Fully Leakage-Resilient Signatures

Elette Boyle[1,⋆], Gil Segev[2,⋆⋆], and Daniel Wichs[3,⋆⋆⋆]

[1] Massachusetts Institute of Technology, Cambridge, MA 02139, USA
eboyle@mit.edu
[2] Microsoft Research, Mountain View, CA 94043, USA
gil.segev@microsoft.com
[3] New York University, New York, NY 10012, USA
wichs@cs.nyu.edu

Abstract. A signature scheme is *fully leakage resilient* (Katz and Vaikuntanathan, ASIACRYPT '09) if it is existentially unforgeable under an adaptive chosen-message attack even in a setting where an adversary may obtain bounded (yet arbitrary) leakage information on *all intermediate values that are used throughout the lifetime of the system.* This is a strong and meaningful notion of security that captures a wide range of side-channel attacks.

One of the main challenges in constructing fully leakage-resilient signature schemes is dealing with leakage that may depend on the random bits used by the signing algorithm, and constructions of such schemes are known only in the random-oracle model. Moreover, even in the random-oracle model, known schemes are only resilient to leakage of less than half the length of their signing key.

In this paper we construct *fully* leakage-resilient signature schemes without random oracles. We present a scheme that is resilient to any leakage of length $(1 - o(1))L$ bits, where L is the length of the signing key. Our approach relies on generic cryptographic primitives, and at the same time admits rather efficient instantiations based on specific number-theoretic assumptions. In addition, we show that our approach extends to the continual-leakage model, recently introduced by Dodis, Haralambiev, Lopez-Alt and Wichs (FOCS '10), and by Brakerski, Tauman Kalai, Katz and Vaikuntanathan (FOCS '10). In this model the signing key is allowed to be refreshed, while its corresponding verification key remains fixed, and the amount of leakage is assumed to be bounded only in between any two successive key refreshes.

[⋆] Research supported by the US National Defense Science and Engineering Graduate Fellowship. This work was partially completed while visiting the Weizmann Institute of Science.

[⋆⋆] This work was partially completed while the author was a Ph.D. student at the Weizmann Institute of Science, and supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities.

[⋆⋆⋆] This work was partially completed while visiting the Weizmann Institute of Science.

K.G. Paterson (Ed.): Eurocrypt 2011, LNCS 6632, pp. 89–108, 2011.

1 Introduction

One of the main goals of research in the foundations of cryptography is designing systems that withstand adversarial behavior. Given a cryptographic task, such as public-key encryption, one must formalize an attack model specifying a class of adversaries, and define a notion of security capturing what it means to break the system. Within such a framework, it is then possible to rigorously analyze the security of cryptographic systems.

Starting with the seminal work of Goldwasser and Micali [18], various and increasingly strong attack models and notions of security have been proposed. Over the years, however, theoreticians and practitioners began to notice that a large class of realistic attacks, called *side-channel attacks*, are not captured by the existing models. In such attacks, the adversary may learn some additional information about the internal secret state of a system, by measuring various properties resulting from specific *physical* implementations (e.g., timing information, detection of internal faults, electromagnetic radiation, power consumption etc.). As a result, it has become an important research agenda to extend the standard models to capture such side-channel attacks, and to design cryptographic systems whose security guarantees can be rigorously analyzed and clearly stated in these stronger models. Our work focuses on the model of *memory attacks*, and its *bounded-leakage* and *continual-leakage* variants, which we describe next (several other models are described in the full version).

Memory attacks: bounded-leakage and continual-leakage. The model of *memory attacks* was introduced by Akavia, Goldwasser, and Vaikuntanathan [1]. Its main premise is that the adversary can learn *arbitrary* information about the secret state of a system, subject only to the constraint that the *amount* of information learned is somehow bounded. More precisely, the adversary can adaptively select *arbitrary poly-time computable* functions $f_i : \{0,1\}^* \rightarrow \{0,1\}^{\lambda_i}$ and learn the value of f_i applied to the internal state of the system, subject only to some constraint on the output sizes λ_i.

The work of [1] assumes that there is an a priori determined *leakage bound* λ, which bounds the *overall* amount of information learned by the adversary throughout the entire lifetime of the system to be $\sum_i \lambda_i \leq \lambda$. We call this the *bounded leakage model.* Usually the leakage bound λ is also related to the secret-key size, so that a *relatively* large fraction $\lambda/|sk|$ of the secret key can be leaked. A great deal of research has gone into devising various cryptographic primitives in this model, such as public-key and identity-based encryption schemes, signature schemes, and more (see [30,26,3,2,28,8,14]).

A drawback of the bounded-leakage model is that, if a system is being used continually for a sufficiently long time, then the amount of leakage observed by the attacker may exceed any a-priori determined leakage bound. Hence, we would like to bound the *rate* of leakage rather than the *overall amount* of leakage. If we do not bound the overall leakage, then any static piece of information that stays unmodified on the system can eventually be fully recovered by the adversary. Hence the secret keys of such systems must be periodically *refreshed.* Recently, Dodis et al. [13] and Brakerski et al. [10] suggested the *continual-leakage model,* in which a scheme periodically *self-refreshes* its internal secret key, while the corresponding public key remains fixed. In this model, only the amount of leakage seen by the adversary *in between any two successive refreshes* is assumed to be a priori bounded by some leakage bound λ[1]. However, there is no a-priori bound on the overall amount of information seen by the adversary throughout the lifetime of the system.

We note that in both the bounded-leakage model and the continual-leakage model the adversary may be able to learn partial, but yet *arbitrary*, information on the *entire* secret key. This is in contrast with other models, where either the leakage is assumed to be of "low complexity" (such as AC^0 circuits) [25,16], or certain secret values are assumed to be leak-free.

Leakage-resilient signature schemes. In this paper we study the security of signature schemes in the bounded-leakage and continual-leakage models. Signature schemes in the bounded-leakage model were proposed by Alwen, Dodis, and Wichs [3] and by Katz and Vaikuntanathan [26], who focused mainly on leakage of *(only) the signing key* of the scheme. Specifically, a signature scheme is leakage-resilient in the bounded-leakage model if it is existentially unforgeable against an adaptive chosen-message attack [19] even when adversarially chosen functions of the signing key are leaked in an adaptive fashion. Signature schemes satisfying this notion of security were constructed both based on generic cryptographic primitives in the standard model [26] and based on the Fiat-Shamir transform [17] in the random-oracle model [26,3].

Although this notion of leakage resilience already captures some attacks, it does not fully capture general leakage attacks, which may depend on the *entire internal state* of the system. In particular, the problem is that both of the signature scheme constructions from [26,3] are *randomized* and hence the internal state includes, in addition to the secret-key, all of the random coins used by the signing algorithm[2]. The prior schemes may therefore be vulnerable to leakage-attacks that (also) depend on this randomness.

This was already noted by Katz and Vaikuntanathan [26], who put forward the stricter notion of a *fully leakage-resilient* signature schemes (in the bounded-leakage model). This notion requires a signature scheme to remain existentially unforgeable under an adaptive chosen-message attack even when the adversary obtains bounded leakage information on *all intermediate values* used by the signer throughout the lifetime of the system, including the secret-keys *and* internal random coins (the notion can be naturally extended to the continual-leakage model [13,10]). This stronger notion seems to better capture real attacks, relying

[1] If the time between refreshing is fixed, we can think of this as bounding the *rate* of leakage.

[2] No known deterministic or public-coin constructions of leakage-resilient signatures are known. Without leakage, the signing algorithm of any signature scheme can be made deterministic by using, as its random coins, the output of a pseudorandom function (PRF) applied to the message, where the seed of the PRF is made part of the secret key. However, in the setting of key leakage, this transformation may no longer be secure since the seed to the PRF can also leak.

on e.g. timing or power consumption patterns, since these likely *do* depend on the internal randomness.

Currently, however, the known constructions of fully leakage-resilient signature schemes are proven secure only in the random-oracle model [3,10,13,26]. Moreover, even in the random-oracle model, known schemes are either resilient to leakage of at most half the length of the signing key [3,13,26], or require refreshing of the signing key after every few invocation of the signing algorithm, even when no leakage occurs [10] (this is required even in the bounded-leakage model, where refreshing is not part of the typical functionality). In the standard model, only constructions of "one-time" signatures[3] from [26] are known to be fully leakage resilient.

In a concurrent and independent work, Malkin, Teranishi, Vahlis and Yung [29] propose a alternate signature scheme in the continual-leakage model. Although the two schemes appear very different at first, they can be seen as separate instantiations of a common strategy, which we will explain shortly.

1.1 Our Contributions

We construct the first fully leakage-resilient signature schemes without random oracles. We first present a scheme in the bounded-leakage model that is resilient to any leakage of $(1 - o(1))L$ bits, where L is the bit-length of the signing key. Our scheme is based on generic cryptographic primitives, and is inspired by the approach of Katz and Vaikuntanathan [26] (although their scheme is resilient to leakage from the signing key only). Moreover, we show that our construction can be instantiated based on specific number-theoretic assumptions to yield a rather *efficient* scheme.

We then extend our approach to the continual-leakage model by relying on any *continual leakage-resilient one-way relation*, a primitive recently introduced by Dodis, Haralambiev, Lopez-Alt and Wichs [13]. Our resulting signature scheme construction inherits the leakage resilience properties of the underlying one-way relation with respect to leakage allowed between successive key updates and during the refreshing algorithm. In particular, instantiating our scheme with existing constructions of the one-way relations from [13,10] yields schemes that are resilient to leakage of logarithmic length from the random bits used by the refreshing algorithm, and any leakage of length $(1 - o(1))L$ bits between any two key refreshes based on the Symmetric External Diffie-Hellman (SXDH) assumption, or $(1/2 - o(1))L$ bits between refreshes based on the Decisional-Linear assumption.

Finally, we note that our approach yields the first separation between the bounded-leakage model and the noisy-leakage model, which was formalized by Naor and Segev [30] and later refined by Dodis et al. [13, Definition 7.2]. Noisy leakage is a realistic generalization of bounded leakage, in which the leakage is not necessarily of bounded length, and it is only guaranteed that the secret key

[3] Such schemes can only be used to sign a single message (or, more generally, some a priori bound t on the number of messages). The amount of leakage-resilience is $\Theta(L/t)$ bits, and thus degrades with t.

still has some min-entropy even given the leakage. This settles an open problem posed by Naor and Segev.

1.2 Overview of Our Approach

In this section we present an overview of our approach for constructing fully leakage-resilient signature schemes. We focus here on our construction in the bounded-leakage model, as it already emphasizes the main ideas underlying our approach, and we refer the reader to the full version of the paper for an overview of our construction in the continual-leakage model. We begin by describing more clearly the notion of a fully leakage-resilient signature scheme in the bounded-leakage model. Then, we briefly describe the leakage-resilient signature scheme of Katz and Vaikuntanathan [26], which serves as our starting point, and explain the main challenges in constructing *fully* leakage-resilient signature schemes. The main part of this overview then focuses on our construction.

Modeling fully leakage-resilient signature schemes. A signature scheme is fully leakage-resilient in the bounded-leakage model if it is existentially unforgeable against an adversary that can obtain both signatures on any message of her choice, and bounded leakage information on all intermediate values used by the signer throughout the lifetime of the system.

This is formalized by considering an experiment that involves a signer and an adversary. First, the signer invokes the key-generation algorithm and obtains a verification key vk and a signing key sk. At this point, a value state is initialized to contain the random coins that were used by the key-generation algorithm. The adversary is given the verification key vk and can adaptively submit two types of queries: *signing queries*, and *leakage queries*. A signing query consists of a message m, and is answered by invoking the signing algorithm with the signing key and the message. Following each such query, the random coins that were used by the signing algorithm are added to the state. A leakage query consists of a leakage function f, and is answered by applying f to the value state. The leakage functions have to be efficiently computable, and the sum of their output lengths has to be upper bounded by a predetermined parameter λ. The adversary is successful if she outputs a pair (m^*, σ^*), where m^* is a message with which she did not issue a signing query, and σ^* is a valid signature on m^* with respect to vk. We refer the reader to Section 3 for a formal definition.

The Katz-Vaikuntanathan scheme. The Katz-Vaikuntanathan signature scheme [26] relies on a second-preimage resistant (SPR) function $\mathsf{F} : \{0,1\}^{\mu(n)} \to \{0,1\}^{\kappa(n)}$ (for some $\kappa(n) < \mu(n)$), a CPA-secure public-key encryption scheme, and a (unbounded simulation-sound) NIZK proof system[4]. The signing key is a

[4] A function F is second-preimage resistant if, given a random input x it is hard to find $x' \neq x$ such that $\mathsf{F}(x') = \mathsf{F}(x)$. See Definition 2.1 in Section 2. We note that when F is only assumed to be a one-way function, the scheme may not always be resilient to leakage, but it is nevertheless existentially unforgeable under an adaptive chosen-message attack. In this case the scheme can be viewed as a variant of the Bellare-Goldwasser signature scheme [4].

random $x \in \{0,1\}^{\mu(n)}$, and the verification key is a triplet $(y = \mathsf{F}(x), pk, \mathsf{crs})$, where pk is a public key for the encryption scheme, and crs is a common-reference string for the proof system. A signature on a message m consists of a ciphertext c which is an encryption of $m||x$ using pk, and a proof that the ciphertext c is indeed an encryption of $m||x'$, for some $x' \in \mathsf{F}^{-1}(y)$[5].

This scheme is leakage resilient in the bounded-leakage model. That is, it satisfies the weaker variant of the above notion of security, where the leakage is allowed to depend on the signing key only. The security of the scheme is based on three main properties:

1. A typical verification key has many possible secret keys. Specifically, the set $\mathsf{F}^{-1}(y)$ is of size roughly $2^{\mu(n)-\kappa(n)}$.
2. The "real" signatures of the scheme are *computationally indistinguishable* from "fake" signatures, which are *statistically independent* of the signing key. This follows from the semantic security of the encryption scheme and from the zero knowledge of the proof system. Specifically, a "fake" signature on a message m can be produced by encrypting $m||0^n$, and then using the NIZK simulator to generate the proof.
3. Given the decryption key corresponding to pk, any valid forgery produced by the adversary can be used to extract a preimage x' of y. This follows from the soundness of the proof system, which guarantees that the adversary's forgery is a "real" signature[6] and therefore the corresponding ciphertext can be decrypted to a valid preimage x'.

These three properties are used to prove the security of the scheme as follows. Assume there is an adversary that breaks the scheme. Then, given a random pre-image x of y, we can run this adversary and (by the third property) extract some valid preimage x' from the adversary's signing forgery with a reasonable probability. This would break second-preimage resistance of F as long as we can argue that $x' \neq x$. To do so, we use the second property to replace "real signatures" with "fake signatures" without affecting the probability of recovering some valid preimage x'. But now, the signing queries do not reveal any additional information about x, given y. So the only correlated information on x that the adversary sees is the value $y = F(x)$ of size $\kappa(n)$ and the leakage of size λ. Therefore, if $\lambda \leq \mu(n) - \kappa(n) - \omega(\log(n))$, then the adversary has (information theoretically) super-logarithmic uncertainty about the value of x and hence the probability of extracting $x' = x$ from her forgery is negligible.

The main challenges. The security proof of the Katz-Vaikuntanathan scheme relies on the argument that, given many signatures of chosen messages and λ bits of leakage from the signing key x, the value x is still hard to guess by

[5] Katz and Vaikuntanathan show that it is actually possible to encrypt only x (instead of $m||x$), and include m as a label in the statement that is proved using the NIZK proof system. However, for making this informal description more intuitive, we consider here an encryption of both m and x.

[6] In fact, a stronger notion called simulation-soundness is required, because the adversary gets to see several fake proofs before generating her signature.

the adversary. However, when the leakage may depend also on the randomness used by the signing algorithm, this is no longer true, and in fact the scheme is insecure in general. The main problem is that, in the above argument, we crucially used the ability to switch "real" signatures for "fake" signatures. This step, in turn, relied on the security of the encryption scheme and the zero-knowledge property of the proofs. However, we cannot rely on these properties if the adversary can also leak on the random coins of the encryption scheme and the proof system! Consider, for example, an instantiation of the scheme with a CPA-secure encryption scheme defined as $\mathsf{Enc}_{pk}(m||x) = (\mathsf{Enc}'_{pk}(s), \mathsf{PRG}(s) \oplus (m||x))$, where Enc' is secure encryption scheme, and PRG is a pseudorandom generator that is applied on a random seed s. Leaking the seed s, whose length may be arbitrarily shorter then λ, completely reveals the signing key x. A similar instantiation for the proof system can be shown to have a similar effect when the leakage may depend on the randomness used by the prover[7].

Our approach. A natural observation is that the above problems can be avoided if the "real" and "fake" signatures cannot be distinguished *even* given the random coins used to generate them. Remember that fake signatures are statistically independent of the secret key x, while real signatures allow us to extract some preimage using an appropriate trapdoor (decryption key).

The first idea toward achieving the above is to replace the (unbounded simulation-sound) NIZK proof system with a *statistical non-interactive witness-indistinguishable (SNIWI) argument system.* On one hand we relax the (unbounded simulation-sound) zero knowledge property to *witness indistinguishability*, and on the other hand we require that proofs generated using different witnesses are *statistically* indistinguishable from each other. In particular, this guarantees that *even* a correctly generated proof is statistically independent of the witness (in our case the signing key x) used to generate it.

The harder part lies in getting an encryption scheme where the ciphertexts are independent of the message (in our case, the signing key x) that they encrypt. In particular, this clearly contradicts the decryptability of a ciphertext. We could imagine using known lossy encryption schemes, where the encryption key pk can be generated in one of two indistinguishable modes: "*injective*" mode which allows for decryptability, and "*lossy*" mode where ciphertexts statistically hide the message. But remember that we need to satisfy the following two properties simultaneously: (1) the ability to answer the adversary's signing queries with fake signatures that reveal no information about x, (2) the ability to extract a witness x' from the adversary's forgery. By setting the pk to be in either injective or lossy mode, we can achieve either property, but not at the same time! The main tool used in resolving this conflict is to design a *partitioned-lossy encryption scheme*, where the encryption of some messages is lossy while that of others is injective.

[7] Note that even a leakage function with only one output bit can be easily used to distinguish an encryption of $m||x$ from an encryption of $m||0^n$, or to distinguish the prover of the proof system from the simulator of the proof system. Thus, technically speaking, it seems that at no point in time during the various experiments of the security proof it is possible to change the way signing queries are answered.

A selectively-unforgeable signature scheme. For the reader's intuition, we first show how to achieve a weaker notion of signature security that we refer to as *selective unforgeability under a chosen-message attack.* For this notion, we assume the adversary specifies the message m^* on which she plans to forge a signature in advance, before receiving the verification key. The signing queries and leakage are still adaptive.

To achieve this notion of security, we introduce the concept of an *all-lossy-but-one (ALBO) public-key encryption scheme.* This is a tag-based public-key encryption scheme, where the encryption procedure takes as input a tag t in addition to the message. The key-generation procedure takes as input a special tag t^* and produces a key pair (pk, sk) such that encrypting under the tag t^* allows for efficient decryption with sk, but encryption under any other tag $t \neq t^*$ statistically hides the encrypted message. We call t^* the *injective* tag, and any other tag a *lossy* tag[8]. The only computational requirement is that the public key hides the injective tag t^* that was used for its generation.

We now modify the Katz-Vaikuntanathan signature scheme by using an ALBO encryption scheme instead of a standard CPA-secure scheme. To sign m, we encrypt (only) the signing key x under the tag $t = m$. We use a SNIWI argument system instead of a simulation-sound NIZK to generate the proof. To argue security, we note that since the adversary's forgery message m^* is chosen ahead of time, we can generate the encryption key pk such that $t^* = m^*$ is the only injective tag, without affecting the adversary's ability to forge – this change is indistinguishable even given full view of the signing key x and randomness of signing. Now we are in a situation where all the signing queries for $m \neq m^*$ yield signatures which are statistically independent of the signing key x, while the forgery can be used to extract some preimage x'. Therefore, we can argue as before: the bounded leakage on the secret key x and randomness of signing is short enough that x must have entropy left given this leakage, and therefore the outcome $x' = x$ is unlikely.

The full scheme. So far we described our approach as leading to the rather weak notion of selective unforgeability under a chosen-message attack. Our actual scheme is fully leakage-resilient according to the stronger notion that was discussed in the beginning of this section (i.e., where the adversary is allowed to adaptively choose m^* after seing vk and responses to all signing and leakage queries).

We note that, in the random-oracle model, there is a simple generic transformation from selective security to full security by signing the output of the random oracle applied to the message. Alternatively, in the standard model, there is a simple transformation with exponential security loss by simply "guessing" the forgery: this can yield fully secure schemes under some exponential hardness assumptions by using complexity-leveraging. Lastly, there is a completely generic transformation due to [9] (abstracting a non-generic approach of [23]) by hashing the message with a chameleon hash function [27] and signing each prefix of

[8] We note that our notion is the opposite of the notion of an all-but-one lossy trapdoor function, where there is one lossy tag and all the other tags are injective.

the hash separately. Unfortunately, this results in long signatures. All of these generic techniques also work in the setting of full-leakage resilience. We present an alternative that does not suffer from the above disadvantages.

For our actual scheme, we follow the approach of Boneh and Boyen [5] for transforming selectively-secure identity-based encryption schemes into fully secure ones using an admissible hash function (see Section 2.3). This relies on a slightly more refined "partitioning strategy" than the "all-but-one" strategy used for the selectively-secure scheme. In particular, we introduce the notion of a *$\mathcal{R}$-lossy public-key encryption scheme*. This is a generalization of an ALBO encryption scheme where the set of possible tags is partitioned into injective tags and lossy tags according to a relation $\mathcal{R}$ (in particular, there may be more than one injective tag). The main idea of this approach is to ensure that, with polynomial probability, all of the adversary's signing queries will fall into the "lossy" partition, while the forgery falls into the "injective" partition.

Comparison to [29]. An alternate way to view our combination of a SNIWI paired with a partitioned lossy encryption is as a tag-based proof system that is partitioned to be extractable for some tags and statistically witness indistinguishable for others. Our main result shows how to build fully leakage-resilient signatures from such a proof system. The work of [29] can be seen as an alternate instantiation of this strategy which relies on Groth-Sahai NIZKs [22]. These NIZKs are either statistically witness indistinguishable or extractable depending on the choice of the CRS. In the reduction in [29], the CRS of the Groth-Sahai NIZK is derived from the tag in a clever way (using the Waters Hash [33]) so as to give an alternate useful partitioning of lossy/extractable tags.

1.3 Paper Organization

In Section 2 we introduce some preliminaries and notation. Section 3 contains a definition of security in the bounded-leakage model. In Section 4 we introduce $\mathcal{R}$-lossy public-key encryption schemes, a tool used in our constructions. Section 5 contains the construction and intuition for the security proof of our signature scheme in the bounded-leakage model. Finally, in Section 6 we discuss several concluding remarks and open problems. We refer the reader to the full version of the paper for a specific instantiation of our scheme based on the Linear assumption and the extension of our scheme to the continual-leakage model.

2 Preliminaries

In this section we present some basic tools that are used in our constructions.

2.1 Second-Preimage Resistance

A family of efficiently computable functions is a pair of polynomial-time algorithms $(\mathsf{KeyGen}, \mathsf{F})$, where KeyGen is a probabilistic algorithm that on input 1^n outputs a description $s \in \{0,1\}^*$ of a function $\mathsf{F}(s, \cdot) : \{0,1\}^{\mu(n)} \to \{0,1\}^{\kappa(n)}$.

Such a family is *second-preimage resistant* (SPR) if given a randomly chosen input $x \in \{0,1\}^{\mu(n)}$ and a description of a randomly chosen function $s \leftarrow \mathsf{KeyGen}(1^n)$, it is computationally infeasible to find an input $x' \in \{0,1\}^{\mu(n)}$ such that $x' \neq x$ and $\mathsf{F}(s,x) = \mathsf{F}(s,x')$. This is a weakening of the notion of a family of universal one-way hash functions introduced by Naor and Yung [31], in which the input x is allowed to be chosen in an adversarial manner (but still independently of the function description s).

Definition 2.1 (Second-preimage resistance). *A family $\mathcal{F} = (\mathsf{KeyGen}, \mathsf{F})$ of efficiently computable functions is* second-preimage resistant *if for any probabilistic polynomial-time algorithm $\mathcal{A}$ is holds that*

$$\Pr\left[\mathsf{F}_s(x') = \mathsf{F}_s(x) \wedge x' \neq x \,\middle|\, \begin{array}{c} s \leftarrow \mathsf{KeyGen}(1^n), x \leftarrow \{0,1\}^{\mu(n)} \\ x' \leftarrow \mathcal{A}(s,x) \end{array}\right] < \nu(n) \enspace ,$$

for some negligible function $\nu(n)$, where the probability is taken over the choice of $x \leftarrow \{0,1\}^{\mu(n)}$ and over the internal randomness of KeyGen *and* $\mathcal{A}$.

In addition, we say that $\mathcal{F} = (\mathsf{KeyGen}, \mathsf{F})$ is a family of *public-coin* second-preimage resistant functions, if it satisfies Definition 2.1 even when the algorithm $\mathcal{A}$ takes as input also the internal randomness that was used by $\mathsf{KeyGen}(1^n)$ for sampling the function. We refer the reader to [24] for more details on public-coin hash functions.

For any integer functions $\mu(n)$ and $\kappa(n)$ that are polynomially related, the existence of universal one-way hash functions (and therefore also of second-preimage resistant functions) with domain $\{0,1\}^{\mu(n)}$ and range $\{0,1\}^{\kappa(n)}$ is known to be equivalent to that of one-way functions [32]. As noted by Katz and Vaikuntanathan [26], standard constructions of universal one-way hash functions are public coin. In practice, such public-coin functions can be constructed rather easily from various number-theoretic assumptions. For example, if the discrete log problem is hard in some group $\mathbb{G}$ of prime order p, the family of functions $f_{g_1,\ldots,g_k} : \mathbb{Z}_p^k \rightarrow \mathbb{G}$ defined as $f_{g_1,\ldots,g_k}(x_1,\ldots,x_k) = \prod_{i=1}^{k} g_i^{x_i}$ is second-preimage resistant (and even collision resistant), where $g_1,\ldots,g_k \in \mathbb{G}$ are chosen uniformly and independently at random by the key-generation algorithm.

We note that for public-coin SPR functions, there is actually no need for an explicit key-generation algorithm. Without loss of generality one can define a single function $\mathsf{F}'_r(x) = (r, \mathsf{F}_s(x))$, where $s = \mathsf{KeyGen}(1^n; r)$, and this is also SPR with the same amount of "lossiness" as the family $\mathcal{F}$.

2.2 Statistical Non-interactive Witness-Indistinguishable Argument Systems

A non-interactive argument system for a language L with witness relation R_L is a triplet of algorithms $(\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$, where CRSGen is an algorithm generating a common reference string crs, and P and V are the prover and verifier algorithms, respectively. The prover takes as input a triplet (crs, x, w), where $(x,w) \in R_L$, and outputs a proof π. The verifier takes as input a triplet (crs, x, π) and either accepts or rejects. In this paper we consider a setting where all three algorithms run in probabilistic polynomial time. The two requirements of an argument system are completeness and soundness with respect to efficient cheating provers. Informally, for every $(x, w) \in R_L$ the prover generates proofs that are always accepted by the verifier, and for every $x \notin L$ any efficient cheating prover has only a negligible probability of convincing the verifier to accept. An argument system is called statistical witness indistinguishable if for any $x \in L$ and any two witnesses $w_0 \neq w_1$ such that $(x, w_0), (x, w_1) \in R_L$, the proofs generated by $\mathsf{P}(\mathsf{crs}, x, w_0)$ and $\mathsf{P}(\mathsf{crs}, x, w_1)$ are statistically indistinguishable given the common reference string.

Definition 2.2 (SNIWI argument system). *A* statistical non-interactive witness-indistinguishable argument system *for a language L with witness relation R_L is a triplet of probabilistic polynomial-time algorithms* $(\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$ *such that the following properties hold:*

1. **Perfect completeness:** *For every $(x, w) \in R_L$ it holds that*
$$\Pr\left[\mathsf{V}(\mathsf{crs}, x, \mathsf{P}(\mathsf{crs}, x, w)) = 1\right] = 1 \ ,$$
where $\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^n)$, and the probability is taken over the internal randomness of CRSGen*,* P*, and* V*.*
2. **Adaptive soundness:** *For every probabilistic polynomial-time prover* P^* *it holds that*
$$\Pr\left[\mathsf{V}(\mathsf{crs}, x, \pi) = 1 \wedge x \notin L \;\middle|\; \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{CRSGen}(1^n) \\ (x, \pi) \leftarrow \mathsf{P}^*(1^n, \mathsf{crs}) \end{array}\right] < \nu(n) \ ,$$
for some negligible function $\nu(n)$
3. **Statistical witness indistinguishability:** *There exists a probabilistic polynomial-time algorithm* CRSGen_{WI} *such that:*
   - *The distributions $\{\mathsf{CRSGen}(1^n)\}$ and $\{\mathsf{CRSGen}_{WI}(1^n)\}$ are* computationally indistinguishable*.*
   - *For any triplet (x, w_0, w_1) such that $(x, w_0) \in R_L$ and $(x, w_1) \in R_L$, the distributions $\{\mathsf{crs}, \mathsf{P}(\mathsf{crs}, x, w_0)\}$ and $\{\mathsf{crs}, \mathsf{P}(\mathsf{crs}, x, w_1)\}$ are* statistically indistinguishable*, when $\mathsf{crs} \leftarrow \mathsf{CRSGen}_{WI}(1^n)$.*

For our construction we are interested in SNIWI argument systems for NP. Such an argument system is implied by the construction of Groth, Ostrovsky and Sahai [21] that satisfies the stronger notion of a perfect non-interactive zero-knowledge argument system. Their construction can be based on the hardness of either the Decisional Subgroup problem [7] or the Decisional Linear problem [6]. As pointed out by Groth et al. we note that in their Linear-based construction the algorithm CRSGen admits oblivious sampling (specifically, the distribution of the common reference string is statistically-close to the uniform distribution), which is a technical property that is required for our construction in the bounded leakage model.

2.3 Admissible Hash Functions

The concept of an *admissible hash function* was first defined by Boneh and Boyen [5] to convert a natural selectively-secure identity-based encryption scheme into a fully-secure one. In this paper we use such hash functions in a similar manner to convert a selectively-secure signature scheme (where the adversary declares the message to be forged ahead of time, before receiving the verification key) into a fully secure one. The main idea of an admissible hash function is that it allows the reduction in the proof of security to secretly partition the message space into two subsets, which we will label as red ($\mathtt{R}$) and blue ($\mathtt{B}$), such that there is a noticeable probability that all of the messages in the adversary's signing queries will be in the blue set, but the forgery will be on a message in the red set. This is useful if the simulator can efficiently answer signing queries in the blue set, yet break some hard problem given a valid forgery on a message from the red set. Our exposition and definition of admissible hash function follow that of Cash, Hofheinz, Kiltz, and Peikert [11].

For $K \in \{0, 1, \perp\}^{\tau(n)}$, we define the function $F_K : \{0,1\}^{\tau(n)} \to \{\mathtt{R}, \mathtt{B}\}$ which "colors" the space $\{0,1\}^{\tau(n)}$ of tags in the following way:

$$F_K(y) := \begin{cases} \mathtt{R} & \text{if } \forall\, i \in \{1, \dots, \tau(n)\} \quad : \quad K_i = y_i \text{ or } K_i = \perp \\ \mathtt{B} & \text{otherwise} \end{cases}$$

For any $u = u(n) < \tau(n)$, we let $\mathcal{K}_{u,n}$ denote the uniform distribution over $\{0, 1, \perp\}^{\tau(n)}$ *conditioned on* exactly u positions having $\perp$ values. (Note, if K is chosen from $\mathcal{K}_{u,n}$, then the map $F_K(\cdot)$ colors exactly 2^u values red.) We would like to pick a distribution $\mathcal{K}_{u,n}$ for choosing K so that, there is a polynomial probability for any set of tags $y_0, \dots, y_q$ of y_0 being colored "red" and all other tags being colored "blue". Unfortunately, this cannot happen if we allow all tags. Instead, we will need to rely on a special hash function the maps messages x to tags y.

Let $\mathcal{H} = \{\mathcal{H}_n\}_{n\in\mathbb{N}}$ be a hash-function ensemble, where each $H \in \mathcal{H}_n$ is a polynomial-time computable function $H : \{0,1\}^* \to \{0,1\}^{\tau(n)}$. For each $H \in \mathcal{H}_n$, we define the function $F_{K,H} : \{0,1\}^* \to \{\mathtt{R}, \mathtt{B}\}$, which "colors" the space $\{0,1\}^*$ according to $F_{K,H}(x) = F_K(H(x))$.

Definition 2.3 (Admissible hash function [5,11]). *We say that $\mathcal{H}$ is an* admissible hash-function ensemble *if for every $H \in \mathcal{H}$ there exists a set* $\mathbf{bad}_H$ *of string-tuples such that the following two properties hold:*

- *For every probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function $\nu(n)$ satisfying*

$$\Pr[(x_0, \dots, x_q) \in \mathbf{bad}_H \mid H \leftarrow \mathcal{H}_n, (x_0, \dots, x_q) \leftarrow \mathcal{A}(1^n, H)] \le \nu(n) \enspace .$$

- *For every polynomial $q = q(n)$ there is a polynomial $p = p(n)$ and an efficiently computable $u = u(n)$ such that, for every $H \in \mathcal{H}_n$ and $(x_0, \dots, x_q) \notin$* $\mathbf{bad}_H$ *with $x_0 \notin \{x_1, \dots, x_q\}$, we have:*

$$\Pr_{K \leftarrow \mathcal{K}_{u,n}} [F_{K,H}(x_0) = \mathtt{R} \wedge F_{K,H}(x_1) = \dots = F_{K,H}(x_q) = \mathtt{B} \,] \ge \frac{1}{p(n)} \enspace .$$

We note that for the application to identity-based encryption [5,11] the bad sets $\mathbf{bad}_H$ are required to be efficiently recognizable, but this is not required for our application. In addition, we say that $\mathcal{H}$ is a *public-coin* admissible hash-function ensemble, if it satisfies Definition 2.3 even when the algorithm $\mathcal{A}$ takes as input also the internal randomness that was used by $\mathsf{KeyGen}(1^n)$ for sampling the function.

The work of Boneh and Boyen [5] shows how to construct admissible hash functions from collision-resistant hash functions. Moreover, if the underlying collision-resistant hash functions are public coin, then so are the resulting admissible hash functions. As already mentioned in Section 2.1, public-coin collision-resistant hash functions can be constructed rather easily from various number-theoretic assumptions.

3 Modeling Leakage-Resilient Signature Schemes

A signature scheme is a triplet $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ of probabilistic polynomial-time algorithms with syntax:

- $(vk, sk) \leftarrow \mathsf{KeyGen}(1^n)$ outputs a verification key and signing key.
- $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$ signs a message m using the singing key sk.
- $\mathsf{Verify}_{vk}(m, \sigma) \in \{0, 1\}$ outputs a bit deciding wether σ is a valid signature for m.

We require perfect correctness, which states that for any valid key pair (vk, sk) output by KeyGen and any message $m \in \{0, 1\}^*$ we have $\mathsf{Verify}_{vk}(m, \mathsf{Sign}_{sk}(m)) = 1$.

A signature scheme is fully leakage-resilient (FLR) in the bounded-leakage model if it is existentially unforgeable against an adversary that can obtain both signatures on any message of her choice, and bounded leakage information on all intermediate values used by the key-generation algorithm and the signer throughout the lifetime of the system. To model this, we define a variable state which includes all secret-state used by the system so far. Initially, we set state to be the random-coins of the KeyGen algorithm (note that we do not need to explicitly add sk to the state, since it can be easily computed from it by any leakage function). On each signing query made by the adversary, we append the random-coins of the signing algorithm to the state. The adversary can leak arbitrary information about state as long as the amount is overall-bounded.

Definition 3.1 (FLR security — bounded leakage). *A signature scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is* λ-fully-leakage-resilient in the bounded-leakage model *if for any probabilistic polynomial-time adversary* $\mathcal{A}$ *it holds that the probability of the event* $\mathsf{Success}^{\lambda\text{-}\mathsf{FLR}}_{\Pi,\mathcal{A}}(n)$ *is negligible in* n*, where this event is defined via the following experiment:*

1. *Sample* $r \leftarrow \{0, 1\}^*$*, compute* $(vk, sk) = \mathsf{KeyGen}(1^n; r)$*, and set* $\mathsf{state} = \{r\}$*.*
2. *The adversary* $\mathcal{A}$ *receives as input the pair* $(1^n, vk)$*, and can adaptively query a signing oracle and a leakage oracle that are defined as follows:*

  - **Signing queries.** *The signing oracle receives as input a message m_i, samples $r_i \leftarrow \{0,1\}^*$, and then computes $\sigma_i \leftarrow \mathsf{Sign}_{sk}(m_i; r_i)$. It updates* $\mathsf{state} := \mathsf{state} \cup \{r_i\}$ *and outputs σ_i.*
  - **Leakage queries.** *The leakage oracle receives as input a description of an efficiently computable function $f_j : \{0,1\}^* \rightarrow \{0,1\}^{\lambda_j}$, and outputs $f_j(\mathsf{state})$. We call λ_j the* output length *of the j-th leakage function.*

3. *The adversary $\mathcal{A}$ outputs a pair (m^*, σ^*).*
4. $\mathsf{Success}^{\lambda\text{-}\mathsf{FLR}}_{\Pi,\mathcal{A}}(n)$ *denotes the event in which:*
  - $\mathsf{Verify}_{vk}(m^*, \sigma^*) = 1$.
  - *m^* was not queried to the signing oracle.*
  - *The sum of output lengths of all leakage functions is at most $\lambda(n)$.*

For the definition of security within the continual-leakage model, we refer the reader to the full version of the paper.

4 $\mathcal{R}$-Lossy Public-Key Encryption

In this section we introduce the notion of an $\mathcal{R}$-lossy public-key encryption scheme. Informally, such a scheme is a tag-based public-key encryption scheme where the set of possible tags is partitioned into two subsets: *injective* tags, and *lossy* tags. When a message is encrypted under an injective tag, the resulting ciphertext can be correctly decrypted using the secret key. On the other hand, when encrypted under a lossy tag, the ciphertext statistically hides the message. The partitioning of the tags in defined by a binary relation $\mathcal{R} \subseteq \mathcal{K} \times \mathcal{T}$: the key-generation algorithm receives as input an *initialization value* $K \in \mathcal{K}$ and this partitions the set tags $\mathcal{T}$ so that $t \in \mathcal{T}$ is injective if and only if $(K, t) \in \mathcal{R}$. More, formally, we require that the relation $\mathcal{R} \subseteq \mathcal{K} \times \mathcal{T}$ consists of a sequence of efficiently (in n) recognizable sub-relations $\mathcal{R}_n \subseteq \mathcal{K}_n \times \mathcal{T}_n$.

The only computational requirement of an $\mathcal{R}$-lossy public-key encryption scheme is that the public key of the encryption scheme hides the initialization value K. That is, public keys produced by different initialization values are computationally indistinguishable.

Definition 4.1 ($\mathcal{R}$-lossy PKE). *Let $\mathcal{R} \subseteq \mathcal{K} \times \mathcal{T}$ be an efficiently computable binary relation. An $\mathcal{R}$-*lossy public-key encryption scheme *is a triplet of probabilistic polynomial-time algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *such that:*

1. **Key generation:** *For any initialization value $K \in \mathcal{K}_n$, the key-generation algorithm* KeyGen *on input $(1^n, K)$ outputs a secret key sk and a public key pk.*
2. **Decryption under injective tags:** *For any initialization value $K \in \mathcal{K}_n$ and tag $t \in \mathcal{T}_n$ such that $(K, t) \in \mathcal{R}_n$, and for any message $m \in \{0,1\}^{\ell(n)}$, it holds that*
$$\Pr\left[\mathsf{Dec}^t_{sk}(\mathsf{Enc}^t_{pk}(m)) = m\right] > 1 - \nu(n) \ ,$$
for some negligible function $\nu(n)$, where $(sk, pk) \leftarrow \mathsf{KeyGen}(1^n, K)$, and the probability is taken over the internal randomness of KeyGen, Enc *and* Dec.

3. **Lossiness under lossy tags:** *For any initialization value $K \in \mathcal{K}_n$ and tag $t \in \mathcal{T}_n$ such that $(K, t) \notin \mathcal{R}_n$, for every pair (sk, pk) of keys produced by $\mathsf{KeyGen}(1^n, K)$, and for every two messages $m_0, m_1 \in \{0,1\}^{\ell(n)}$, the distributions $\mathsf{Enc}^t_{pk}(m_0)$ and $\mathsf{Enc}^t_{pk}(m_1)$ are statistically indistinguishable.*
4. **Indistinguishability of initialization values:** *For every sequence of pairs $\{(K_n, K'_n)\}_{n\in\mathbb{N}}$ such that $K_n, K'_n \in \mathcal{K}_n$, the two ensembles $\{pk : (sk, pk) \leftarrow \mathsf{KeyGen}(1^n, K_n)\}_{n\in\mathbb{N}}$ and $\{pk : (sk, pk) \leftarrow \mathsf{KeyGen}(1^n, K'_n)\}_{n\in\mathbb{N}}$ are computationally indistinguishable.*

As with the other primitives that are used in our construction, we need to be able to obliviously sample public keys in a way that is computationally indistinguishable from those produced by $\mathsf{KeyGen}(1^n, \cdot)$. Specifically, we require that there exists a sequence of initialization values $\{K_n\}_{n\in\mathbb{N}}$ such that the ensemble $\{pk : (sk, pk) \leftarrow \mathsf{KeyGen}(1^n, K_n)\}_{n\in\mathbb{N}}$ is computationally indistinguishable from the uniform distribution over $\{0,1\}^*$. Note that by the indistinguishability of initialization values property defined above, this in fact holds for every sequence $\{K_n\}_{n\in\mathbb{N}}$.

For our constructions of fully leakage-resilient signature schemes we consider two relations: the equality relation $\mathcal{R}^{\mathsf{EQ}}$, and the more general "bit-matching" relation $\mathcal{R}^{\mathsf{BM}}$ that is defined below.

The relation $\mathcal{R}^{\mathsf{EQ}}$. The relation $\mathcal{R}^{\mathsf{EQ}}$ is the equality relation for binary tags of length $\tau(n)$ bits. That is, $\mathcal{K}_n = \mathcal{T}_n = \{0,1\}^{\tau(n)}$, and $(K, t) \in \mathcal{R}^{\mathsf{EQ}}_n$ if and only if $K = t$. An $\mathcal{R}^{\mathsf{EQ}}$-lossy encryption is just an *all-but-one-lossy* (ALBO) public-key encryption scheme, a primitive discussed in the introduction. In this case there is one injective tag, corresponding to the value of K used during initialization, and all the other tags are lossy.

The relation $\mathcal{R}^{\mathsf{BM}}$. The bit-matching relation $\mathcal{R}^{\mathsf{BM}}$ is a generalization of equality, which allows for more complex partitions. For $\mathcal{K}_n = \{0, 1, \perp\}^{\tau(n)}$, $\mathcal{T}_n = \{0,1\}^{\tau(n)}$ define $(K, t) \in \mathcal{R}^{\mathsf{BM}}_n \subseteq \mathcal{K}_n \times \mathcal{T}_n$ iff for every $i \in \{1, \ldots, \tau(n)\}$ it holds that $K_i = t_i$ or $K_i = \perp$. That is, given some fixed initialization value K, the set of injective tags t are exactly those whose bits match K in all positions i for which $K_i \neq \perp$. Notice that, if K does not contain any $\perp$ symbols, then there is a *single* injective tag $t = K$ and all other tags are lossy. Therefore $\mathcal{R}^{\mathsf{BM}}$-lossy encryption is a strict generalization of $\mathcal{R}^{\mathsf{EQ}}$-lossy encryption.

In our signature scheme construction, the $\mathcal{R}^{\mathsf{BM}}$-lossy encryption will be used in combination with an *admissible hash function* (discussed in Section 2.3). The admissible hash function gives us a way to map messages to encryption tags such that, with high probability over an appropriate distribution of K, all signing queries map to lossy tags while the forgery maps to an injective tag.

Constructions. In the full version, we propose two constructions of $\mathcal{R}^{\mathsf{BM}}$-lossy public-key encryption schemes[9]. Our first construction is rather generic and is based on any lossy public-key encryption scheme. In turn, this implies $\mathcal{R}^{\mathsf{BM}}$-lossy public-key encryption schemes can be based on a variety of number-theoretic

[9] We note that rather straightforward variants of these constructions yield $\mathcal{R}^{\mathsf{EQ}}$-lossy public-key encryption schemes.

assumptions. Our second construction is based on a specific number-theoretic assumption (the DDH assumption[10]) and is significantly more efficient than our generic construction.

5 A Signature Scheme in the Bounded-Leakage Model

In this section we present our construction of a fully leakage-resilient signature scheme in the bounded-leakage model (see Definition 3.1). We use the following primitives in a generic manner:

- Let $\mathcal{F} = (\mathsf{KeyGen}_{\mathsf{SPR}}, \mathsf{F})$ be a family of public-coin second-preimage resistant functions $\mathsf{F}_s(\cdot) : \{0,1\}^{\mu(n)} \to \{0,1\}^{\kappa(n)}$ for some $\kappa(n) < \mu(n)$ (see Section 2.1).
- Let $\mathcal{H}$ be a public-coin admissible hash function ensemble (see Section 2.3).
- Let $\mathcal{E} = (\mathsf{KeyGen}_{\mathcal{R}^{\mathsf{BM}}}, \mathsf{Enc}, \mathsf{Dec})$ be an $\mathcal{R}^{\mathsf{BM}}$-lossy public-key encryption scheme (see Section 4).
- Let $\Pi = (\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$ be a SNIWI argument system for the language

  $$L = \{(s, y, pk, t, C) : \exists x, \omega \text{ st } C = \mathsf{Enc}^t_{pk}(x; \omega) \text{ and } \mathsf{F}_s(x) = y\}$$

  (see Section 2.2).

We assume that the distribution of public keys and common-reference strings produced by the algorithms $\mathsf{KeyGen}_{\mathcal{R}^{\mathsf{BM}}}$ and CRSGen, respectively, are computationally indistinguishable from the uniform distribution over $\{0,1\}^*$[11]. Define the signature scheme $\mathcal{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$:

- **Key generation:** On input 1^n, the algorithm KeyGen samples a uniformly distributed $x \leftarrow \{0,1\}^{\mu(n)}$, a function description $s \leftarrow \mathsf{KeyGen}_{\mathsf{SPR}}(1^n)$ from the SPR family, and computes $y = \mathsf{F}_s(x)$. Then, it samples a description of an admissible hash function $H \leftarrow \mathcal{H}_n$, and samples $pk \leftarrow \{0,1\}^*$ and $\mathsf{crs} \leftarrow \{0,1\}^*$ to be used as a public key for the $\mathcal{R}^{\mathsf{BM}}$-lossy encryption scheme and a common-reference string for the SNIWI argument system, respectively. It outputs the signing key $sk = x$ and the verification key $vk = (s, y, H, pk, \mathsf{crs})$.
- **Signing:** On input message m, the algorithm Sign computes an encryption $C = \mathsf{Enc}^{H(m)}_{pk}(x; \omega)$ of x under the tag $H(m)$ using fresh randomness ω. Then, it invokes the prover of the argument system to obtain a proof $\pi \leftarrow \mathsf{P}(\mathsf{crs}, (s, y, pk, H(m), C), (x, \omega))$, and outputs the signature (C, π).
- **Verifying:** On input message m and signature $\sigma = (C, \pi)$, the algorithm Verify invokes the verifier of the argument system and outputs 1 if and only if $\mathsf{V}(\mathsf{crs}, (s, y, pk, H(m), C), \pi) = 1$.

[10] Our construction easily generalizes to rely on the d-Linear assumption for any $d \geq 1$.

[11] More generally, we just require "oblivious"sampling, but we will assume uniform distribution for simplicity. See Appendix 2.

Theorem 5.1. *Assuming the existence of the schemes $\mathcal{F}$, $\mathcal{H}$, $\mathcal{E}$ and Π with properties described above, the scheme $\mathcal{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is λ-fully-leakage-resilient in the bounded-leakage model for any $\lambda = \mu(n) - \kappa(n) - \omega(\log n)$. The relative leakage is given by $\lambda/|sk| \approx (1 - \kappa(n)/\mu(n)) = (1 - o(1))$ for an appropriate choice of $\kappa(n) = o(\mu(n))$.*

Due to space limitations the proof of Theorem 5.1 is left to the full version of the paper, and we only give a short proof outline here.

Proof outline. Suppose there is an adversary who breaks the security of the scheme. We can then use the adversary to break the security of the SPR function as follows. Choose a random crs for the SNIWI argument honestly, and a (pk, sk) pair for $\mathcal{R}^{\mathsf{BM}}$-lossy public-key encryption using an initialization value K sampled from an appropriate distribution (dictated by the admissible hash function, depending on the number of signing queries the adversary makes). Given a random challenge x from the SPR challenger, we embed $y = \mathsf{F}(x), \mathsf{crs}, pk$ into the verification key and then run the forging adversary, using x to answer all its signing/leakage queries. If the adversary's forgery is on a message m^* that corresponds to a injective tag of the encryption scheme, then we use sk to decrypt a (hopefully second preimage) x' from the adversary's forged signature. We argue that, with polynomial probability, we do recover a *second preimage* $x' \neq x$, using the following steps:

- Using the partitioning argument of Boneh-Boyen [5], there is a noticeable probability that the all of the adversary's signing queries correspond to "lossy" tags while the forgery corresponds to an "injective" tag. Here we rely on the property that the initialization value K is hidden by the public-key. We call an execution where the above occurs a "good execution."
- In a good execution, the adversary's forgery can be decrypted to a valid preimage $x' \in \mathsf{F}^{-1}(y)$, by the soundness of the SNIWI argument.
- Information theoretically, the probability of $x' = x$ in a good execution is negligible, since the adversary just doesn't have enough information about x. That is, the signature-query responses are independent of x, and the leakage-query responses and the verification key y are too short. This is formalized with an entropy argument.

6 Concluding Remarks and Open Problems

Deterministic leakage-resilient signatures. An alternative approach for constructing fully leakage-resilient signature schemes is constructing a signature scheme that is resilient to leakage from the signing key, and has a deterministic signing algorithm (this is indeed the idea underlying the fully leakage-resilient *one-time* signature schemes of Katz and Vaikuntanathan [26]). In general, the signing algorithm of any signature scheme can be made deterministic by using as its random coins the output of a pseudorandom function applied to the message. This requires, however, that the signing key will include also the key of the pseudorandom function, and therefore it is not clear that such a transformation can preserve leakage resilience.

Bounded leakage vs. noisy leakage. In some scenarios it is not always possible to assume that the total amount of leakage is upper bounded by λ bits. This motivated the approach of Naor and Segev [30] (later refined by Dodis et al. [13, Definition 7.2]) who considered the more general notion of *noisy leakage*, in which the leakage is not necessarily of bounded length, but is guaranteed to reduce the average min-entropy of the secret key by at most λ. Although our schemes are secure with respect to bounded leakage, they are in fact insecure with respect to noisy leakage. This seems to be the first separation between bounded leakage and noisy leakage, and this settles an open problem posed by Naor and Segev.

Specifically, in our schemes the public key for the $\mathcal{R}^{\mathsf{BM}}$-lossy encryption scheme is sampled obliviously as a uniformly random string $pk \in \{0,1\}^*$. For our specific constructions based on the DDH or Linear assumptions (see full version), this can be easily seen to imply that with an overwhelming probability all possible tags for the $\mathcal{R}^{\mathsf{BM}}$-lossy scheme are lossy. An analysis almost identical to that presented in the security proofs of our schemes then shows that a leakage function that simply outputs a signature on any message m^* is a valid leakage function with respect to noisy leakage (yet clearly invalid with respect to bounded leakage).

Modeling hard-to-invert leakage for signature schemes. In the setting of public-key encryption a more general model of leakage was formalized by only assuming that the decryption key cannot be efficiently recovered given the leakage (see [15,12,20,8] and the references therein). For signature schemes, however, it is not clear how to meaningfully formalize such an attack model. It would be interesting to formalize hard-to-invert leakage for signature schemes (especially when any intermediate value may leak, and not only the signing key), and to construct schemes that are leakage resilient in such a model.

Acknowledgements

We thank Moni Naor, Brent Waters, and the Eurocrypt '11 referees for many useful comments on this work.

References

1. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009)
2. Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-key encryption in the bounded-retrieval model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 113–134. Springer, Heidelberg (2010)
3. Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 36–54. Springer, Heidelberg (2009)

4. Bellare, M., Goldwasser, S.: New paradigms for digital signatures and message authentication based on non-interactive zero knowledge proofs. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 194–211. Springer, Heidelberg (1990)
5. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
6. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
7. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
8. Brakerski, Z., Goldwasser, S.: Circular and leakage resilient public-key encryption under subgroup indistinguishability (or: Quadratic Residuosity strikes back). In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 1–20. Springer, Heidelberg (2010)
9. Brakerski, Z., Tauman Kalai, Y.: A framework for efficient signatures, ring signatures and identity based encryption in the standard model. Cryptology ePrint Archive, Report 2010/086 (2010)
10. Brakerski, Z., Tauman Kalai, Y., Katz, J., Vaikuntanathan, V.: Cryptography resilient to continual memory leakage. In: Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science, pp. 501–510 (2010)
11. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
12. Dodis, Y., Goldwasser, S., Tauman Kalai, Y., Peikert, C., Vaikuntanathan, V.: Public-key encryption schemes with auxiliary inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 361–381. Springer, Heidelberg (2010)
13. Dodis, Y., Haralambiev, K., Lopez-Alt, A., Wichs, D.: Cryptography against continuous memory attacks. In: Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science, pp. 511–520 (2010)
14. Dodis, Y., Haralambiev, K., Lopez-Alt, A., Wichs, D.: Efficient public-key cryptography in the presence of key leakage. Cryptology ePrint Archive, Report 2010/154 (2010)
15. Dodis, Y., Tauman Kalai, Y., Lovett, S.: On cryptography with auxiliary input. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp. 621–630 (2009)
16. Faust, S., Rabin, T., Reyzin, L., Tromer, E., Vaikuntanathan, V.: Protecting circuits from leakage: the computationally-bounded and noisy cases. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 135–156. Springer, Heidelberg (2010)
17. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
18. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
19. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing 17(2), 281–308 (1988)
20. Goldwasser, S., Tauman Kalai, Y., Peikert, C., Vaikuntanathan, V.: Robustness of the learning with errors assumption. In: Proceedings of the 1st Symposium on Innovations in Computer Science, pp. 230–240 (2010)

21. Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for NP. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 339–358. Springer, Heidelberg (2006)
22. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
23. Hohenberger, S., Waters, B.: Short and stateless signatures from the RSA assumption. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 654–670. Springer, Heidelberg (2009)
24. Hsiao, C.-Y., Reyzin, L.: Finding collisions on a public road, or do secure hash functions need secret coins? In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 92–105. Springer, Heidelberg (2004)
25. Ishai, Y., Sahai, A., Wagner, D.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (2003)
26. Katz, J., Vaikuntanathan, V.: Signature schemes with bounded leakage resilience. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 703–720. Springer, Heidelberg (2009)
27. Krawczyk, H., Rabin, T.: Chameleon signatures. In: Proceedings of the Network and Distributed System Security Symposium (NDSS) (2000)
28. Lyubashevsky, V., Palacio, A., Segev, G.: Public-key cryptographic primitives provably as secure as subset sum. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 382–400. Springer, Heidelberg (2010)
29. Malkin, T., Teranishi, I., Vahlis, Y., Yung, M.: Signatures resilient to continual leakage on memory and computation (2010) (manuscript)
30. Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009)
31. Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing, pp. 33–43 (1989)
32. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, pp. 387–394 (1990)
33. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)