction, and potentially tolerate an unbounded amount of leakage overall, if the secret key is updated over time; even in this case, one must somehow limit the amount of leakage between successive key updates. This approach to leakage resilience was considered in [11, 31] in the context of stateful symmetric-key primitives, and [12] in the context of stateful signature schemes.

One can also avoid imposing a bound on the leakage by restricting the $\{f_i\}$, as discussed next.

Computational min-entropy of the secret key. If the leakage is much shorter than the secret key (as discussed above), then the secret key will have high min-entropy conditioned on the leakage. This setting is considered in [1, 26, 10, 2], and is also enforced on a per-period basis in the work of [11, 31] (i.e., the leakage per time period is required to be shorter than the secret key). More recent work [9, 10] shows schemes that remain secure for leakage of arbitrary length, as long as the secret key remains exponentially hard to compute given the leakage (but even if the secret key is fully determined by the leakage in an information-theoretic sense). A drawback of this guarantee is that given some collection of functions $\{f_i\}$ (say, as determined experimentally for some particular set of side-channel attacks) there is no way to tell, in general, whether they satisfy the stated requirement or not. Furthermore, existing results in this direction currently require super-polynomial hardness assumptions.

Inputs to the leakage functions. A final issue is the allowed inputs to the leakage functions. Work of [11, 31] assumes, following [25], that *only computation leaks information*; this is modeled by letting each f_i take as input only those portions of the secret key that are accessed during the ith phase of the scheme. Halderman et al. [17], however, show that memory contents can be leaked even when they are not being accessed. Motivated (in part) by this result, the schemes of [1, 9, 10, 26, 2] allow the $\{f_i\}$ to take the entire secret key as input at all times.

For the specific primitives considered in [11, 1, 31, 9, 10, 26], the secret key sk is the only internal state maintained by the party holding the secret key,

and so allowing the $\{f_i\}$ to depend on sk is (almost) the most general choice.[2] For signature schemes, however, any randomness used during signing might also be leaked to an adversary. The strongest definition of leakage resilience is thus obtained by allowing the $\{f_i\}$ to depend on *all* the state information used by the honest signer during the course of the experiment.

All these variants may be meaningful depending on the particular attacks one is trying to model. Memory attacks [17, 1], which probe long-term secret information during a time when computation is *not* taking place, can be faithfully modeled by allowing the leakage functions to take only sk as input. On the other hand, side-channel attacks that collect information while computation is occurring might be more accurately captured by allowing the leakage functions to take as input only those portions of the internal state that are being accessed.

1.2 Our Results

With the preceding discussion in mind, we can now describe our results in further detail. In all cases, we allow the leakage function(s) to be *arbitrary* as long as the total leakage is *bounded* as some function of the secret key length n; recall that such a restriction on the leakage is essential if the secret key is unchanging, as it is in all our schemes. Our results can be summarized as follows:

1. We show a construction of a leakage-resilient signature scheme that is existentially unforgeable against chosen-message attacks in the standard model, based on general (as opposed to number-theoretic) assumptions. This scheme tolerates leakage of $n - n^{\epsilon}$ bits of information about the secret key for any $\epsilon > 0$ based on polynomial hardness assumptions, and can tolerate (optimal) $n - \omega(\log n)$ bits of leakage based on sub-exponential hardness assumptions.
2. We also construct two leakage-resilient *one-time* (resp., t-time) signature schemes in the standard model. These schemes are more efficient than the scheme above; they also tolerate leakage that may depend on the entire state of the signer (rather than just the secret key).
   - Our first scheme is based on the minimal assumption that one-way functions exist, and tolerates leakage of $(\frac{1}{4} - \epsilon) \cdot n$ bits for any $\epsilon > 0$. The construction extends to give a t-time signature scheme tolerating leakage of $\Theta(n/t)$ bits.
   - Our second scheme, which can be based on various concrete assumptions, is more efficient and tolerates leakage of up to $(\frac{1}{2} - \epsilon) \cdot n$ bits for any $\epsilon > 0$. This construction also extends to give a t-time signature scheme tolerating leakage of $\Theta(n/t)$ bits.

In the full version of this work, we also discuss efficient constructions of full-fledged signature schemes based on number-theoretic assumptions (in the random oracle model) that are secure as long as the leakage is bounded by $(\frac{1}{2} - \epsilon) \cdot n$

[2] More generally, one could also allow the $\{f_i\}$ to depend on the *randomness* used to generate the (public and) secret key(s); this possibility is mentioned in [26, Section 8.2]. (For the specific schemes considered in [11, 1, 31, 9, 10, 26], however, this makes no substantive difference.)

bits for any $\epsilon > 0$. Similar schemes were discovered independently by Alwen et al. [2], but our analysis offers some advantages as compared to theirs. Specifically, we make explicit the fact that the leakage can depend on the entire state of the signer, and we allow leakage queries to depend on the random oracle.

Independent of our work, Faust et al. [12] describe a transformation from any 3-time signature scheme tolerating $\alpha(n)$ bits of leakage to a full-fledged (but stateful) signature scheme where the secret key is updated over time; the resulting scheme tolerates $\alpha(n)$ bits of leakage *between key updates*, and unbounded leakage overall. (In the transformed signature scheme, security is ensured as long as the leakage depends only on the *active portion* of the secret-key.) Applying this transformation to our constructions, we get *full-fledged* signature schemes that tolerate *unbounded leakage* (subject to the restrictions mentioned above).

1.3 Overview of Our Techniques

Our constructions all rely on the same basic idea. Roughly, we consider signature schemes with the following properties:

- A given public key pk corresponds to a set S_{pk} of *exponentially many* secret keys. Furthermore, given (sk, pk) with $sk \in S_{pk}$ it remains hard to compute any other $sk' \in S_{pk}$.
- The secret key sk used by the signer has high min-entropy (at least in a computational sense) even for an adversary who observes signatures on messages of its choice. (For our one-time scheme, this is only required to hold for an adversary who observes a single signature.)
- A signature forgery can be used to compute a secret key in S_{pk}.

To prove that any such signature scheme is leakage resilient, we show how to use an adversary $\mathcal{A}$ attacking the scheme to find distinct $sk, sk' \in S_{pk}$ given (sk, pk) (in violation of the assumed hardness of doing so). Given (sk, pk), we simply run $\mathcal{A}$ on input pk and respond to its signing queries using the given key sk. Leakage queries can also be answered using sk. If the adversary forges a signature, we extract some $sk' \in S_{pk}$; it remains only to show that $sk' \neq sk$ with high probability. Let $n = \log |S_{pk}|$ be the (computational) min-entropy of sk conditioned on pk and the signatures seen by the adversary. (We assume that all secret keys in S_{pk} are equally likely, which will be the case in our constructions.) A standard argument (cf. Lemma 1) shows that if the leakage is bounded by ℓ bits, then the conditional min-entropy of the secret key is still at least $n - \ell - t$ bits except with probability 2^{-t}. So as long as the leakage is bounded away from n, with high probability the min-entropy of sk conditioned on $\mathcal{A}$'s entire view is still at least 1. But then $sk' \neq sk$ with probability at least $1/2$. This concludes the outline of the proof. We remark, however, that various subtleties arise in the formal proofs of security.

Some existing signature schemes in the random oracle model already satisfy the requirements stated above. In particular, these include schemes constructed using the Fiat-Shamir transform [13] applied to a witness-indistinguishable Σ-protocol where there are an *exponential* number of witnesses for to a given

statement. Concrete examples include the signature schemes of Okamoto [29] (extending the Schnorr [34] and Guillou-Quisquater [16] schemes) based on the discrete logarithm or RSA assumptions, as well as the signature scheme of Fischlin and Fischlin [14] (extending the Ong-Schnorr [30] scheme) based on the hardness of factoring. This class of schemes was also considered by Alwen et al. [2]. See the full version of our paper for further discussion.

We are not aware of any existing signature scheme in the standard model that meets our requirements. We construct one as follows. Let H be a universal one-way hash function (UOWHF) [27] mapping n-bit inputs to n^ϵ-bit outputs. The secret key of the signature scheme is $x \in \{0,1\}^n$, and the public key is $(y = H(x), pk, r)$ where pk is a public key for a CPA-secure public-key encryption scheme, and r is a common reference string for an unbounded simulation-sound NIZK proof system [33, 8]. A signature on a message m consists of an encryption $C \leftarrow \mathsf{Enc}_{pk}(m\|x)$ of both m and x, along with a proof π that C is an encryption of $m\|x'$ with $H(x') = y$. Observe that, with high probability over choice of x, there are exponentially many pre-images of $y = H(x)$ and hence exponentially many valid secret keys; furthermore, finding another such secret key $sk' \neq sk$ requires finding a collision in H. Details are given in Section 3.

Our leakage-resilient one-time signature schemes are constructed using a similar idea. The first construction is inspired by the Lamport signature scheme [23]. The secret key is $\{(x_{i,0}, x_{i,1})\}_{i=1}^k$ and the public key is $\{(y_{i,0}, y_{i,1})\}_{i=1}^k$ where $y_{i,b} = H(x_{i,b})$ for H a UOWHF. Once again, there are exponentially many secret keys associated with any public key and finding any two such keys yields a collision in H. Adapting the Lamport scheme, so that the signature on a message $m = m_1 \cdots m_k$ is $\{x_{i,m_i}\}_{i=1}^k$, yields a signature scheme secure against leakage of $n^{1-\epsilon}$ bits. By first encoding the message using an error-correcting code with high minimum distance, it is possible to "boost" the leakage resilience to $(\frac{1}{4} - \epsilon) \cdot n$ bits. Using cover-free families this approach extends also to give a leakage-resilient t-time signature scheme. These constructions are all described in Section 4.

Our second construction builds on ideas that can be traced back to [7, 24]. Roughly, let $(G, \oplus)$ and $(G', \otimes)$ be groups with $\log |G'| \leq \epsilon \cdot \log |G|$, and let $\mathcal{H} = \{H_s : G \to G'\}$ be a family of collision-resistant hash functions that are also *homomorphic* (i.e., for which $H_s(a) \otimes H_s(b) = H_s(a \oplus b)$); such hash functions can be constructed based on a variety of concrete assumptions (see Section 4.3). The secret key is a pair of elements $a, b \in G$, and the public-key is $(s, H_s(a), H_s(b))$ for a random key s. Note, there are exponentially many secret keys associated with any public key and finding any two such secret keys yields a collision in H_s. The signature on a message $m \in \{1, \ldots, \mathsf{ord}(G)\}$ is simply $\sigma = a \oplus mb$, which can be verified by checking that $H_s(\sigma) \stackrel{?}{=} H_s(a) \otimes mH_s(b)$. The important property for our purposes is that given a single signature $a \oplus mb$, the secret key (a, b) still has high min-entropy. So if the adversary forges another signature σ' for a message $m' \neq m$, with high probability it holds that $\sigma' \neq a \oplus m'b$ and we obtain a collision in H_s. (This description omits various technical details; see Section 4.2.)

2 Definitions and Preliminaries

We provide a formal definition of leakage resilience for signature schemes, and state a technical lemma that will be used in our analysis. We denote the security parameter by k, and let PPT stand for "probabilistic polynomial time".

Definition 1. *A* signature scheme *is a tuple of* PPT *algorithms* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *such that:*

- Gen *is a randomized algorithm that takes as input* 1^k *and outputs* (pk, sk)*, where* pk *is the public key and* sk *is the secret key.*
- Sign *is a (possibly) randomized algorithm that takes as input the secret key* sk*, the public key* pk*, and a message* m*, and outputs a signature* σ*. We denote this by* $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$*, leaving the public key implicit.*[3]
- Vrfy *is a deterministic algorithm that takes as input a public key* pk*, a message* m*, and a purported signature* σ*. It outputs a bit* b *indicating acceptance or rejection, and we write this as* $b := \mathsf{Vrfy}_{pk}(m, \sigma)$*.*

It is required that for all k*, all* (pk, sk) *output by* $\mathsf{Gen}(1^k)$*, and all messages* m *in the message space, we have* $\mathsf{Vrfy}_{pk}(m, \mathsf{Sign}_{sk}(m)) = 1$*.*

Our definition of leakage resilience is the standard notion of existential unforgeability under adaptive chosen-message attacks [15], except that we additionally allow the adversary to specify arbitrary leakage functions $\{f_i\}$ and obtain the value of these functions applied to the secret key (and possibly other state information).

Definition 2. *Let* $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *be a signature scheme, and let* λ *be a function. Given an adversary* $\mathcal{A}$*, define the following experiment parameterized by* k*:*

1. *Choose* $r \leftarrow \{0,1\}^*$ *and compute* $(pk, sk) := \mathsf{Gen}(1^k; r)$*. Set* $\mathsf{state} := \{r\}$*.*
2. *Run* $\mathcal{A}(1^k, pk)$*. The adversary may then adaptively access a* signing oracle $\mathsf{Sign}_{sk}(\cdot)$ *and a* leakage oracle $\mathsf{Leak}(\cdot)$ *that have the following functionality:*
   - *In response to the* i*th query* $\mathsf{Sign}_{sk}(m_i)$*, this oracle chooses random* $r_i \leftarrow \{0,1\}^*$*, computes* $\sigma_i := \mathsf{Sign}_{sk}(m_i; r_i)$*, and returns* σ_i *to* $\mathcal{A}$*. It also sets* $\mathsf{state} := \mathsf{state} \cup \{r_i\}$*.*
   - *In response to the* i*th query* $\mathsf{Leak}(f_i)$ *(where* f_i *is specified as a circuit), this oracle gives* $f_i(\mathsf{state})$ *to* $\mathcal{A}$*. (To make the definition meaningful in the random oracle model, the* $\{f_i\}$ *are allowed to be oracle circuits that depend on the random oracle* H*.)*
     The $\{f_i\}$ *can be arbitrary, subject to the restriction that the total output length of all the* f_i *is at most* $\lambda(|sk|)$*.*
3. *At some point,* $\mathcal{A}$ *outputs* (m, σ)*.*

[3] Usually one assumes without loss of generality that the public key is included as part of the secret key. Since we measure leakage as a function of the secret-key length, however, we seek to minimize the size of the secret key.

$\mathcal{A}$ succeeds *if (1)* $\mathsf{Vrfy}_{pk}(m,\sigma)=1$ *and (2)* m *was not previously queried to the* $\mathsf{Sign}_{sk}(\cdot)$ *oracle. We denote the probability of this event by* $\Pr[\mathsf{Succ}^{\lambda\text{-}\mathsf{leakage}^*}_{\mathcal{A},\Pi}(k)]$. *We say* Π *is* fully λ-leakage resilient *if* $\Pr[\mathsf{Succ}^{\lambda\text{-}\mathsf{leakage}^*}_{\mathcal{A},\Pi}(k)]$ *is negligible for every* PPT *adversary* $\mathcal{A}$.

If state *is not updated after each signing query (and therefore, always contains only the randomness* r *used to generate the secret key), we denote the probability of success by* $\Pr[\mathsf{Succ}^{\lambda\text{-}\mathsf{leakage}}_{\mathcal{A},\Pi}(k)]$ *and say* Π *is* λ-leakage resilient *if* $\Pr[\mathsf{Succ}^{\lambda\text{-}\mathsf{leakage}}_{\mathcal{A},\Pi}(k)]$ *is negligible for every* PPT *adversary* $\mathcal{A}$.

Leakage resilience in the definition above corresponds to the *memory attacks* of [1] (except that we allow the leakage to depend also on the random coins used to generate the secret key). Other variations of the definition are, of course, also possible: state could include only sk (and not the random coins r used to generate it), or could include only the most recently used random coins r_i.

2.1 A Technical Lemma

Let X be a random variable taking values in $\{0,1\}^n$. The *min-entropy* of X is

$$H_\infty(X) \stackrel{\text{def}}{=} \min_{x\in\{0,1\}^n}\{-\log_2 \Pr[X=x]\}.$$

The conditional min-entropy of X given an event E is defined as:

$$H_\infty(X \mid E) \stackrel{\text{def}}{=} \min_{x\in\{0,1\}^n}\{-\log_2 \Pr[X=x \mid E]\}.$$

Lemma 1. *Let* X *be a random variable with* $H \stackrel{\text{def}}{=} H_\infty(X)$*, and fix* $\delta \in [0,H]$. *Let* f *be a function whose range has size* 2^λ*, and set*

$$Y \stackrel{\text{def}}{=} \left\{y \in \{0,1\}^\lambda \mid H_\infty(X \mid y=f(X)) \le H-\Delta\right\}.$$

Then

$$\Pr[f(X)\in Y] \le 2^{\lambda-\Delta}.$$

In words: the probability that knowledge of $f(X)$ decreases the min-entropy of X by Δ or more is at most $2^{\lambda-\Delta}$. Put differently, the min-entropy of X after observing $f(X)$ is greater than H' except with probability at most $2^{\lambda-H+H'}$.

Proof. Fix y in the range of f and $x\in\{0,1\}^n$ with $f(x)=y$. Since

$$\Pr[X=x \mid y=f(X)] = \frac{\Pr[X=x]}{\Pr[y=f(X)]},$$

we have that $y\in Y$ only if $\Pr[y=f(X)] \le 2^{-\Delta}$. The assumption regarding the range of f implies $|Y|\le 2^\lambda$, and so $\Pr[f(X)\in Y] \le 2^{\lambda-\Delta}$ as claimed.

3 A Leakage-Resilient Signature Scheme

We construct a leakage-resilient signature scheme in the standard model, following the intuition described in Section 1.2. Let (Gen_H, H) be a public-coin[4] UOWHF [27] mapping n-bit inputs to $\frac{1}{2} \cdot n^\epsilon$-bit outputs for $n = \mathsf{poly}(k)$ and $\epsilon \in (0,1)$. Let $(\mathsf{Gen}_E, \mathsf{Enc}, \mathsf{Dec})$ be a CPA-secure, dense[5] public-key encryption scheme, and let $(\ell, \mathcal{P}, \mathcal{V}, \mathcal{S}_1, \mathcal{S}_2)$ be an unbounded simulation-sound NIZK proof system [8] for the following language L:

$$L = \{(s, y, pk, m, C) : \exists x, \omega \text{ s.t. } C = \mathsf{Enc}_{pk}(x; \omega) \text{ and } H_s(x) = y\}.$$

The signature scheme is defined as follows:

Key generation: Choose random $x \leftarrow \{0,1\}^n$ and compute $s \leftarrow \mathsf{Gen}_H(1^k)$. Obliviously sample a public key pk for the encryption scheme, and choose a random string $r \leftarrow \{0,1\}^{\ell(k)}$. The public key is $(s, y := H_s(x), pk, r)$ and the secret key is x.

Signing: To sign message m using secret key x and public key (s, y, pk, r), first choose random ω and compute $C := \mathsf{Enc}_{pk}(x; \omega)$. Then compute $\pi \leftarrow \mathcal{P}_r((s, y, pk, m, C), (x, \omega))$; i.e., π is a proof that $(s, y, pk, m, C) \in L$ using witness (x, ω). The signature is (C, π).

Verification: Given a signature (C, π) on the message m with respect to the public key (s, y, pk, r), output 1 iff $\mathcal{V}_r((s, y, pk, m, C), \pi) = 1$.

Theorem 1. *Under the stated assumptions, the signature scheme above is* $(n - n^\epsilon)$*-leakage resilient.*

Proof (Sketch) Let Π denote the scheme given above, and let $\mathcal{A}$ be a PPT adversary with $\delta = \delta(k) \stackrel{\text{def}}{=} \Pr[\mathsf{Succ}^{\lambda\text{-leakage}}_{\mathcal{A},\Pi}(k)]$. We consider a sequence of experiments, and let $\Pr_i[\cdot]$ denote the probability of an event in experiment i. We abbreviate $\mathsf{Succ}^{\lambda\text{-leakage}}_{\mathcal{A},\Pi}(k)$ by Succ.

Experiment 0: This is the experiment of Definition 2. Given the public key (s, y, pk, r) defined by the experiment, Succ denotes the event that $\mathcal{A}$ outputs $(m, (C, \pi))$ where $\mathcal{V}_r((s, y, pk, m, C), \pi) = 1$ and m was never queried to the signing oracle. By assumption, we have $\Pr_0[\mathsf{Succ}] = \delta$.

Experiment 1: We introduce the following differences with respect to the preceding experiment: when setting up the public key, we now generate the common random string r of the simulation-sound NIZK by computing $(r, \tau) \leftarrow \mathcal{S}_1(1^k)$. Furthermore, signing queries are now answered as follows: to sign m, generate $C \leftarrow \mathsf{Enc}_{pk}(x)$ as before but compute π as $\pi \leftarrow \mathcal{S}_2((s, y, pk, m, C), \tau)$.

[4] For a public-coin UOWHF (cf. [18]), it is hard to find a second pre-image even given the randomness used to generate the hash key. Standard constructions of UOWHFs have this property.

[5] This means it is possible to sample a public key "obliviously," without knowing the corresponding secret key.

It follows from the (adaptive) zero-knowledge property of $(\ell, \mathcal{P}, \mathcal{V}, \mathcal{S}_1, \mathcal{S}_2)$, that the difference $|\Pr_1[\mathsf{Succ}] - \Pr_0[\mathsf{Succ}]|$ must be negligible.

Experiment 2: We modify the preceding experiment in the following way: to answer a signing query for a message m, compute $C \leftarrow \mathsf{Enc}_{pk}(0^n)$ (and then compute π as in Experiment 1). CPA-security of the encryption scheme implies that $|\Pr_2[\mathsf{Succ}] - \Pr_1[\mathsf{Succ}]|$ is negligible.

Experiment 3: We now change the way the public key is generated. Namely, instead of obliviously sampling the encryption public key pk we compute it as $(pk, sk) \leftarrow \mathsf{Gen}_E(1^k)$. Note that this is only a syntactic change and so $\Pr_3[\mathsf{Succ}] = \Pr_2[\mathsf{Succ}]$. (This assumes perfect oblivious sampling; if an obliviously generated public key and a legitimately generated public key are only computationally indistinguishable, then the probability of Succ is affected by a negligible amount.)

Given the public key (s, y, pk, r) defined by the experiment, let Ext be the event that $\mathcal{A}$ outputs $(m, (C, \pi))$ such that the event Succ occurs and furthermore, $H_s(\mathsf{Dec}_{sk}(C)) = y$. Unbounded simulation soundness of the NIZK proof system implies that $|\Pr_3[\mathsf{Ext}] - \Pr_3[\mathsf{Succ}]|$ is negligible. (Note that by definition of L the message m is included as part of the statement being proved, and so if $\mathcal{A}$ did not request a signature on m then it was never given a simulated proof of the statement (s, y, pk, m, C).)

To complete the proof, we show that $\Pr_3[\mathsf{Ext}]$ is negligible. Consider the following adversary $\mathcal{B}$ finding a second preimage in the UOWHF: $\mathcal{B}$ chooses random $x \leftarrow \{0,1\}^n$ and is given key s (along with the randomness used to generate s). $\mathcal{B}$ then runs Experiment 3 with $\mathcal{A}$. In this experiment all signatures given to $\mathcal{A}$ are simulated (as described in Experiment 3 above); furthermore $\mathcal{B}$ can easily answer any leakage queries made by $\mathcal{A}$ since $\mathcal{B}$ knows a legitimate secret key. (Recall that here we allow the leakage functions to be applied only to [the randomness used to generate] the secret key, but not to any auxiliary state used during signing.) If event Ext occurs when $\mathcal{A}$ terminates, then $\mathcal{B}$ recovers a value $x' \stackrel{\text{def}}{=} \mathsf{Dec}_{sk}(C)$ for which $H_s(x') = y = H_s(x)$; i.e., $\mathcal{B}$ recovers such an x' with probability exactly $\Pr_3[\mathsf{Ext}]$. We now argue that $x' \neq x$ with high probability.

The only information about x revealed to $\mathcal{A}$ in Experiment 3 comes from the value y included in the public key and the leakage queries asked by $\mathcal{A}$; these total at most $\frac{1}{2} \cdot n^\epsilon + (n - n^\epsilon) = n - \frac{1}{2} \cdot n^\epsilon$ bits. Using Lemma 1 with $\Delta = H_\infty(x) = n$, the probability that $H_\infty(x \mid \mathcal{A}\text{'s view}) = 0$ (i.e., the probability that x is uniquely determined by the view of $\mathcal{A}$) is at most $2^{-n^\epsilon/2}$, which is negligible. When the conditional min-entropy of x is greater than 0 there are at least two (equally probable) possibilities for x and so $x' \neq x$ with probability at least $\frac{1}{2}$. Taken together, the probability that $\mathcal{B}$ recovers $x' \neq x$ with $H_s(x') = H_s(x)$ is at least

$$\frac{1}{2} \cdot \left(\Pr_3[\mathsf{Ext}] - 2^{-n^\epsilon/2} \right).$$

We thus see that if $\Pr_3[\mathsf{Ext}]$ is not negligible then $\mathcal{B}$ violates the security of the UOWHF with non-negligible probability, a contradiction. $\square$

If we are willing to rely on sub-exponential hardness assumptions, we can construct a UOWHF with $\omega(\log n)$-bit outputs. In that case, it is easy to see that the same signature scheme tolerates (optimal) leakage of $n - \omega(\log n)$ bits.

4 Fully Leakage-Resilient Bounded-Use Signature Schemes

In this section we describe constructions of fully leakage-resilient one-time and t-time signature schemes. These results are incomparable to the result of the previous section: on the positive side, here we achieve *full* leakage resilience (that is, when the leakage depends not only on the secret-key, but also on the randomness used by the signer) as well as better efficiency (and, in one case, rely on weaker assumptions); on the downside, the schemes given here are only secure when the adversary obtains a bounded number of signatures, and the leakage that can be tolerated is lower.

4.1 A Construction Based on One-Way Functions

We describe a basic one-time signature scheme, and then present an extension that tolerates leakage of up to a constant fraction of the secret key length. Let (Gen_H, H) be a UOWHF mapping k^c-bit inputs to k-bit outputs for some $c > 1$. (As before, we assume that H is a public-coin UOWHF, i.e, H is secure even given the randomness used to generate the hash key.) Our basic scheme is a variant on Lamport's signature scheme [23], using H as the one-way function:

Key generation: Choose random $x_{i,0}, x_{i,1} \leftarrow \{0,1\}^{k^c}$ for $i = 1, \ldots, k$, and generate $s \leftarrow \mathsf{Gen}_H(1^k)$. Compute $y_{i,b} := H_s(x_{i,b})$ for $i \in \{1, \ldots, k\}$ and $b \in \{0,1\}$. The public key is $(s, \{y_{i,b}\})$ and the secret key is $\{x_{i,b}\}$.

Signing: The signature on a k-bit message $m = m_1 \cdots m_k$ consists of the k values $x_{1,m_1}, \ldots, x_{k,m_k}$.

Verification: Given a signature $x_1, \ldots, x_k$ on the k-bit message $m = m_1 \cdots m_k$ with respect to the public key $(s, \{y_{i,b}\})$, output 1 iff $y_{i,m_i} \stackrel{?}{=} H_s(x_i)$ for all i.

It can be shown that the above scheme is fully $n^{(c-1)/(c+1)}$-leakage resilient (as a one-time signature scheme), where $n = 2k^{c+1}$ denotes the length of the secret key. Setting c appropriately, the above approach thus tolerates leakage $n^{1-\epsilon}$ for any desired $\epsilon > 0$. (We omit the proof, since we will prove security for an improved scheme below.) Note that the bound on the leakage is essentially tight, since an adversary who obtains the signature on the message 0^k and then leaks the value $x_{1,1}$ (which is only $k^c = (n/2)^{c/(c+1)}$ bits) can forge a signature on the message 10^{k-1}.

Tolerating leakage linear in the secret key length. An extension of the above scheme allows us to tolerate greater leakage: specifically, we apply Lamport's scheme to a high-distance *encoding* of the message. Details follow.

If A is a $k \times \ell$ matrix over $\{0,1\}$ (viewed as the field $\mathbb{F}_2$), then A defines a (linear) error-correcting code $\mathcal{C} \subset \{0,1\}^\ell$ where the message $m \in \{0,1\}^k$ (viewed as a row vector) is mapped to the codeword $m \cdot A$. It is well known that for every $\epsilon > 0$ there exists a constant R such that choosing $A \in \{0,1\}^{k \times Rk}$ uniformly at random defines a code with relative minimum distance $\frac{1}{2} - \epsilon$, except with probability negligible in k. (We will not need efficient decodability.)

Fix a constant $\epsilon \in (0,1)$ and let R be as above; set $\ell = Rk$. Let (Gen_H, H) be a UOWHF mapping ℓ_{in}-bit inputs to k-bit outputs where $\ell_{in} = 2k/\epsilon$. The signature scheme is defined as:

Key generation: Choose random $A \in \{0,1\}^{k \times \ell}$ and $x_{i,0}, x_{i,1} \leftarrow \{0,1\}^{\ell_{in}}$ for $i = 1, \ldots, \ell$. Generate $s \leftarrow \mathsf{Gen}_H(1^k)$. Compute $y_{i,b} := H_s(x_{i,b})$ for $i \in \{1, \ldots, \ell\}$ and $b \in \{0,1\}$. The public key is $(A, s, \{y_{i,b}\})$ and the secret key is $\{x_{i,b}\}$.

Signing: To sign a message $m \in \{0,1\}^k$, first compute $\bar{m} = m \cdot A \in \{0,1\}^\ell$. The signature then consists of the ℓ values $x_{1,\bar{m}_1}, \ldots, x_{\ell,\bar{m}_\ell}$.

Verification: Given a signature $x_1, \ldots, x_\ell$ on the message m with respect to the public key $(A, s, \{y_{i,b}\})$, first compute $\bar{m} = m \cdot A$ and then output 1 iff $y_{i,\bar{m}_i} \stackrel{?}{=} H_s(x_i)$ for all i.

Theorem 2. *If H is a UOWHF then the scheme above is a one-time signature scheme that is fully $(\frac{1}{4} - \epsilon) \cdot n$-leakage resilient, where $n = 2\ell \cdot \ell_{in}$ denotes the length of the secret key.*

Proof. Let Π denote the scheme given above, and let $\mathcal{A}$ be a PPT adversary with $\delta = \delta(k) \stackrel{\text{def}}{=} \Pr[\mathsf{Succ}^{\lambda\text{-}\mathsf{leakage}^*}_{\mathcal{A},\Pi}(k)]$. We construct an adversary $\mathcal{B}$ breaking the security of H with probability at least $(\delta - \mathsf{negl}(k))/4\ell$, implying that δ must be negligible.

$\mathcal{B}$ chooses random $A \in \{0,1\}^{k \times \ell}$ and $x_{i,0}, x_{i,1} \leftarrow \{0,1\}^{\ell_{in}}$ for $i = 1, \ldots, \ell$; we let $\mathcal{X} = \{x_{i,b}\}$ denote the set of secret key values $\mathcal{B}$ chooses and observe that $H_\infty(\mathcal{X}) = 2\ell \cdot \ell_{in}$. Next, $\mathcal{B}$ selects a random $b^* \in \{0,1\}$ and a random index $i^* \in \{1, \ldots, \ell\}$, and outputs x_{i^*,b^*}; it is given in return a hash key s. Then $\mathcal{B}$ computes $y_{i,b} := H_s(x_{i,b})$ for all i, b and gives the public key $(A, s, \{y_{i,b}\})$ to $\mathcal{A}$.

$\mathcal{B}$ answers the signing and leakage queries of $\mathcal{A}$ using the secret key $\{x_{i,b}\}$ that it knows. Since this secret key is distributed identically to the secret key of an honest signer, the simulation for $\mathcal{A}$ is perfect and $\mathcal{A}$ outputs a forgery with probability δ.

Let $\bar{m}$ denote the encoding of the message m whose signature was requested by $\mathcal{A}$. The information $\mathcal{A}$ has about the secret-key $\mathcal{X}$ consists of: (1) the signature $(x_{1,\bar{m}_1}, \ldots, x_{\ell,\bar{m}_\ell})$ it obtained; (2) the values $\{y_{i,1-\bar{m}_i}\}_{i=1}^{\ell}$ from the public key and (3) the answers to the leakage queries asked by $\mathcal{A}$. Together, these total $\ell \cdot \ell_{in} + \ell k + (\frac{1}{4} - \epsilon) \cdot 2\ell \cdot \ell_{in}$ bits. By Lemma 1, it follows that $H_\infty(\mathcal{X} \mid \mathcal{A}\text{'s view}) > (\frac{1}{2} + \epsilon) \cdot \ell \cdot \ell_{in}$ except with probability at most

$$2^{\left(\ell \cdot \ell_{in} + \ell k + (\frac{1}{2} - 2\epsilon)\ell \cdot \ell_{in}\right) - 2\ell \cdot \ell_{in} + (\frac{1}{2} + \epsilon) \cdot \ell \cdot \ell_{in}} = 2^{\ell k - \epsilon \ell \cdot \ell_{in}},$$

which is negligible.

Assuming $H_\infty(\mathcal{X} \mid \mathcal{A}\text{'s view}) > (\frac{1}{2}+\epsilon)\cdot\ell\cdot\ell_{in}$, there is *no* set $I \subseteq [\ell]$ with $|I| \geq (\frac{1}{2}-\epsilon)\cdot\ell$ for which the values $\{x_{i,1-\bar{m}_i}\}_{i\in I}$ are all fixed given $\mathcal{A}$'s view. To see this, assume the contrary. Then

$$H_\infty(\mathcal{X} \mid \mathcal{A}\text{'s view}) \leq \sum_{i\notin I} H_\infty(x_{i,1-\bar{m}_i} \mid \mathcal{A}\text{'s view}) \leq \left(\frac{1}{2}+\epsilon\right)\ell\cdot\ell_{in},$$

in contradiction to the assumed bound on the conditional min-entropy of $\mathcal{X}$.

Let $(m^*, (x_1^*, \ldots, x_\ell^*))$ denote the forgery output by $\mathcal{A}$, and let $\bar{m}^* = m^* \cdot A$ denote the encoding of m^*. Let I be the set of indices where $\bar{m}$ and $\bar{m}^*$ differ; with all but negligible probability over choice of the matrix A it holds that $|I| \geq (\frac{1}{2}-\epsilon)\cdot\ell$ and so we assume this to be the case. By the argument of the previous paragraph, it cannot be the case that the $\{x_{i,1-\bar{m}_i}\}_{i\in I}$ are all fixed given $\mathcal{A}$'s view. But then with probability at least half we have $x_i^* \neq x_{i,\bar{m}_i^*}$ for at least one index $i \in I$. Assuming this to be the case, with probability at least $1/2\ell$ this difference occurs at the index (i^*, b^*) guessed at the outset by $\mathcal{B}$; when this happens $\mathcal{B}$ has found a collision in H for the given hash key s. Putting everything together, we see that $\mathcal{B}$ finds a collision in H with probability at least $(\delta - \mathsf{negl}(k)) \cdot \frac{1}{2} \cdot \frac{1}{2\ell}$, as claimed.

A t-time signature scheme. The idea above can be further extended to give a fully leakage resilient t-time signature scheme using *cover-free families*. We follow the definition of [22].

Definition 3. *A family of non-empty sets $\mathcal{S} = \{S_1, \ldots, S_N\}$, where $S_i \subset U$, is* $(t, \frac{1}{2})$-cover-free *if for all distinct $S, S_1, \ldots, S_t \in \mathcal{S}$ we have $\left|S \setminus \bigcup_{i=1}^t S_i\right| \geq |S|/2$.*

Porat and Rothschild [32] show an explicit construction that, for any t and k, yields a $(t, \frac{1}{2})$-cover free family $\mathcal{S} = \{S_1, \ldots, S_N\}$ where the number of sets is $N = \Omega(2^k)$, the size of each set is $|S_i| = \mathcal{O}(kt)$, and the universe size is $|U| = \mathcal{O}(kt^2)$. If we let $f : \{0,1\}^k \to \mathcal{S}$ denote an injective map, we obtain the following scheme:

Key generation: Set $\ell = \mathcal{O}(kt^2)$ and $\ell_{in} = 8tk$. Choose $x_i \leftarrow \{0,1\}^{\ell_{in}}$ for $i = 1, \ldots, \ell$. Generate $s \leftarrow \mathsf{Gen}_H(1^k)$, and compute $y_i := H_s(x_i)$ for $i \in \{1, \ldots, \ell\}$. The public key is $(s, \{y_i\}_{i=1}^\ell)$ and the secret key is $\{x_i\}_{i=1}^\ell$.

Signing: To sign a message $m \in \{0,1\}^k$, first compute $f(m) = S_m \in \mathcal{S}$. The signature then consists of $\{x_i\}_{i\in S_m}$.

Verification: Given a signature $\{x_i\}$ on the message m with respect to the public key $(s, \{y_{i,b}\})$, first compute $S_m = f(m)$ and then output 1 iff $y_i \stackrel{?}{=} H_s(x_i)$ for all $i \in S_m$.

A proof of the following proceeds along exactly the same lines as the proof of Theorem 2:

Theorem 3. *If H is a UOWHF then the scheme above is a t-time signature scheme that is fully $\Theta(n/t)$-leakage resilient, where $n = \ell \cdot \ell_{in}$ denotes the length of the secret key.*

4.2 A Construction from Homomorphic Collision-Resistant Hashing

Our second construction of fully leakage-resilient bounded-use signature schemes relies on homomorphic collision-resistant hash functions, defined below. Here, we concentrate on the case of one-time signatures, and defer a treatment of t-time signatures to the full version of this work. In Section 4.3, we describe efficient instantiations of the hash functions we need based on several concrete assumptions.

Definition 4. *Fix $\epsilon \in (0,1)$. A pair of* PPT *algorithms (Gen_H, H) is an* ϵ-homomorphic collision-resistant hash function family *(ϵ-hCRHF) if:*

1. *$\mathsf{Gen}_H(1^k)$ outputs a key s that specifies groups $(G,\oplus),(G',\otimes)$ (written additively), and two sets $S,T \subseteq G$ such that*
   - *$\log|S| = \omega(\log k)$ and $\log|G'| \leq \epsilon \cdot \log|S|$ and $\log|T| \leq (1+\epsilon)\log|S|$.*
   - *S is efficiently sampleable, and elements of S can be represented using $\log|S| + \mathcal{O}(1)$ bits.*
   - *T is efficiently recognizable, and $\{x + my \mid x, y \in S,\ 0 \leq m < 2^k\} \subseteq T$.*
2. *The key s defines a function $H_s : G \to G'$ with $H_s(x \oplus y) = H_s(x) \otimes H_s(y)$ for all $x, y \in G$.*
3. *There exists a constant c (independent of k) for which the following holds. For any s, any m, m' with $0 \leq m < m' < 2^k$, and any σ, σ':*

$$\left| \{x, y \in S \mid H_s(x + my) = \sigma \wedge H_s(x + m'y) = \sigma'\} \right| \leq 2^c.$$

4. *No* PPT *algorithm $\mathcal{A}$ can find two elements $x, y \in T$ such that $H_s(x) = H_s(y)$. Namely, the following is negligible for all* PPT *$\mathcal{A}$:*

$$\Pr[s \leftarrow \mathsf{Gen}_H(1^k); (x,y) \leftarrow \mathcal{A}(s) : x, y \in T_k \wedge x \neq y \wedge H_s(x) = H_s(y)].$$

   If the above holds even when $\mathcal{A}$ is given the randomness used to generate s, then (Gen_H, H) is a strong *ϵ-hCRHF.*

Define a signature scheme as follows.

Key generation: Compute $s \leftarrow \mathsf{Gen}_H(1^k)$; this specifies groups $(G,\oplus),(G',\otimes)$ and sets S, T. Choose x, y uniformly at random from S. Output $sk := (x, y)$ and $pk := (s, H_s(x), H_s(y))$.

Signing: The scheme is defined for messages m satisfying $0 \leq m < 2^k$. Given m, output the signature $\sigma := x \oplus my$.

Verification: Given a signature σ on the message m with respect to the public key $pk = (s, a, b)$, output 1 iff $\sigma \in T$ and $H_s(\sigma) \stackrel{?}{=} a \otimes mb$.

Theorem 4. *If (Gen_H, H) is a (strong) ϵ-hCRHF, then the above is a one-time signature scheme that is (fully) $\left(\frac{1}{2} - 2\epsilon\right) \cdot n$-leakage resilient.*

Proof. Correctness is easily verified. Let Π denote the scheme given above, and let $\mathcal{A}$ be a PPT adversary with $\delta = \delta(k) \stackrel{\text{def}}{=} \Pr[\mathsf{Succ}^{\lambda\text{-}\mathsf{leakage}^*}_{\mathcal{A},\Pi}(k)]$. We construct an adversary $\mathcal{B}$ breaking the security of (Gen_H, H) with probability at least $\delta/2 - \mathsf{negl}(k)$, implying that δ must be negligible.

$\mathcal{B}$ is given as input a key s (along with the randomness used to generate it). $\mathcal{B}$ chooses $x, y \in S$, sets $sk := (x, y)$, and gives the public key $pk := (s, H_s(x), H_s(y))$ to $\mathcal{A}$. Algorithm $\mathcal{B}$ then answers the signing and leakage queries of $\mathcal{A}$ using the secret key (x, y) that it knows. Since this secret key is distributed identically to the secret key of an honest signer, the simulation for $\mathcal{A}$ is perfect and $\mathcal{A}$ outputs a valid forgery (m', σ') with probability δ. If this occurs, then $\mathcal{B}$ outputs $(\sigma', x \oplus m'y)$ as a candidate collision for H_s.

Note that $x \oplus m'y \in T$. If σ' is a valid signature on m', we have $\sigma' \in T$ and

$$H_s(\sigma') = H_s(x) \otimes m' H_s(y) = H_s(x \oplus m'y).$$

It remains to show that $\sigma' \neq x \oplus m'y$ with significant probability.

Let c be the constant guaranteed to exist by condition 3 of Definition 4. The length of the secret key is $n \stackrel{\text{def}}{=} 2\log|S|$ bits.[6] The information $\mathcal{A}$ has about $sk = (x, y)$ consists of: (1) the signature $x \oplus my$ it obtained; (2) the values $H_s(x), H_s(y)$ from the public key; and (3) the answers to the leakage queries asked by $\mathcal{A}$. These total at most

$$\begin{aligned}\log|T| + 2\log|G'| + \left(\frac{1}{2} - 2\epsilon\right) 2\log|S| &\leq (1+\epsilon)\log|S| + 2\epsilon\log|S| \\ &\quad + \log|S| - 4\epsilon\log|S| \\ &= 2\log|S| - \epsilon\log|S|\end{aligned}$$

bits of information about sk. The min-entropy of sk is $2\log|S|$ bits, so by Lemma 1 it follows that $H_\infty(sk \mid \mathcal{A}\text{'s view}) \geq c + 1$ except with probability at most $2^{-\epsilon\log|S|+c+1}$, which is negligible.

Assuming $H_\infty(sk \mid \mathcal{A}\text{'s view}) \geq c + 1$, we claim that for any $m' \neq m$ (with $0 \leq m' < 2^k$) the value $x \oplus m'y$ has min-entropy at least 1; this follows from the fact that, for any fixed $\hat{\sigma}'$, the two equations $\sigma = x \oplus my$ and $\hat{\sigma}' = x \oplus m'y$ constrain (x, y) to a set of size at most 2^c (by condition 3 of Definition 4). Thus, $\sigma' = x \oplus m'y$ with probability at most $1/2$. Putting everything together, we see that $\mathcal{B}$ finds a collision in H_s with probability at least $(\delta - \mathsf{negl}(k)) \cdot \frac{1}{2}$ as claimed.

4.3 Constructing (Strong) Homomorphic CRHFs

Homomorphic CRHFs can be constructed from a variety of standard assumptions. Here, we describe constructions based on the discrete logarithm and the RSA assumptions; in the full version, we show a construction based on lattices. All except the RSA-based construction are *strong* ϵ-hCRHFs.

[6] We assume for simplicity that elements of S can be described using exactly $\log|S|$ bits; the proof can be modified suitably if this is not the case.

An instantiation based on the discrete logarithm assumption. Let G' be a group of prime order $p > 2^k$ where the discrete logarithm problem is hard. Let $\ell = \lceil \frac{1}{\epsilon} \rceil$, and set $S = T = G = \mathbb{Z}_p^\ell$.

The key-generation algorithm Gen_H outputs random $g_1, \ldots, g_\ell \in G$ as the key. Given $s = (g_1, \ldots, g_\ell)$, define $H_s(x_1, \ldots, x_\ell) = \prod_{i=1}^{\ell} g_i^{x_i}$. This function is clearly homomorphic, and collision resistance follows by standard arguments.

An instantiation based on the RSA assumption. Fix $\ell = \lceil \frac{2}{\epsilon} \rceil$. On security parameter k, algorithm $\mathsf{Gen}_H(1^k)$ chooses safe primes $p = 2p' + 1$ and $q = 2q' + 1$ with $p', q' > 2^k$, and sets $N = pq$. (The primes p and q are not used after key generation, but because they are in memory during key generation this construction is not strong.) Gen_H then chooses a random element $u \in \mathbb{Z}_N^*$, as well as a prime $e > 2^{(\ell+1)\cdot k}$. The key is $s = (N, e, u)$.

Let $G = \mathbb{Z}_N^* \times \mathbb{Z}$ and $G' = \mathbb{Z}_N^*$. Define

$$H_s(r, x) = r^e \cdot u^x \bmod N.$$

Take $S = \mathcal{QR}_N \times \{0, \ldots, 2^{\ell k}\} \subset G$ (where $\mathcal{QR}_N$ denotes the set of quadratic residues modulo N) and $T = \mathbb{Z}_N^* \times \{0, \ldots, 2^{(\ell+1)\cdot k}\}$.

The homomorphic property of H_s is easy to see. One can also verify that:

1. $\log |S| = \omega(\log k)$ and $\log |G'| \leq \epsilon \cdot \log |S|$ and $\log |T| \leq (1 + \epsilon) \log |S|$.
2. T is efficiently recognizable, and $\{x + my \mid x, y \in S, 0 \leq m < 2^k\} \subseteq T$.
3. For any s, any m, m' with $0 \leq m < m' < 2^k$, and any σ, σ':

$$\left| \{x, y \in S \mid H_s(x + my) = \sigma \wedge H_s(x + m'y) = \sigma'\} \right| \leq 1.$$

   (This uses the fact that $\mathcal{QR}_N \simeq \mathbb{Z}_{p'} \times \mathbb{Z}_{q'}$ has no elements other than the identity whose order is less than 2^k.)

Collision resistance follows via standard arguments (e.g., [29]).

References

1. A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *6th Theory of Cryptography Conference — TCC 2009*, volume 5444 of *LNCS*, pages 474–495. Springer, 2009.
2. J. Alwen, Y. Dodis, and D. Wichs. Public key cryptography in the bounded retrieval model and security against side-channel attacks. In *Advances in Cryptology — Crypto 2009 (to appear)*, volume ??? of *LNCS*, pages ???–??? Springer, 2009.
3. E. Biham, Y. Carmeli, and A. Shamir. Bug attacks. In *Advances in Cryptology — Crypto 2008*, volume 5157 of *LNCS*, pages 221–240. Springer, 2008.
4. D. Boneh and D. Brumley. Remote timing attacks are practical. *Computer Networks*, 48(5):701–716, 2005.
5. D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults. In *Advances in Cryptology — Eurocrypt '97*, volume 1233 of *LNCS*, pages 37–51. Springer, 1997.

6. R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, and A. Sahai. Exposure-resilient functions and all-or-nothing transforms. In *Advances in Cryptology — Eurocrypt 2000*, volume 1807 of *LNCS*, pages 453–469. Springer, 2000.
7. R. Cramer and I. Damgård. Secure signature schemes based on interactive protocols. In *Advances in Cryptology — Crypto '95*, volume 963 of *LNCS*, pages 297–310. Springer, 1995.
8. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero knowledge. In *Advances in Cryptology — Crypto 2001*, volume 2139 of *LNCS*, pages 566–598. Springer, 2001.
9. Y. Dodis, Y. Kalai, and S. Lovett. On cryptography with auxiliary input. In *41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 621–630. ACM Press, 2009.
10. Y. Dodis, Y. Kalai, and V. Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. Manuscript, 2009.
11. S. Dziembowski and K. Pietrzak. Leakage-resilient cryptography. In *49th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 293–302. IEEE, 2008. Full version available at `http://eprint.iacr.org/2008/240`.
12. S. Faust, E. Kiltz, K. Pietrzak, and G. Rothblum. Leakage-resilient signatures, 2009. Available at `http://eprint.iacr.org/2009/282`.
13. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology — Crypto '86*, volume 263 of *LNCS*, pages 186–194. Springer, 1987.
14. M. Fischlin and R. Fischlin. The representation problem based on factoring. In *Cryptographers' Track — RSA 2002*, volume 2271 of *LNCS*, pages 96–113. Springer, 2002.
15. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.
16. L. C. Guillou and J.-J. Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In *Advances in Cryptology — Crypto '88*, volume 403 of *LNCS*, pages 216–231. Springer, 1990.
17. A. Halderman, S. Schoen, N. Heninger, W. Clarkson, W. Paul, J. Calandrino, A. Feldman, J. Applebaum, and E. Felten. Lest we remember: Cold boot attacks on encryption keys. In *Proc. 17th USENIX Security Symposium*, pages 45–60. USENIX Association, 2008.
18. C.-Y. Hsiao and L. Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In *Advances in Cryptology — Crypto 2004*, volume 3152 of *LNCS*, pages 92–105. Springer, 2004.
19. Y. Ishai, A. Sahai, and D. Wagner. Private circuits: Securing hardware against probing attacks. In *Advances in Cryptology — Crypto 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, 2003.
20. P. C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology — Crypto '96*, volume 1109 of *LNCS*, pages 104–113. Springer, 1996.
21. P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Advances in Cryptology — Crypto '99*, volume 1666 of *LNCS*, pages 388–397. Springer, 1999.
22. R. Kumar, S. Rajagopalan, and A. Sahai. Coding constructions for blacklisting problems without computational assumptions. In *Advances in Cryptology — Crypto '99*, volume 1666 of *LNCS*, pages 609–623. Springer, 1999.
23. L. Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, Oct. 1979.

24. V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In *5th Theory of Cryptography Conference — TCC 2008*, volume 4948 of *LNCS*, pages 37–54. Springer, 2008.
25. S. Micali and L. Reyzin. Physically observable cryptography. In *1st Theory of Cryptography Conference — TCC 2004*, volume 2951 of *LNCS*, pages 278–296. Springer, 2004.
26. M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. In *Advances in Cryptology — Crypto 2009 (to appear)*, volume ??? of *LNCS*, pages ???–??? Springer, 2009. Available at `http://eprint.iacr.org/2009/105`.
27. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 33–43. ACM Press, 1989.
28. P. Q. Nguyen and I. Shparlinski. The insecurity of the digital signature algorithm with partially known nonces. *Journal of Cryptology*, 15(3):151–176, 2002.
29. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Advances in Cryptology — Crypto '92*, volume 740 of *LNCS*, pages 31–53. Springer, 1993.
30. H. Ong and C.-P. Schnorr. Fast signature generation with a Fiat-Shamir-like scheme. In *Advances in Cryptology — Eurocrypt '90*, volume 473 of *LNCS*, pages 432–440. Springer, 1990.
31. K. Pietrzak. A leakage-resilient mode of operation. In *Advances in Cryptology — Eurocrypt 2009*, volume 5479 of *LNCS*, pages 462–482. Springer, 2009.
32. E. Porat and A. Rothschild. Explicit non-adaptive combinatorial group testing schemes. In *Intl. Colloquium Automata, Languages, and Programming (ICALP), Part I*, volume 5125 of *LNCS*, pages 748–759. Springer, 2008.
33. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 543–553. IEEE, 1999.
34. C.-P. Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptology — Crypto '89*, volume 435 of *LNCS*, pages 239–252. Springer, 1990.
35. F.-X. Standaert, T. Malkin, and M. Yung. A unified framework for the analysis of side-channel key recovery attacks. In *Advances in Cryptology — Eurocrypt 2009*, volume 5479 of *LNCS*, pages 443–461. Springer, 2009.