Leakage-Resilient Signatures with Graceful Degradation

Jesper Buus Nielsen[1], Daniele Venturi[2]*, and Angela Zottarel[1]

[1] Aarhus University
[2] Sapienza University of Rome

Abstract. We investigate new models and constructions which allow leakage-resilient signatures secure against existential forgeries, where the signature is much shorter than the leakage bound. Current models of leakage-resilient signatures against existential forgeries demand that the adversary cannot produce a new valid message/signature pair (m, σ) even after receiving some λ bits of leakage on the signing key. If $|\sigma| \leq \lambda$, then the adversary can just choose to leak a valid signature σ, and hence signatures must be larger than the allowed leakage, which is impractical as the goal often is to have large signing keys to allow a lot of leakage.
We propose a new notion of leakage-resilient signatures against existential forgeries where we demand that the adversary cannot produce $n = \lfloor \lambda/|\sigma| \rfloor + 1$ distinct valid message/signature pairs $(m_1, \sigma_1), \ldots, (m_n, \sigma_n)$ after receiving λ bits of leakage. If $\lambda = 0$, this is the usual notion of existential unforgeability. If $1 < \lambda < |\sigma|$, this is essentially the usual notion of existential unforgeability in the presence of leakage. In addition, for $\lambda \geq |\sigma|$ our new notion still guarantees the best possible, namely that the adversary cannot produce more forgeries than he could have leaked, hence graceful degradation.
Besides the game-based notion hinted above, we also consider a variant which is more simulation-based, in that it asks that from the leakage a simulator can "extract" a set of $n - 1$ messages (to be thought of as the messages corresponding to the leaked signatures), and no adversary can produce forgeries not in this small set. The game-based notion is easier to prove for a concrete instantiation of a signature scheme. The simulation-based notion is easier to use, when leakage-resilient signatures are used as components in larger protocols.
We prove that the two notion are equivalent and present a generic construction of signature schemes meeting our new notion and a concrete instantiation under fairly standard assumptions. We further give an application, to leakage-resilient identification.

1 Introduction

The problem of message authentication is one of the most basic in cryptography. Alice wants to transmit a message m to Bob via an insecure channel, with the

* Part of the work done while at Aarhus University supported by the DFF Starting Grant 10-081612.

guarantee that the message will reach the destination without any modification by a third party on the communication channel. In a world where public-key cryptography exists the latter can be achieved via a *digital signature*: Before sending m, Alice computes a signature σ (via her signing key sk) of the message, and transmits (m, σ) over the channel. The idea is that Bob can later verify the signature using Alice's verification key vk, and thus establish whether the received message is consistent with the original.

Traditionally, security of signatures schemes (and other primitives) is modeled in a black-box fashion where an adversary can only access the algorithms underlying the scheme as a black-box. For instance, in the case of a signature scheme, we require that no computationally bounded adversary is able to forge a signature of a message (with respect to some verification key vk) even given black-box access to an oracle returning signatures of arbitrarily chosen messages (computed via the signing key corresponding to vk).[3] However, as pointed out by recent research, the model above might be too restrictive, in that in practice there are several ways by which an adversary can learn partial information (a.k.a. leakage) on the secrets used within a cryptographic primitive, and thus easily step out of the security model. This includes so-called side-channel attacks, based on timings [27], power analysis [28] and electromagnetic radiation [35].

A large body of work has extended standard cryptographic definitions such that they can capture different flavours of security against leakage, both in the game-based setting (e.g. [13, 33, 1, 30, 24, 14, 8, 10, 11, 6]) and in the simulation-based setting [17, 21, 4, 31]. In the case of a signature scheme, a simple extension of the black-box setting requires that no computationally bounded adversary is able to forge a signature of a message (with respect to some verification key vk) even given black-box access to an oracle returning signatures on arbitrarily chosen messages (computed via the signing key corresponding to vk) and to a leakage oracle returning bounded (but otherwise arbitrary) information on the signing key sk. This is often referred to as the *bounded* leakage model, and on this we focus our work. See Section 1.2 for a discussion on other models.

The modeling above requires two *necessary* limitations. The first limitation is that the total amount of leakage must be smaller than the length of the signing key, as otherwise the entire key can be learned by the adversary, leaving no hope for security. The second limitation is that a signature has to be longer than the leakage bound, as otherwise a leakage query can just leak a forgery which is a valid attack against the security definition. A similar issue was already observed by Alwen, Dodis and Wichs [1], in their work on leakage-resilient public-key cryptography in the so-called bounded retrieval model. In this setting, the secret key is made intentionally large (say, 100 gigabytes) such that it may be infeasible/impractical for the attacker to download "too much" data (say, more than 1 gigabytes). Still, the length of the public key and the computational overhead are essentially independent from the size of the secret key. For the very same reason pointed out above, no signature scheme can be proven existentially unforgeable

[3] The restriction is of course that the forgery should not correspond to one of the messages asked to the oracle.

in the bounded retrieval model, as the leakage could simply consist of a forgery. To tackle this issue the authors in [1] considered a weaker notion, which they name *entropic unforgeability*, where, after the leakage phase, the adversary is required to forge the signature of a message sampled from a (potentially adversarially chosen) distribution of high enough min-entropy (given the entire view of the adversary). [1] then shows that entropic unforgeability can be achieved in the random oracle model [3], by applying the Fiat-Shamir transform [16] to a certain class of interactive protocols.

In this work we propose more granular ways to model (bounded) leakage resilience for signature schemes where the length of the signature is smaller than the length of the secret key. In a nutshell, our simplest notion says that an adversary leaking λ bits will always be able to produce $\lfloor \lambda/|\sigma| \rfloor$ forgeries, but not more than that. At first glance it may seem that our notion gives a weaker guarantee. However, the number of forgeries the adversary is required to produce strictly depends on the actual leakage, so if an adversary asks for no leakage (i.e. we are in the black-box model), our notion is equivalent to standard existential unforgeability, as now $\lfloor \lambda/|\sigma| \rfloor = 0$. On the other hand, when leakage does happen, our definition offers a graceful degradation of security and, as we argue in more details below, still allows for interesting, non-trivial, applications.

1.1 Our Contribution

We investigate new models and constructions which allow leakage-resilient signatures secure against existential forgeries, where the signature is much shorter than the leakage bound. Our main contributions are discussed in detail below.

One-more unforgeability. As a first contribution, we state a variant of leakage resilience for signature schemes where the length of the secret key is much larger than the length of a signature.[4] We name our notion *one-more unforgeability*, since it has a similar flavour to the unforgeability notion for blind signatures [34]. The attacker (given the verification key vk) can access a signing oracle and a leakage oracle; at the end he has to output n forgeries $(m_1, \sigma_1), \ldots, (m_n, \sigma_n)$ and wins the game if and only if all the forgeries are valid, the messages are pairwise distinct, and n is strictly larger than the number of forgeries one could have leaked via leakage queries. See Section 3 for a precise definition.

We also formulate a seemingly stronger variant, which we name *constrained*-one-more unforgeability. Here we introduce a simulator S which first looks at the state of the adversary A after the leakage phase ended and then defines a set of messages $\mathcal{Q}^*$ of size strictly smaller than n, as defined above. A signature scheme is secure in this setting if, for all A, there exists such a simulator for which A is not able to forge a message which is not contained in $\mathcal{Q}^*$ (and was not already asked to the signing oracle). This captures the intuition that the forgeries are

[4] Note that this in general encompasses schemes with short signatures, and not necessarily signature schemes in the bounded retrieval model.

already fixed after the leakage is ended, and the adversary is "constrained" in the sense that those are the only messages for which he can forge.[5]

We show that one-more unforgeability and constrained-one-more unforgeability are equivalent. The tricky direction is to show that the former implies the latter. The intuition is using an adversary breaking constrained-one-more unforgeability and rewinding him to obtain a sufficiently large set of forgeries: if at each rewinding we use a strictly larger set $\mathcal{Q}^*$ (including all previous forgeries output by the adversary), after n steps we end-up with n forgeries which allow to break one-more unforgeability. The actual analysis is more involved, as we need to take care of the fact that we are rewinding the adversary at the point where he is already committed to the leakage.

A construction. As a second contribution we present a scheme achieving one-more-unforgeability, based on a perfectly hiding (homomorphic) commitment scheme and a non-interactive zero knowledge argument of knowledge system.

The secret key consists of the coefficients δ_i of a d-degree polynomial $\delta(\cdot)$ over a finite field, together with the openings r_i for the commitments com_i to δ_i. The verification key consists of the set of all com_i together with a common reference string for the argument system. To sign a message m, we compute $\delta(m)$ and we produce a zero-knowledge argument of knowledge that the evaluation of the polynomial was performed correctly using the coefficients whose commitments are in the verification key. The signature consists of such an argument.

We prove that the scheme is one-more unforgeable whenever the commitment is perfectly hiding (and computationally binding), as long as the leakage is smaller than $(1/2 - o(1)) \cdot |sk|$. We also show a particular instantiation, using standard building blocks such as Pedersen commitments [32] and Groth-Sahai proofs [19, 20]. Security follows from the DLIN assumption [5]. We remark that for our concrete instantiation it is indeed the case that the length of a signature is essentially independent of the length of the secret key.

Application to identification protocols. Besides being a notion of theoretical interest, we also show that one-more unforgeability can be applied in the context of identification protocols. We focus on the public-key setting, where a prover P wants to be identified from a verifier V holding P's public key.

Following [1], we define security in the presence of leakage by considering an adversary having black-box access to the prover and to a leakage oracle (depend-

[5] We note that constrained-one-more unforgeability is strictly stronger than entropic unforgeability [1]. If a scheme is constrained-one-more unforgeable, then after the leakage is done, a poly-sized set of messages $\mathcal{Q}^*$ is defined and the adversary cannot forge for a message outside $\mathcal{Q}^*$, whereas a high entropy message will hit inside $\mathcal{Q}^*$ with negligible probability. On the other hand consider a signature scheme where a signature is given as $\sigma = \Pi^{-1}(m)$ for a one-way trapdoor permutation Π hard to invert on high-entropy m. Such a scheme is entropic secure in the presence of $\lambda = 0$ bits of leakage, by definition, but is clearly not constrained-one-more unforgeable in the presence of $\lambda = 0$ bits of leakage, as the adversary can always sample one more random message/signature pair as $m = \Pi(\sigma)$ for random σ.

ing on the prover's secret key) in a first phase. In a second phase the adversary is given one chance to convince the verifier. The above notion is reminiscent of so-called *active* security [23, 25].

We show that the classical protocol for public-key identification, where the verifier challenges the prover with a random message and the prover has to respond with a signature on that message, achieves the above notion of active security[6] with leakage, provided that the underlying signature scheme is constrained-one-more unforgeable.

1.2 Other Related Work

In this work (similarly to [1, 10, 15]) we focus on bounded leakage resilience, i.e., we assume that there is an a-priori upper bound on the length of the maximum tolerated leakage. Furthermore, we consider a setting where the leakage can only depend on the signing key and not on the full state of the signer (including, e.g., the signer's random coins). A strictly stronger notion of *fully* leakage-resilient signatures (where the leakage is bounded but can depend on the entire state of the signer) was considered in [24, 6].

In the continual leakage setting [7, 9, 29, 6], there is no *a priori* bound on the length of the leakage. This requires an efficient procedure to update the secret key (while leaving the public key unchanged), and to assume that the leakage is bounded only between two updates (and during the update process itself).

An independent line of research (see, e.g. [22, 26]) aims at constructing signature schemes (in the black-box model) which are as short as possible. Even though this is not our purpose, we believe that our notions could have interesting implications in this setting, when studying leakage resilience of such schemes.

2 Preliminaries

2.1 Notation

For $a, b \in \mathbb{R}$, we let $[a, b] = \{x \in \mathbb{R} \ ; \ a \leq x \leq b\}$; for $a \in \mathbb{N}$ we let $[a] = \{1, 2, \ldots, a\}$. If x is a string, we denote its length by $|x|$; if $\mathcal{X}$ is a set, $|\mathcal{X}|$ represents the number of elements in $\mathcal{X}$. When x is chosen randomly in $\mathcal{X}$, we write $x \leftarrow \mathcal{X}$. When A is an algorithm, we write $y \leftarrow \mathsf{A}(x)$ to denote a run of A on input x and output y; if A is randomized, then y is a random variable and $\mathsf{A}(x; r)$ denotes a run of A on input x and randomness r. An algorithm A is *probabilistic polynomial-time* (PPT) if A is randomized and for any input $x, r \in \{0, 1\}^*$ the computation of $\mathsf{A}(x; r)$ terminates in at most $\mathit{poly}(|x|)$ steps.

Throughout the paper we let κ denote the security parameter. We say that a function $\nu : \mathbb{N} \to \mathbb{R}$ is negligible in the security parameter κ if $\nu(\kappa) = \kappa^{-\omega(1)}$. For two ensembles $\mathcal{X} = \{X_\kappa\}_{\kappa \in \mathbb{N}}$ and $\mathcal{Y} = \{Y_\kappa\}_{\kappa \in \mathbb{N}}$, we write $\mathcal{X} \equiv \mathcal{Y}$ if they

[6] In fact, as argued in [2], without leakage the signature based protocol is even secure against man-in-the-middle attacks. It is not hard to see, however, that our result does not extend to man-in-the-middle security.

are identically distributed and $\mathcal{X} \approx_s \mathcal{Y}$ to denote that the statistical distance between the two distributions is negligible in the security parameter. We say that $\mathcal{X}$ and $\mathcal{Y}$ are computationally indistinguishable if for all PPT distinguishers D it holds that $|\mathbb{P}[\mathsf{D}(1^\kappa, X) = 1] - \mathbb{P}[\mathsf{D}(1^\kappa, Y) = 1]|$ is negligible in κ.

The min-entropy of a random variable X over a set $\mathcal{X}$ is defined as $\mathbb{H}_\infty(X) := -\log \max_x \mathbb{P}[X = x]$ and represents the best chance of guessing X by an unbounded adversary. Average min-entropy captures how hard it is to guess X on average, given some side information Z (possibly related to X):

$$\widetilde{\mathbb{H}}_\infty(X|Z) = -\log \mathbb{E}_z \left[\max_x \mathbb{P}[X = x|Z = z]\right].$$

The min-entropy of a distribution conditioned to some side information cannot decrease more than the bit-length of the side information itself:

Lemma 1 ([12]). *For all random variables $X \in \mathcal{X}$ and $\Lambda \in \{0,1\}^\lambda$ we have that $\widetilde{\mathbb{H}}_\infty(X|\Lambda) \geq \mathbb{H}_\infty(X) - \lambda$.*

We let $\mathcal{O}^\ell(s)$ be an oracle parametrized by a value s, which takes as input efficiently computable functions $f : \{0,1\}^* \to \{0,1\}^*$ and outputs $f(s)$, returning a total of at most ℓ bits.

2.2 Commitment Schemes

A (non-interactive) commitment scheme $\mathcal{COM}$ is a tuple of algorithms (Setup, Commit), defined as follows: (1) Algorithm Setup takes as input the security parameter and outputs a public key pk; (2) Algorithm Commit takes as input a message $m \in \mathcal{M}$, randomness $r \in \mathcal{R}$, the public key pk and outputs a value $com \in \mathcal{C}$. To open a commitment com we output (m, r); an opening is valid if and only if $com = \mathsf{Commit}(m; r)$.

A commitment scheme has two properties, known as binding and hiding. In Section 4 we need a scheme with the following flavour.

Computationally Binding : For any PPT adversary A, the following is negligible:

$$\mathbb{P}\left[\mathsf{Commit}(m_0; r_0) = \mathsf{Commit}(m_1; r_1) : \begin{array}{c} pk \leftarrow \mathsf{Setup}(1^\kappa); \\ ((m_0, r_0), (m_1, r_1)) \leftarrow \mathsf{A}(pk) \end{array}\right].$$

Statistically Hiding : For all messages $m_0, m_1 \in \mathcal{M}$, we have that

$$\{pk, \mathsf{Commit}(pk, m_0)\}_{\kappa \in \mathbb{N}} \approx_s \{pk, \mathsf{Commit}(pk, m_1)\}_{\kappa \in \mathbb{N}},$$

where the two ensembles are considered as random variables over the choice of the randomness to generate $pk \leftarrow \mathsf{Setup}(1^\kappa)$ and to compute the commitment. If the two ensembles are identically distributed, we say that the commitment is *perfectly* hiding.

Whenever $\mathcal{M}$ and $\mathcal{R}$ are a finite field $\mathbb{F}$, we say that $\mathcal{COM}$ is *linearly homomorphic* in the following sense: Given commitments *com* and *com*$'$ and a field element $c \in \mathbb{F}$, one can compute commitments com^* and com'' such that being able to open *com* and com' to m and m' (respectively) allows to open com^* to $m + m'$ and com'' to $c \cdot m$. We will write the mapping $(com, com') \mapsto com^*$ as $com \cdot com'$ and the mapping $(c, com) \mapsto com''$ as com^c. Similarly, for the opening information we will write the mappings as $com^* = \mathsf{Commit}(pk, m + m'; r + r')$ and $com'' = \mathsf{Commit}(pk, c \cdot m; c \cdot r)$. The above can be generalized to abstract operations over $\mathcal{M}$, $\mathcal{R}$ and $\mathcal{C}$, but for simplicity, and to be consistent with the concrete instantiation given in Section 4.2, we stick to this formulation here.

2.3 Non-Interactive Zero-Knowledge Arguments of Knowledge

For a relation $\mathfrak{R} \subseteq \{0,1\}^* \times \{0,1\}^*$, the language associated with $\mathfrak{R}$ is $\mathfrak{L}_{\mathfrak{R}} = \{x : \exists w \text{ s.t. } (x, w) \in \mathfrak{R}\}$. A non-interactive argument system $\mathcal{NIZK}$ for a relation $\mathfrak{R}$ is a tuple of algorithms $(\mathsf{Init}, \mathsf{Prove}, \mathsf{Ver})$, defined as follows: (1) Algorithm Init takes as input the security parameter and outputs a common reference string $\mathsf{crs} \leftarrow \mathsf{Init}(1^\kappa)$; (2) Algorithm Prove takes as input a pair (x, w) such that $(x, w) \in \mathfrak{R}$ and outputs an argument π; (3) Algorithm Ver takes as input a pair (x, π) and outputs a judgement in $\{0, 1\}$.

We require the following properties for $\mathcal{NIZK}$ [36, 10].

Completeness: For every $(x, w) \in \mathfrak{R}$ we have that

$$\Pr[\mathsf{Ver}(\mathsf{crs}, (x, \pi)) = 1 : \mathsf{crs} \leftarrow \mathsf{Init}(1^\kappa); \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, (x, w))] \geq 1 - negl(\kappa).$$

Multi-theorem zero-knowledge: There exists a PPT simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ such that, for all PPT adversaries A, the ensembles $\{\mathsf{Real}(\kappa)\}_{\kappa \in \mathbb{N}}$ and $\{\mathsf{Simu}(\kappa)\}_{\kappa \in \mathbb{N}}$ are computationally close, where

$$\mathsf{Real}(\kappa) := \left\{\mathsf{crs} \leftarrow \mathsf{Init}(1^\kappa); out \leftarrow \mathsf{A}^{\mathsf{Prove}(\mathsf{crs}, \cdot)}(\mathsf{crs})\right\}$$
$$\mathsf{Simu}(\kappa) := \left\{(\mathsf{crs}, tk) \leftarrow \mathsf{Sim}_1(1^\kappa); out' \leftarrow \mathsf{A}^{\widetilde{\mathsf{Sim}_2}(tk, \cdot)}(\mathsf{crs})\right\}$$

and $\widetilde{\mathsf{Sim}_2}(tk, (x, w))$ outputs $\mathsf{Sim}_2(tk, x)$ if $(x, w) \in \mathfrak{R}$, and $\perp$ otherwise.

Simulation extractability: There exists a PPT algorithm $\mathsf{Xtr} = (\mathsf{Xtr}_1, \mathsf{Xtr}_2)$ such that, for all PPT adversaries A, we have that

$$\mathbb{P}\left[\begin{array}{c}(\mathsf{crs}, tk, xk) \leftarrow \mathsf{Xtr}_1(1^k); (x, \pi) \leftarrow \mathsf{A}^{\mathsf{Sim}_2(tk, \cdot)}(\mathsf{crs});\\ w \leftarrow \mathsf{Xtr}_2(xk, (x, \pi)); (x, w) \notin \mathfrak{R} \wedge (x, \pi) \notin \mathcal{Q} \wedge \mathsf{Ver}(\mathsf{crs}, (x, \pi)) = 1\end{array}\right]$$

is negligible, where the list $\mathcal{Q}$ contains the successful pairs (x_i, π_i) that A has queried to Sim_2. We say that $\mathcal{NIZK}$ is *true* simulation-extractable if oracle $\mathsf{Sim}_2(tk, x)$ is replaced by $\widetilde{\mathsf{Sim}_2}(tk, (x, w))$ that outputs the same as $\mathsf{Sim}_2(tk, x)$ if and only if $(x, w) \in \mathfrak{R}$ (and outputs $\perp$ otherwise).

3 One-More Unforgeability

A signature scheme is a triple of algorithms $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ defined as follows: (1) The key generation algorithm takes as input the security parameter κ and outputs a verification key/signing key pair $(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$; (2) The signing algorithm takes as input a message $m \in \mathcal{M}$ and the signing key sk and outputs a signature $\sigma \leftarrow \mathsf{Sign}(sk, m)$; (3) The verification algorithm takes as input the verification key vk and a pair (m, σ) and outputs $\mathsf{Verify}(vk, (m, \sigma)) \in \{0, 1\}$. We denote by $|\sigma|$ the size of a signature output via $\mathsf{Sign}(sk, \cdot)$.

Given a signature scheme $\mathcal{SS}$, consider the following experiment $\mathsf{Exp}^{\mathsf{one-more}}_{\mathcal{SS},\mathsf{A}}(\kappa, \ell, \gamma)$ running with a PPT adversary A and parametrized by the security parameter $\kappa \in \mathbb{N}$, the leakage bound $\ell \in \mathbb{N}$ and the slack parameter $\gamma \in (0, 1]$:

1. Compute $(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$ and give vk to A.
2. The adversary A can adaptively access oracles $\mathsf{Sign}(sk, \cdot)$ and $\mathcal{O}^\ell(sk, \cdot)$, where $\mathcal{O}^\ell(sk, f)$ returns $f(sk)$. We let $\Lambda \in \{0, 1\}^\lambda$ be the total information returned by $\mathcal{O}^\ell$ (with $\lambda \leq \ell$), and we write $\mathcal{Q}$ for the set of messages A forwarded to the signing oracle.
3. A outputs n pairs $(m_1, \sigma_1), \ldots, (m_n, \sigma_n)$.
4. The experiment outputs 1 iff if the following conditions are satisfied:
   (a) $\mathsf{Verify}(vk, (m_i, \sigma_i)) = 1$ and $m_i \notin \mathcal{Q}$, for all $i \in [n]$.
   (b) The messages $m_1, \ldots, m_n$ are pairwise distinct.
   (c) $n \geq \lfloor \lambda/(\gamma|\sigma|) \rfloor + 1$.

Definition 1 (One-more unforgeability). *We say that* $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ *is* $(\ell, \gamma, \varepsilon)$-one-more unforgeable *if for every PPT adversary* A *we have that* $\mathbb{P}[\mathsf{Exp}^{\mathsf{one-more}}_{\mathcal{SS},\mathsf{A}}(\kappa, \ell, \gamma) = 1] \leq \varepsilon$. *Whenever* ε *is negligible in the security parameter, we simply say that* $\mathcal{SS}$ *is* (ℓ, γ)*-one-more unforgeable.*

Remark 1 (on γ). The parameter γ specifies how close to optimal security $\mathcal{SS}$ is. In particular, in case $\gamma = 1$ one-more unforgeability requires that A cannot forge even a single signature more than what it could have leaked via leakage queries. As γ decreases, so does the strength of the signature scheme (the extreme case being $\gamma = |\mathcal{M}|^{-1}$, where we have no security).

Note that the number of signatures the adversary has to forge depends on the length of the leakage he asks to see. In particular (ℓ, γ)-one-more unforgeability implies standard unforgeability for any adversary asking no leakage ($\lambda = 0$).

Finally, we remark that for any $\gamma \in (0, 1]$ we have that (ℓ, γ)-one-more unforgeability implies (ℓ', γ)-one-more unforgeability for all $\ell' \leq \ell$.

3.1 An Alternative Definition

Definition 1 may seem a weak security guarantee for a signature scheme, as an adversary is able to forge a certain number of signatures. If the messages to forge could be chosen at will at any time, this would be a rather useless security guarantee. Here, we state a seemingly stronger flavour of one-more unforgeability where a simulator can look at the state of the adversary after he is done with leakage queries and output a set $\mathcal{Q}^* \subset \mathcal{M}$, of size less than n, thought of as the messages corresponding to the forgeries leaked so far; now the adversary is successful if he can produce a forgery for a message of his choice not contained in $\mathcal{Q}^*$ (and not already asked to the signing oracle). In a certain sense, we get a notion that is similar to the standard unforgeability notion, with the twist that the adversary can ask a few extra signing queries (via leakage queries, though).

Given a signature scheme $\mathcal{SS}$, consider the experiment $\mathsf{Exp}^{\mathsf{poly-sim-one-more}}_{\mathcal{SS},\mathsf{A},\mathsf{S}}(\kappa, \ell, \gamma)$ below, running with a PPT adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ and a PPT simulator S and parametrized by the security parameter $\kappa \in \mathbb{N}$, the leakage bound $\ell \in \mathbb{N}$ and the slack parameter $\gamma \in (0, 1]$:

1. Compute $(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$ and give vk to A.
2. The adversary A_1 can adaptively access oracles $\mathsf{Sign}(sk, \cdot)$ and $\mathcal{O}^\ell(sk, \cdot)$, where $\mathcal{O}^\ell(sk, f)$ returns $f(sk)$. We let $\Lambda \in \{0,1\}^\lambda$ be the total information returned by $\mathcal{O}^\ell$ (with $\lambda \leq \ell$), and we write $\mathcal{Q}$ for the set of messages A_1 forwarded to the signing oracle.
3. Let st be the state of A_1 at the end of step 2 above, i.e., all his inputs, all his random choices, and all replies from the oracles. The simulator is given st and outputs $\mathcal{Q}^* \leftarrow \mathsf{S}(1^\kappa, vk, st)$ such that $\mathcal{Q}^* \subset \mathcal{M}$ and $|\mathcal{Q}^*| \leq \lfloor \lambda/(\gamma|\sigma|) \rfloor$.
4. A_2 is given $\mathcal{Q}^*$ and st and outputs a forgery (m^*, σ^*).
5. The experiment outputs 1 iff $\mathsf{Verify}(vk, (m^*, \sigma^*)) = 1$ and $m^* \notin \mathcal{Q} \cup \mathcal{Q}^*$.

Definition 2 (Poly-constrained one-more unforgeability). *We say that* $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ *is* $(\ell, \gamma, \varepsilon)$-poly-constrained one-more unforgeable *if for every PPT adversary* A *there exists a PPT simulator* S *such that*

$$\mathbb{P}[\mathsf{Exp}^{\mathsf{poly-sim-one-more}}_{\mathcal{SS},\mathsf{A},\mathsf{S}}(\kappa, \ell, \gamma) = 1] \leq \varepsilon.$$

Whenever ε *is negligible in the security parameter, we simply say that* $\mathcal{SS}$ *is* (ℓ, γ)*-poly-constrained one-more unforgeable.*

3.2 Yet Another Alternative Definition

Definition 2 requires that $\mathcal{Q}^*$ can be computed in poly-time, effectively requiring that the adversary *knows* the small set of forgeries he leaked. In most applications we are aware of, it seems, however, enough that such a small set *exists*. And, there seems to be a difference between these notions. Consider an adversary who leaks a few values of the form $v_i = H(m_i) \oplus \sigma_i$, where H is a hash function, for random messages m_i (with $i \in [n]$) and σ_i a signature on m_i. Given any m_i as input it can compute a "forgery" $\sigma_i = v_i \oplus H(m_i)$, but until it is given m_i it does not know the set of messages it can forge signatures on, at least it would be hard to compute this set efficiently in a black-box manner. We formulate a security notion which still considers leakage of a few such "unknown" σ_i as benign.

We simply restate Definition 2, but we now allow S unbounded computing time. We can massage this relaxed definition a bit to get a simpler, equivalent definition. Consider the following generic simulator $\mathsf{S}_{\min}(1^\kappa, vk, st)$: it iterates over all $\mathcal{Q}^* \subset \mathcal{M}$ with $|\mathcal{Q}^*| \leq \lfloor \lambda/(\gamma|\sigma|) \rfloor$ and computes the probability $p_{\mathcal{Q}^*}$ that $\mathsf{A}_2(\mathcal{Q}^*, st)$ outputs (m^*, σ^*) such that $\mathsf{Verify}(vk, (m^*, \sigma^*)) = 1$ and $m^* \notin \mathcal{Q} \cup \mathcal{Q}^*$. It then outputs the $\mathcal{Q}^*$ minimizing $p_{\mathcal{Q}^*}$. It is clear that if for some adversary A there exists an unbounded simulator S fulfilling Definition 2 for A, then also $\mathsf{S}_{\min}$ will fulfil Definition 2 for A. Hence we can equivalently hardwire $\mathsf{S}_{\min}$ into the definition. If we at the same time use that the expected value of a random value over $\{0,1\}$ is equal to the probability that it is 1, we get the below more compact definition. Consider the following experiment $\mathsf{Exp}^{\mathsf{sim-one-more}}_{\mathcal{SS},\mathsf{A}}(\kappa, \ell, \gamma)$:

1. Compute $(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$ and give vk to A_1.
2. The adversary A_1 can adaptively access oracles $\mathsf{Sign}(sk, \cdot)$ and $\mathcal{O}^\ell(sk, \cdot)$, where $\mathcal{O}^\ell(sk, f)$ returns $f(sk)$. We let $\Lambda \in \{0,1\}^\lambda$ be the total information returned by $\mathcal{O}^\ell$ (with $\lambda \leq \ell$), and we write $\mathcal{Q}$ for the set of messages A_1 forwarded to the signing oracle.
3. Let st be the state of A_1 at the end of step 2 above.
4. Output

$$\min_{\substack{\mathcal{Q}^* \subset \mathcal{M}: \\ |\mathcal{Q}^*| \leq \lfloor \lambda/(\gamma|\sigma|) \rfloor}} \left(\mathbb{P}[(m^*, \sigma^*) \leftarrow \mathsf{A}_2(\mathcal{Q}^*, st) : \mathsf{Verify}(vk, (m^*, \sigma^*)) \wedge m^* \notin \mathcal{Q} \cup \mathcal{Q}^*] \right) .$$

Definition 3 (Constrained one-more unforgeability). *We say that $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ is $(\ell, \gamma, \varepsilon)$*-constrained one-more unforgeable *if it holds that $\mathbb{E}[\mathsf{Exp}^{\mathsf{sim-one-more}}_{\mathcal{SS},\mathsf{A}}(\kappa, \ell, \gamma)] \leq \varepsilon$ for every PPT adversary A, where the expected value is over the random choices used to generate (vk, sk) and the random choices of A_1. Whenever ε is negligible in the security parameter, we simply say that $\mathcal{SS}$ is (ℓ, γ)-constrained one-more unforgeable.*

3.3 Equivalence of two Definitions

We argue below that one-more unforgeability and constrained-one-more unforgeability are equivalent. It is clear that security under Definition 2 implies security under Definition 3. We conjecture that Definition 3 is strictly weaker than Definition 2.

Theorem 1. *Definition 1 and Definition 3 are equivalent up to a constant factor 4 in security.*

Proof. For space reasons, we prove only that Definition 1 implies Definition 3; the proof of the other direction can be found in the full version. We give a proof by contradiction. Assume there exists a polynomial ε and PPT adversary $\mathsf{A}' = (\mathsf{A}'_1, \mathsf{A}'_2)$ such that $\mathbb{E}[\mathsf{Exp}^{\mathsf{sim-one-more}}_{\mathcal{SS},\mathsf{A}'}(\kappa, \ell, \gamma)] > \varepsilon$ for infinitely many values of κ.

Since $0 \leq \mathbb{E}[\mathsf{Exp}^{\mathsf{sim-one-more}}_{\mathcal{SS},\mathsf{A}'}(\kappa, \ell, \gamma)] \leq 1$ this implies that $\mathbb{P}[\mathsf{Exp}^{\mathsf{sim-one-more}}_{\mathcal{SS},\mathsf{A}'}(\kappa, \ell, \gamma) \geq \varepsilon/2] \geq \varepsilon/2$ for infinitely many values of κ. Let E be the event that $\mathbb{P}[\mathsf{Exp}^{\mathsf{sim-one-more}}_{\mathcal{SS},\mathsf{A}'}(\kappa, \ell, \gamma) \geq \varepsilon/2]$.

We now describe A, running in experiment $\mathsf{Exp}^{\mathsf{one-more}}_{\mathcal{SS},\mathsf{A}}(\kappa, \ell, \gamma)$. When reading the description keep in mind that it is defined to work when E occurs.

1. Receive the verification key vk and initialize $\mathcal{Q}^* = \emptyset$.
2. Run $\mathsf{A}'_1(1^\kappa, vk)$ and simulate leakage queries and signature queries using oracles $\mathcal{O}^\ell(sk, \cdot)$ and $\mathsf{Sign}(sk, \cdot)$. Let $\Lambda \in \{0,1\}^\lambda$ be the overall information retrieved by A'_1.
3. Define $n := \lfloor \lambda/(\gamma|\sigma|) \rfloor + 1$. Repeat the following steps, for $i = 1, \ldots, n$:
   (a) Run $8(\log_2(n) + \kappa)/\varepsilon$ copies of $\mathsf{A}'_2(1^\kappa, st, \mathcal{Q}^*)$ in parallel. If any of the copies outputs (m_i^*, σ_i^*) such that $\mathsf{Verify}(vk, (m^*, \sigma^*)) = 1$ and $m^* \notin \mathcal{Q} \cup \mathcal{Q}^*$, then go to the next step, otherwise give up and terminate.
   (b) Set $\mathcal{Q}^* := \mathcal{Q}^* \cup \{m_i^*\}$ for one of the forgeries from above.
4. Output $(m_1^*, \sigma_1^*), \ldots, (m_n^*, \sigma_n^*)$.

Assume that E occurs. Then the probability that any copy $\mathsf{A}'_2(1^\kappa, st, \mathcal{Q}^*)$ in Step 1 outputs (m_i^*, σ_i^*) such that $\mathsf{Verify}(vk, (m^*, \sigma^*)) = 1$ and $m^* \notin \mathcal{Q} \cup \mathcal{Q}^*$ is $\geq \varepsilon/2$. Hence one of the copies will output such (m_i^*, σ_i^*), except with probability $2^{-\log_2(n)-\kappa}$, by construction. Thus, by a union bound, the probability that A gives up in any of the iterations is at most $n \cdot 2^{-\log_2(n)-\kappa} = 2^{-\kappa}$.

Clearly, when A does not give up in any of the iterations, we have that $\mathsf{Exp}^{\mathsf{one-more}}_{\mathcal{SS},\mathsf{A}}(\kappa, \ell, \gamma) = 1$. Hence $\mathbb{P}[\mathsf{Exp}^{\mathsf{one-more}}_{\mathcal{SS},\mathsf{A}}(\kappa, \ell, \gamma) = 1] \geq \mathbb{P}[E](1 - 2^{-\kappa}) = \varepsilon(1 - 2^{-\kappa})/2 > \varepsilon/4$ for infinitely many values of κ. This concludes the proof as A is PPT.

4 Construction

We give a construction of a one-more unforgeable signature scheme (cf. Definition 1) based on the following building blocks:

- A non-interactive zero knowledge argument of knowledge system $\mathcal{NIZK} = (\mathsf{Init}, \mathsf{Prove}, \mathsf{Ver})$.
- A perfectly hiding and computationally binding, linearly homomorphic[7] commitment scheme $\mathcal{COM} = (\mathsf{Setup}, \mathsf{Commit})$, with message and randomness space equal to a finite field $\mathbb{F}$.

Our scheme $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ has message space equal to $\mathbb{F}$ and is described below:

Key Generation. Run $pk \leftarrow \mathsf{Setup}(1^\kappa)$ and $\mathsf{crs} \leftarrow \mathsf{Init}(1^\kappa)$. For some parameter $d \in \mathbb{N}$, sample $\delta_0, \ldots, \delta_d$ and $r_0, \ldots, r_d$ uniformly from $\mathbb{F}$, and compute commitments $com_i = \mathsf{Commit}(pk, \delta_i; r_i)$ for $i = 0, \ldots, d$. Let $\boldsymbol{\delta} = (\delta_0, \ldots, \delta_d)$ and $\mathbf{r} = (r_0, \ldots, r_d)$; output $sk = (\boldsymbol{\delta}, \mathbf{r})$ and $vk = (\mathsf{crs}, pk, \{com_i\}_{i=0}^d)$.

Signature. To sign a message $m \in \mathbb{F}$, let $\delta(X)$ be the degree d polynomial having δ_i's as coefficients, i.e. $\delta(X) = \sum_{i=0}^d \delta_i \cdot X^i$. Consider the following polynomial-time relation:

$$\mathfrak{R} := \{(pk, com^*); (\tilde{m}, \tilde{r}) : \; com^* = \mathsf{Commit}(pk, \tilde{m}; \tilde{r})\} \; .$$

[7] For notational convenience, we assume that the product of commitments give commitments to the sum of messages using the sum of the randomness as randomness, *à la* Pedersen [32].

Compute $\tilde{m} = \delta(m)$ and $\tilde{r} = \sum_{i=0}^{d} r_i \cdot m^i$. Note that both values $\tilde{m}$, $\tilde{r}$ can be computed efficiently as a function of the signing key $(\boldsymbol{\delta}, \mathbf{r})$ and the message to be signed. Using crs as common reference string, generate a NIZK argument π that $(pk, \prod_{i=0}^{d}(com_i)^{m^i}) \in \mathfrak{L}_{\mathfrak{R}}$, the language generated by the above relation $\mathfrak{R}$. Output $\sigma = \pi$.

Verification. Given a pair (m, σ), parse σ as $\sigma = \pi$ and compute $com^* = \prod_{i=0}^{d}(com_i)^{m^i}$. Output the same as $\mathsf{Ver}(\mathsf{crs}, \pi, (pk, com^*))$.

Let us first argue that the signature scheme satisfies the correctness property. This follows from the fact that $\mathcal{COM}$ is linearly homomorphic (cf. Section 2.2):

$$com^* = \prod_{i=0}^{d}(com_i)^{m^i} = \prod_{i=0}^{d} \mathsf{Commit}(\delta_i \cdot m^i; r_i \cdot m^i) = \mathsf{Commit}\big(\underbrace{\sum_{i=0}^{d} \delta_i \cdot m^i}_{\tilde{m}}; \underbrace{\sum_{i=0}^{d} r_i \cdot m^i}_{\tilde{r}}\big).$$

We prove the following result:

Theorem 2. *Assume that $\mathcal{COM}$ is perfectly hiding and computationally binding, and that $\mathcal{NIZK}$ is a NIZK argument of knowledge system for relation $\mathfrak{R}$. Then the scheme $\mathcal{SS}$ described above is (ℓ, γ)-one-more unforgeable, as long as*

$$\ell = d \log |\mathbb{F}| \quad \textit{and} \quad \gamma = \frac{\log |\mathbb{F}|}{|\sigma|}.$$

4.1 Proof of Theorem 2

To prove the theorem we will rely on the following property of any perfectly hiding commitment scheme $\mathcal{COM} = (\mathsf{Setup}, \mathsf{Commit})$. Define the following experiment $\mathsf{Exp}^{\mathsf{guess}}_{\mathcal{COM},\mathsf{A}}(\kappa, \ell, d)$, featuring an unbounded adversary A:

1. Run $pk \leftarrow \mathsf{Setup}(1^\kappa)$ and sample $x_1, \ldots, x_d \in \mathcal{M}$ uniformly at random. Compute $com_i = \mathsf{Commit}(pk, x_i; r_i)$ and give $(\{com_i\}_{i=1}^{d}, pk)$ to A. Store $s = (\{x_i\}_{i=1}^{d}, \{r_i\}_{i=1}^{d})$.
2. The adversary can access adaptively oracle $\mathcal{O}^{\ell}(s, \cdot)$. Let $\Lambda \in \{0,1\}^\lambda$ be the overall information retrieved by A (with $\lambda \leq \ell$).
3. The adversary can open a subset of size t of $(x_1, \ldots, x_d)$: Given a set of indexes $(i_1, \ldots, i_t)$ such that each $i_j \in [d]$, the values $(\{x_{i_j}\}_{j=1}^{t}, \{r_{i_j}\}_{j=1}^{t})$ are forwarded to A.
4. The experiment returns 1 if A outputs the remaining values x_i, for all $i \in [d] \setminus \{i_1, \ldots, i_t\}$.

Lemma 2. *Let $\mathcal{COM} = (\mathsf{Setup}, \mathsf{Commit})$ be a perfectly hiding commitment scheme with message space $\mathcal{M}$. Then for every computationally unbounded adversary A we have that*

$$\mathbb{P}\left[\mathsf{Exp}^{\mathsf{guess}}_{\mathcal{COM},\mathsf{A}}(\kappa, \ell, d) = 1\right] \leq \frac{2^\lambda}{|\mathcal{M}|^{d-t}}.$$

The proof of Lemma 2 appears in the full version of this paper.

We now prove Theorem 2. Let A be a PPT machine running in experiment $\mathsf{Exp}^{\mathsf{one-more}}_{\mathcal{SS},\mathsf{A}}(\kappa,\ell,\gamma)$. We recall how the experiment is held for our scheme $\mathcal{SS}$.

1. The signing key $sk = (\boldsymbol{\delta},\mathbf{r})$ and the verification key $vk = (pk, \{com_i\}_{i=0}^{d})$ are computed. In particular, $pk \leftarrow \mathsf{Setup}(1^\kappa)$ and $\mathsf{crs} \leftarrow \mathsf{Init}(1^\kappa)$. Here, $\boldsymbol{\delta} = (\delta_0,\dots,\delta_d) \leftarrow \mathbb{F}^{d+1}$, $com_i = \mathsf{Commit}(\delta_i; r_i)$ and $\mathbf{r} = (r_0,\dots,r_d) \leftarrow \mathbb{F}^{d+1}$.
2. The adversary A is given vk and can access oracles $\mathsf{Sign}(sk,\cdot)$ and $\mathcal{O}^\ell(sk,\cdot)$.
   - The signing oracle $\mathsf{Sign}(sk,m)$ computes $\tilde{m} = \sum_{i=0}^{d}\delta_i \cdot m^i$ and $\tilde{r} = \sum_{i=0}^{d} r_i \cdot m^i$, together with an argument π that $(pk, com^*) \in \mathcal{L}_{\mathfrak{R}}$ for $com^* = \mathsf{Commit}(\tilde{m};\tilde{r})$; hence, it returns $\sigma = \pi$.
   - The leakage oracle $\mathcal{O}^\ell(sk,f)$ returns $f(sk)$.
3. A outputs n pairs $(m_1,\pi_1),\dots,(m_n,\pi_n)$.
4. The experiment outputs 1 iff the following conditions are satisfied:
   (a) $\mathsf{Verify}(vk,(m_i,\sigma_i)) = 1$ and $m_i \notin \mathcal{Q}$, for all $i \in [n]$.
   (b) The messages $m_1,\dots,m_n$ are pairwise distinct.
   (c) $n \geq \lfloor \lambda/(\gamma|\sigma|)\rfloor + 1$.

The proof proceeds by a series of games.

Game_0. This is the real experiment, as described above.

Game_1. This game is identical to Game_0, but we replace the Init algorithm with $(\mathsf{crs}, tk) \leftarrow \mathsf{Sim}_1(1^\kappa)$. Moreover, each time a signing query for message m is asked, we simulate the argument by running $\pi \leftarrow \mathsf{Sim}_2(tk,(pk,com^*))$. Everything else remains the same.
By a standard argument, the multi-theorem zero-knowledge property of the argument system implies that $\mathbb{P}[\mathsf{A}\text{ wins }\mathsf{Game}_0]$ is negligibly close to $\mathbb{P}[\mathsf{A}\text{ wins }\mathsf{Game}_1]$.

Game_2. This game is identical to Game_1, but the common reference string is sampled as $(\mathsf{crs}, tk, xk) \leftarrow \mathsf{Xtr}_1(1^\kappa)$ and before outputting 1 we check that all arguments contained in A's forgeries can be extracted via $\mathsf{Xtr}_2(xk,\cdot)$.
By a standard argument, (true) simulation extractability of $\mathcal{NIZK}$ implies that $\mathbb{P}[\mathsf{A}\text{ wins }\mathsf{Game}_1]$ is negligibly close to $\mathbb{P}[\mathsf{A}\text{ wins }\mathsf{Game}_2]$.

Now, we show that $\mathbb{P}[\mathsf{A}\text{ wins }\mathsf{Game}_2]$ is negligible which proves the theorem. Define the following event Bad in Game_2: The event occurs whenever for at least one of the forgeries (m_j,σ_j) returned by A it holds that $\tilde{m}'_j \neq \sum_{i=0}^{d}\delta_i \cdot m_j^i$ for $(\tilde{m}'_j,\tilde{r}'_j) \leftarrow \mathsf{Xtr}_2(xk,\pi_j)$. In other words, there exists a valid pair (m_j,σ_j) for which the extracted value $\tilde{m}'_j$ is not the evaluation of m_j through the polynomial $\delta(X)$ having $\boldsymbol{\delta}$ as coefficients. We write

$$\begin{aligned}\mathbb{P}[\mathsf{A}\text{ wins }\mathsf{Game}_2] &\leq \mathbb{P}[\mathsf{A}\text{ wins }\mathsf{Game}_2 \wedge Bad] + \mathbb{P}\left[\mathsf{A}\text{ wins }\mathsf{Game}_2 \wedge \overline{Bad}\right]\\ &\leq negl(\kappa),\end{aligned}$$

where the last inequality comes from the two claims below.

Claim. $\mathbb{P}\left[\mathsf{A}\text{ wins }\mathsf{Game}_2 \wedge \overline{Bad}\right] \leq 2^\lambda/|\mathbb{F}|^n$.

Proof. By contradiction, assume that $\mathbb{P}\left[\mathsf{A} \text{ wins } \mathsf{Game}_2 \wedge \overline{\mathit{Bad}}\right] > 2^\lambda/|\mathbb{F}|^n$. We build a PPT reduction B (running A) which wins the game of experiment $\mathsf{Exp}^{\mathsf{guess}}_{\mathcal{COM},\mathsf{B}}(\kappa, \ell, d+1)$ with at least the same advantage.

1. Given $\{com_i = \mathsf{Commit}(\delta_i; r_i)\}_{i=0}^d$ as input, implicitly define $sk := (\boldsymbol{\delta}, \mathbf{r})$, where
$$\boldsymbol{\delta} = (\delta_0, \delta_1, \ldots, \delta_d) \qquad \mathbf{r} = (r_0, r_1, \ldots, r_d).$$
Run $(\mathsf{crs}, tk, xk) \leftarrow \mathsf{Xtr}_1(1^\kappa)$ and give $vk := (\mathsf{crs}, pk, \{com_i\}_{i=0}^d)$ to A.
2. Whenever A asks a leakage query f to $\mathcal{O}^\ell((\boldsymbol{\delta}, \mathbf{r}), \cdot)$, forward the same query to $\mathcal{O}^\ell(s, \cdot)$ (where $s = (\boldsymbol{\delta}, \mathbf{r})$). Give to A the same value returned by $\mathcal{O}^\ell(s)$.
3. Whenever A asks a signing query m to $\mathsf{Sign}(sk, \cdot)$, answer as follows. Simulate an argument $\pi \leftarrow \mathsf{Sim}(tk, (pk, com^*))$ where $com^* = \prod_{i=0}^d (com_i)^{m^i}$. Give $\sigma = \pi$ to A.
4. Let $(m_1, \sigma_1), \ldots, (m_n, \sigma_n)$ be the forgeries output by A. Ask to open the last $t := d+1-n$ values, i.e. $(\{\delta_i\}_{i=n}^d, \{r_i\}_{i=n}^d)$. Set $\tilde{\delta}_i = \delta_i$ for each $i = n, \ldots, d$.
5. For each of the values $\sigma_i = \pi_i$ returned by A compute $(\tilde{m}'_i, \tilde{r}'_i) \leftarrow \mathsf{Xtr}_2(xk, \pi_i)$. Solve the following linear system:
$$\begin{pmatrix} 1 \ m_1 \ \ldots \ m_1^{n-1} \\ \ddots \\ 1 \ m_n \ \ldots \ m_n^{n-1} \end{pmatrix} \cdot \begin{pmatrix} \delta_0 \\ \vdots \\ \delta_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_{n-1} \end{pmatrix}, \qquad (1)$$
where each of the values y_i is computed from known values as $y_i = \tilde{m}'_i - \sum_{j=n}^d \tilde{\delta}_j \cdot m_i^j$.
6. Output $(\delta_1, \ldots, \delta_n)$.

Note that B perfectly simulates the environment for A in Game_2. The choice of the parameters $\gamma = \log|\mathbb{F}|/|\sigma|$ and $\ell = d\log|\mathbb{F}|$ (as in the theorem statement), ensures that $1 \le n \le d+1$. In particular, the total number of field elements known by B is at most $\frac{\lambda}{\log|\mathbb{F}|} + \left(d - \left\lfloor \frac{\lambda}{\gamma|\sigma|} \right\rfloor\right) < d+1$, and thus there is some entropy left in the commitments.

Moreover, as the values $(m_1, \ldots, m_n)$ are pairwise distinct, the matrix of Eq. (1) has full rank and the linear system always admits a solution. Since the event *Bad* does not happen, we have that $\tilde{m}'_i = \tilde{m}_i = \delta(m_i)$ (for all $i \in [n]$), and thus the solution $(\delta_1, \ldots, \delta_n)$ corresponds to the same elements in the vector $\boldsymbol{\delta}$. The above contradicts Lemma 2, as
$$\mathbb{P}\left[\mathsf{Exp}^{\mathsf{guess}}_{\mathcal{COM},\mathsf{B}}(\kappa, \ell, d+1) = 1\right] \ge \mathbb{P}\left[\mathsf{A} \text{ wins } \mathsf{Game}_2 \wedge \overline{\mathit{Bad}}\right] > \frac{2^\lambda}{|\mathbb{F}|^n}.$$

Claim. $\mathbb{P}[\mathsf{A} \text{ wins } \mathsf{Game}_2 \wedge \mathit{Bad}] \le \mathit{negl}(\kappa)$.

Proof. Assume that $\mathbb{P}[\mathsf{A} \text{ wins } \mathsf{Game}_2 \wedge \mathit{Bad}] > 1/\mathit{poly}(\kappa)$ for infinitely many κ. We build an attacker C breaking the binding property of the commitment scheme $\mathcal{COM}$ with non-negligible advantage. A description of C follows:

1. Receive the public parameter pk for $\mathcal{COM}$. Choose $\delta_0, \ldots, \delta_d \leftarrow \mathbb{F}$ and compute commitments $com_i = \mathsf{Commit}(pk, \delta_i; r_i)$ for randomly chosen r_0, $\ldots, r_d \leftarrow \mathbb{F}$. Generate $(\mathsf{crs}, tk, xk) \leftarrow \mathsf{Xtr}_1(1^\kappa)$.
2. Run A with input $vk := (\mathsf{crs}, pk, \{com_i\}_{i=0}^d)$ and answer signing/leakage queries from A as it would be done in Game_2.
3. When A outputs $(m_1, \pi_1), \ldots, (m_n, \pi_n)$, extract the witness $(\tilde{m}'_i, \tilde{r}'_i)$ from each argument of knowledge π_i, for $i \in [n]$.
4. If there exists an index $j \in [n]$ such that $\tilde{m}'_j \neq \delta(m_j)$, compute $com^*_j = \prod_{i=0}^d (com_i)^{m_j^i}$, $\tilde{m}_j = \sum_{i=0}^d \delta_i \cdot m_j^i$ and $\tilde{r}_j = \sum_{i=0}^d r_i \cdot m_j^i$ and output $(com^*_j, (\tilde{m}_j, \tilde{r}_j), (\tilde{m}'_j, \tilde{r}'_j))$; otherwise abort and output $\perp$.

Notice that if Game_2 outputs 1 and event *Bad* occurs, then C outputs a valid pair breaking the binding property of $\mathcal{COM}$ with non-negligible probability (a contradiction). This is because both $(\tilde{m}_j, \tilde{r}_j)$ and $(\tilde{m}'_j, \tilde{r}'_j)$ are valid openings for com^*_j and moreover *Bad* implies that $(\tilde{m}_j, \tilde{r}_j) \neq (\tilde{m}'_j, \tilde{r}'_j)$.

4.2 A Concrete Instantiation

In this section we show how to instantiate our signature scheme, reducing security to the DLIN assumption [5]. For each of the building blocks we present an instantiation and concrete parameters.

In the following let $\mathbb{G}$ be a cyclic group of order a prime number q. Before introducing our concrete construction, let us recall the DLIN assumption:

Definition 4. *The* DLIN assumption *states that for any PPT algorithm* A *it holds that*

$$\left|\mathbb{P}\left[\mathsf{A}(\mathbb{G}, (g, g_1, g_2, g_1^a, g_2^b, g^c)) = 1\right] - \mathbb{P}\left[\mathsf{A}(\mathbb{G}, (g, g_1, g_2, g_1^a, g_2^b, g^{a+b}) = 1\right]\right| \leq negl(\kappa),$$

where $g, g_1, g_2 \leftarrow \mathbb{G}$ *and* $a, b, c \leftarrow \mathbb{F}_q$.

$\mathcal{COM}$: We use Pedersen commitments. The setup algorithm Setup outputs public parameters $pk = (h_1, h_2)$, where h_1 is a generator for $\mathbb{G}$ and $h_2 = h_1^a$ for a random $a \in \mathbb{F}_q$. The commitment to an element $m \in \mathbb{F}_q$ using randomness $r \leftarrow \mathbb{F}_q$ is computed as $com = \mathsf{Commit}(pk, m; r) := h_1^m \cdot h_2^r$. Whenever we want to open the commitment, we reveal (m, r).
Note that Pedersen commitment is linearly homomorphic: given $com_1 = \mathsf{Commit}(m_1; r_1)$ and $com_2 = \mathsf{Commit}(m_2; r_2)$ it holds that

$$com_1 \cdot com_2 = h_1^{m_1+m_2} \cdot h_2^{r_1+r_2} = \mathsf{Commit}(m_1 + m_2; r_1 + r_2).$$

Moreover, for all constants $c \in \mathbb{F}_q$ we have that $com^c = h_1^{c \cdot m} \cdot h_2^{c \cdot r} = \mathsf{Commit}(c \cdot m; c \cdot r)$.

$\mathcal{NIZK}$: Recall that our relation is as follows:

$$\mathfrak{R} = \{(pk, com^*); (\tilde{m}, \tilde{r}) : \; com^* = \mathsf{Commit}(pk, \tilde{m}; \tilde{r})\}\,.$$

When using Pedersen commitment, we get

$$com^* = \prod_{i=0}^{d}(com_i)^{m^i} = \prod_{i=0}^{d}\left(h_1^{\delta_i + ar_i}\right)^{m^i} = h_1^{\sum_{i=0}^{d}\delta_i m^i + ar_i m^i} = h_1^{\tilde{m} + a\cdot\tilde{r}}.$$

Thus, we can reduce the proof of knowledge of an opening for com^* to the proof of knowledge of a discrete logarithm. Groth [18] gives a simulation-extractable NIZK for proving knowledge of discrete logarithms of a group element. We remark that the length of a proof is constant, and in particular independent of the degree d of the polynomial.
Alternatively, as true simulation-extractability is sufficient for our construction, one could instantiate the NIZK using the transformation of [10], which requires a standard (non-simulation-extractable) NIZK and a labeled CCA-secure encryption scheme.

5 Application to Leaky Identification

We show how to apply one-more unforgeability to the context of (leaky) identification protocols. In a public key identification scheme a prover with public key vk attempts to prove its identity to a verifier holding vk. More formally, an identification scheme $\mathcal{ID} = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{P}, \mathsf{V})$ consists of four PPT algorithms described as follows: (1) The parameters generation algorithm takes as input the security parameter and outputs public parameters $\mathsf{params} \leftarrow \mathsf{PGen}(1^\kappa)$, shared by all users.[8] (2) The key generation algorithm takes as input the security parameter and outputs a verification key/secret key pair $(vk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$. (3) P and V are probabilistic Turing machines interacting in a protocol $(\mathsf{P}(sk) \rightleftarrows \mathsf{V})(vk)$; at the end of the execution V outputs a judgment $d \in \{0, 1\}$, where $d = 1$ means that the identification was successful.

Following [1], we define a leaky variant of the standard notion of active security (dubbed active ℓ-security under pre-impersonation attacks with leakage), where an adversary, in a first stage, is given black-box access to the honest prover, and in a second stage is given one shot to convince the verifier. In the leaky case, during the first phase, the adversary can also access adaptively a leakage oracle $\mathcal{O}^\ell(sk)$.

We then show that the below standard way (see [2]) of constructing an identification scheme $\mathcal{ID}$ from a signature scheme $\mathcal{SS} = (\mathsf{KGen}', \mathsf{Sign}, \mathsf{Verify})$, achieves active ℓ-security under pre-impersonation attacks with leakage provided that $\mathcal{SS}$ is one-more unforgeable.

– *Parameters generation.* Algorithm PGen samples the public parameters params for the signature schemes (if any).
– *Key Generation.* Algorithm KGen runs the key generation algorithm of the signature scheme, obtaining $(vk, sk) \leftarrow \mathsf{KGen}'(1^\kappa)$.

[8] In what follows all algorithms take as input params, but we omit to explicitly write this for ease of notation.

– *Identification protocol.* The interaction is as follows: (a) The verifier sends a random $m^* \leftarrow \mathcal{M}$ to the prover; (b) The prover replies with $\sigma^* \leftarrow \mathsf{Sign}(sk, m^*)$; (c) The verifier outputs $\mathsf{Verify}(vk, (m^*, \sigma^*))$.

Theorem 3. *Assume that $\mathcal{SS}$ is (ℓ, γ)-constrained-one-more unforgeable. Then $\mathcal{ID}$ from above is actively ℓ-secure under pre-impersonation attacks with leakage.*

For space reasons, a formal definition and a proof of the above theorem are deferred to the full version of this paper.

References

1. Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *CRYPTO*, pages 36–54, 2009.
2. Mihir Bellare, Marc Fischlin, Shafi Goldwasser, and Silvio Micali. Identification protocols secure against reset attacks. In *EUROCRYPT*, pages 495–511, 2001.
3. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
4. Nir Bitansky, Ran Canetti, and Shai Halevi. Leakage-tolerant interactive protocols. In *TCC*, pages 266–284, 2012.
5. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO*, pages 41–55, 2004.
6. Elette Boyle, Gil Segev, and Daniel Wichs. Fully leakage-resilient signatures. In *EUROCRYPT*, pages 89–108, 2011.
7. Zvika Brakerski, Yael Tauman Kalai, Jonathan Katz, and Vinod Vaikuntanathan. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *FOCS*, pages 501–510, 2010.
8. Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In *SCN*, pages 121–137, 2010.
9. Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Cryptography against continuous memory attacks. In *FOCS*, pages 511–520, 2010.
10. Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Efficient public-key cryptography in the presence of key leakage. In *ASIACRYPT*, pages 613–631, 2010.
11. Yevgeniy Dodis, Allison B. Lewko, Brent Waters, and Daniel Wichs. Storing secrets on continually leaky devices. In *FOCS*, pages 688–697, 2011.
12. Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.
13. Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302, 2008.
14. Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy N. Rothblum. Leakage-resilient signatures. In *TCC*, pages 343–360, 2010.
15. Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the Fiat-Shamir transform. In *INDOCRYPT*, pages 60–79, 2012.
16. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

17. Sanjam Garg, Abhishek Jain, and Amit Sahai. Leakage-resilient zero knowledge. In *CRYPTO*, pages 297–315, 2011.
18. Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *ASIACRYPT*, pages 444–459, 2006.
19. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In *CRYPTO*, pages 97–111, 2006.
20. Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3):11, 2012.
21. Shai Halevi and Huijia Lin. After-the-fact leakage in public-key encryption. In *TCC*, pages 107–124, 2011.
22. Dennis Hofheinz, Tibor Jager, and Eike Kiltz. Short signatures from weaker assumptions. In *ASIACRYPT*, pages 647–666, 2011.
23. Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In *CRYPTO*, pages 293–308, 2005.
24. Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *ASIACRYPT*, pages 703–720, 2009.
25. Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. Efficient authentication from hard learning problems. In *EUROCRYPT*, pages 7–26, 2011.
26. Eike Kiltz, Krzysztof Pietrzak, and Mario Szegedy. Digital signatures with minimal overhead from indifferentiable random invertible functions. In *CRYPTO (1)*, pages 571–588, 2013.
27. Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *CRYPTO*, pages 104–113, 1996.
28. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *CRYPTO*, pages 388–397, 1999.
29. Tal Malkin, Isamu Teranishi, Yevgeniy Vahlis, and Moti Yung. Signatures resilient to continual leakage on memory and computation. In *TCC*, pages 89–106, 2011.
30. Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. *IACR Cryptology ePrint Archive*, 2009:105, 2009.
31. Jesper Buus Nielsen, Daniele Venturi, and Angela Zottarel. On the connection between leakage tolerance and adaptive security. In *Public Key Cryptography*, pages 497–515, 2013.
32. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO*, pages 129–140, 1991.
33. Krzysztof Pietrzak. A leakage-resilient mode of operation. In *EUROCRYPT*, pages 462–482, 2009.
34. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.
35. Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In *E-smart*, pages 200–210, 2001.
36. Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *CRYPTO*, pages 566–598, 2001.