An extended abstract of this paper is published in the proceedings of the 42nd International Colloquium on Automata, Languages, and Programming—ICALP 2015. This is the full version.

Fully Leakage-Resilient Signatures Revisited: Graceful Degradation, Noisy Leakage, and Construction in the Bounded-Retrieval Model

Antonio Faonio*[1], Jesper Buus Nielsen†[1], and Daniele Venturi[2]

[1]*Department of Computer Science, Aarhus University, Aarhus, Denmark*
[2]*Department of Computer Science, Sapienza University of Rome, Rome, Italy*

March 15, 2016

Abstract

We construct new leakage-resilient signature schemes. Our schemes remain unforgeable against an adversary leaking arbitrary (yet bounded) information on the entire state of the signer (sometimes known as *fully* leakage resilience), including the random coin tosses of the signing algorithm.

The main feature of our constructions is that they offer a graceful degradation of security in situations where standard existential unforgeability is impossible. This property was recently put forward by Nielsen, Venturi, and Zottarel (PKC 2014) to deal with settings in which the secret key is much larger than the size of a signature. One remarkable such case is the so-called Bounded-Retrieval Model (BRM), where one intentionally inflates the size of the secret key while keeping constant the signature size and the computational complexity of the scheme.

Our main constructions have leakage rate $1 - o(1)$, and are proven secure in the standard model. We additionally give a construction in the BRM, relying on a random oracle. All of our schemes are described in terms of generic building blocks, but also admit efficient instantiations under fairly standard number-theoretic assumptions. Finally, we explain how to extend some of our schemes to the setting of noisy leakage, where the only restriction on the leakage functions is that the output does not decrease the min-entropy of the secret key by too much.

Keywords: signature scheme; leakage resilience; noisy leakage; bounded retrieval model.

*Part of the work done while at Sapienza University of Rome.

†Partially supported by Danish Council for Independent Research via DFF Starting Grant 10-081612. Partially supported by the European Research Commission Starting Grant 279447.

Contents

1 Introduction

Cryptography relies on secret information and random sources to accomplish its tasks. In order for a given cryptographic primitive to be secure, it is typically required that its secrets and randomness are well-protected, and cannot be influenced by an attacker. In practice, however, it is not always possible to fulfill this requirement, and partial information about the secret state of a cryptosystem can leak to an external adversary, e.g., via so-called side-channel attacks exploiting physical characteristics of a crypto-device, such as power consumption [47], electromagnetic radiation [61], and running times [46].

Recently a lot of effort has been put into constructing cryptographic primitives that come along with some form of leakage resilience, meaning that the scheme should remain secure even in case the adversary obtains some type of leakage on the secrets used within the system. A common way to model leakage attacks, is to empower the adversary with access to a leakage oracle, taking as input (adaptively chosen) functions f_i and returning $f_i(st)$ where st is the current secret state of the cryptosystem under attack. Clearly some restriction on the functions f_i has to be put, as otherwise there is no hope for security. By now, a plethora of leakage models (corresponding to different ways how to restrict the functions f_i) have been proposed. We review the ones that are more relevant to our work below (and refer the reader to Section 1.3 for a bigger picture).

- **Bounded leakage:** One natural restriction is to just assume that the total bit-length of the leakage obtained via the functions f_i is smaller than some a priori determined leakage bound ℓ. Usually the leakage bound ℓ is also related to the secret key size, so that a relatively large fraction of the secret key can be leaked. Leakage-resilient schemes in this model include storage schemes [22, 27], public-key and identity-based encryption [53, 2,

50, 1, 11, 24, 10, 15, 42], signature schemes [45, 2, 11, 24, 9, 51, 34, 54, 55, 18], and more (see, e.g., [37, 8, 57, 56]).

- **Noisy leakage:** A drawback of the bounded leakage model is that physical leakage rarely obeys to the restriction that the length of the leakage is a priori bounded (e.g., a power trace could be much longer than the secret key). A milder restriction (which weakens the above) is to just assume that the total amount of leakage does not reduce the entropy of the secret key by too much. Leakage-resilient primitives in this model include one-way relations, public-key encryption, and signature schemes [53, 24, 37].

The focus of this paper is on constructing leakage-resilient signatures in the bounded leakage and noisy leakage model, where one demands that a signature scheme remains unforgeable even against an adversary leaking arbitrary (yet restricted as above) information on the signing key and the overall randomness used within the life-time of the system (this flavour is sometimes known as *fully* leakage resilience). We give a more in-depth description of our contributions in Section 1.1; an overview of our techniques can be found in Section 1.2.

1.1 Our Contribution

Note that in order for a signature scheme to remain existentially unforgeable in the bounded leakage model, it must necessarily be the case that the length of a signature is larger than the length of the secret key (as otherwise an adversary could simply leak a forgery). A first consequence of this is that signatures are very long, as the goal is to enlarge the secret key to tolerate more and more leakage, which is impractical. A second consequence is that we cannot make any meaningful security statement (w.r.t. security in the bounded leakage model) for schemes where the size of the secret key is much larger than the size of a single signature. One remarkable such case is the setting of the Bounded-Retrieval Model [17, 29, 30] (BRM), where one intentionally inflates the size of the secret key while keeping constant the size of a signature and the verification key, as well as the computational complexity of the scheme (w.r.t. signature computation/verification).

A similar concern applies to the noisy leakage model, for those schemes where signatures (statistically) reveal little information about the secret key. In such cases leaking a forgery is, again, a valid leakage query, as a signature does not decrease the uncertainty of the secret key by too much. Still, we would like to not consider a scheme completely insecure if the adversary cannot do better than that.

A first step towards addressing the above issues, has recently been taken by Nielsen, Venturi and Zottarel [55] (for the bounded leakage model) who introduced a "graceful degradation" property, essentially requiring that an adversary should not be able to produce more forgeries than what he could have leaked via leakage queries. More precisely, in order to break unforgeability, an adversary has to produce n forgeries where $n \approx \lambda/(\gamma \cdot s) + 1$ for signatures of size s, a total of λ bits of leakage, and a "slack parameter" $\gamma \in (0, 1]$ measuring how close to optimal security a scheme is. The main advantage is that one can design schemes where the size of the secret key is independent of the signature size, leading to shorter signatures. Moreover, this flavour of leakage resilience still allows for interesting applications (e.g., to leaky identification [55]).

New definitions. We start by generalizing the above graceful degradation property to the setting of fully-leakage resilience (both in the bounded and noisy leakage model). Our main notion, dubbed fully-leakage one-more unforgeability, is essentially the same as the one of [55],

Scheme	Model			G. D.	Assumption	Efficiency		
	bounded	noisy	fully			leak	$\frac{\lvert vk\rvert}{\lvert sk\rvert}$	$\frac{\lvert \sigma\rvert}{\lvert sk\rvert}$
DHLW10[25]	✓	✗	✗	—	DLIN, SXDH	$1-o(1)$	1	9
BSW11[9]	✓	✗	✓	—	DLIN, SXDH	$1-o(1)$	$\omega(\log^2\kappa)\log^2 q_s$	$\omega(\log^2\kappa)\log^2 q_s$
MTVY11[51]	✓	✗	✓	—	SXDH	$1-o(1)$	$\omega(\log\kappa)$	$O(1)$
GJS11[37]*	✓	✓	✓	—	SXDH	$1-o(1)$	$\Omega\left(\frac{\kappa+\psi}{\psi}\right)$	$\Omega\left(\frac{\lvert C\rvert\kappa}{\psi}\right)$
NVZ14[55]	✓	✗	✗	$\gamma = O(1)$	DLIN	$\frac{1}{2}-o(1)$	$\frac{1}{2}$	$\Omega(1/\psi)$
$\mathcal{SS}_1^\dagger$ (§ 4)	✓	✗	✓	$\gamma = O(1/\kappa)$	DLIN	$1-\epsilon$	$(\kappa+12+\epsilon^{-1})/\psi$	$(24\epsilon^{-1}+90)/\psi$
$\mathcal{SS}_2^\dagger$ (§ 4)	✓	✗	✓	$\gamma = O(1/\kappa)$	DLIN	$1-\epsilon$	$(\kappa+30+\epsilon^{-1})/\psi$	$(15\epsilon^{-1}+133)/\psi$
$\mathcal{SS}_3$ (§ 5)	✓	✓	✓	$\gamma = O(1/q_s)$	DLIN	$1-\epsilon$	$(\kappa+12+\epsilon^{-1})/\psi$	$(15\log\kappa\epsilon^{-1}+13)/\psi$
ADW09[2]‡	✓	✗	✗	✗	BDH	$\frac{1}{2}-\epsilon$	$\Omega(\epsilon^{-1}/\psi)$	$\Omega(\epsilon^{-1}/\psi)$
$\mathcal{SS}_4^\ddagger$ (§ 6)	✓	✓	✓	$\gamma = O(1)$	BDH	$1-\epsilon$	$\Omega(\log\kappa\epsilon^{-1}/\psi)$	$\Omega(\log\kappa\epsilon^{-1}/\psi)$

Table 1: Comparison of known efficient leakage-resilient signature schemes in the bounded and noisy leakage model. Top: Standard model schemes. Bottom: Schemes in the BRM. When there are multiple instantiations we considered the most efficient one. The parameter ψ is the number of group elements in the secret key; κ is the security parameter, γ is the slack parameter, and q_s the number of signature queries done by the adversary. The † symbol means the scheme relies on memory erasures. The ‡ symbol means the scheme is in the random oracle model; [2] achieved the weaker notion of entropic unforgeability. G.D. stands for graceful degradation. For the scheme * we wrote $|C|$ for the size of the boolean circuit representing the NP-relation used within the scheme.

with the twist that leakage functions can be applied to the entire state of the signer (both for noisy and length-bounded leakage).

We also establish a "middle-ground" notion, which models a setting where secure erasures of the state are available. In particular, we imagine a situation in which the random coins sampled by the signer are completely erased after each invocation of the signing algorithm. Note that in this setting the leakage can essentially depend only on the secret key and the random coins used to compute a single signature. While requiring perfect erasure is a strong assumption (see, e.g., [13]), we believe our notion might still make sense for some applications, as it in particular allows to design simpler and more efficient schemes.

New generic constructions. Next, we present new constructions of fully leakage-resilient signature schemes based on generic cryptographic building blocks, improving over previous work in several directions. All of our schemes tolerate leakage on the entire state of the signer, up to a $1-o(1)$ fraction of the secret key length (in the bounded leakage model). They also offer graceful degradation, allowing to have short signatures of size independent of the size of the secret key.

The first three schemes are secure in the standard model (the first two assume perfect erasures, the third one not). Our fourth scheme is secure in the BRM, but requires a random oracle. Some of our schemes additionally maintain security in the noisy leakage model. We refer the reader to Table 1.1 for a more detailed overview and comparison with previous works.

Concrete instantiations. We finally explain how to instantiate all building blocks required for our generic schemes. All our instantiations are efficient, and rely on fairly standard assumptions. We refer the reader to Section 7 for the details.

1.2 Technical Overview

We proceed with a high level description of the rationale behind our signatures constructions. For simplicity, we focus here on the bounded leakage model and refer the reader to Section 8 for the extension to of our results to the setting of noisy leakage.

Schemes template. All of our schemes follow a common template, which is inspired by the construction of [55]. The main ingredients are: (1) A hybrid[1] linearly homomorphic non-interactive commitment scheme; (2) A statistical non-interactive witness indistinguishable (NIWI) argument system. The secret key consists of two random degree-d polynomials, $\delta(X)$ and $r(X)$, over a finite field $\mathbb{F}$. The verification key includes $t+1$ verification keys for the commitment scheme, which we denote by $(\vartheta, \{\vartheta_i\}_{i=1}^t)$, and commitments com_i to the coefficients δ_i of $\delta(X)$ computed using the coefficients r_i of $r(X)$ as randomness.

To sign a message m one computes $\tilde{m} = \delta(m)$, and commits to it in two different ways: (1) Using verification key ϑ and randomness $\tilde{r} = r(m)$ (call the result $c\tilde{o}m$); (2) Using another verification key $\bar{\vartheta}$ (depending on $\{\vartheta_i\}_{i=1}^t$ and on fresh randomness) and uniform randomness $\bar{r}$ (call the result $c\bar{o}m$).[2] A signature then consists of the commitment $c\bar{o}m$ together with a NIWI argument π that $(\tilde{m}, \tilde{r})$ and $(\tilde{m}, \bar{r})$ are valid openings of, respectively, $c\tilde{o}m$ and $c\bar{o}m$. In order to verify a signature, the verifier can compute the value $c\tilde{o}m$ non-interactively (using the fact that the commitment scheme is linearly homomorphic), and hence verify the argument π.

Proof outline. To prove security, we define a series of computationally indistinguishable hybrids experiments, where, in each experiment, we either modify the way keys are sampled, or the way signature/leakage queries are handled. The goal is to reach a final hybrid in which signature queries *on average* do not reveal information about the secret polynomial $\delta(X)$. In order to achieve this, we set-up the keys in such a way that the verification key ϑ is always equivocable, whereas the verification key $\bar{\vartheta}$ is binding with some probability p (and equivocable with probability $1-p$); let us call Bad the event that $\bar{\vartheta}$ is binding. In case Bad does not happen, we can simply commit to 0 and later equivocate the commitment to sample consistent randomness to simulate a signature/leakage query. In case Bad happens, we will be forced to ask an additional leakage query and reveal the value $\tilde{m} = \delta(m)$, which allows to answer a signature query correctly.

In the second part of the proof we analyse the uncertainty of $\delta(X)$ given the view of the adversary in the last hybrid, and we show two facts: (1) The conditional min-entropy of $\delta(X)$ is high with high probability; (2) There exists a predictor able to recover $\delta(X)$ with probability depending on the advantage of the adversary in the unforgeability game. Finally one observes that fact (2) contradicts fact (1) and, whenever the advantage is noticeable, the probability of fact (2) is high, which cannot be true, since also the probability of fact (1) is, and thus the scheme has to be secure.

This is the most critical part of the proof, as in particular, we cannot show directly that the conditional *average* min-entropy of the random variable $\delta(X)$ is high.[3] To get around this, we use a careful averaging argument and set the slack parameter γ appropriately.

Amplifying the leakage rate. Any scheme following the above template can tolerate leakage of at most a $1/2 - o(1)$ fraction of the secret key. Our actual constructions achieve optimal

[1] Intuitively this is a commitment scheme which can be either unconditionally binding or equivocable, depending on the distribution of the public parameters; moreover these two distributions of the parameters are computationally indistinguishable.

[2] The actual way $\bar{\vartheta}$ is computed is different in each of the schemes.

[3] In fact it can be arbitrarily small, due to the presence of the event Bad.

leakage rate $1 - o(1)$, by replacing the polynomial $\delta(X)$ with a matrix of polynomials $\mathbf{\Delta}$ and exploiting a commitment scheme with a message space of "higher dimension".

1.3 Related Work

On a high level, our scheme follows the template given in [45]; our techniques are mostly related to the ones in [9, 51]. Below, we first briefly describe the standard model scheme of [45], then we provide a more detailed comparison of our schemes and [9, 51]. Finally, we discuss other leakage models and constructions therein.

The signature scheme of Katz and Vaikuntanathan [45] relies on a second-preimage resistant (SPR) function, a CPA-secure public-key encryption scheme and an unbounded simulation-sound NIZK proof system. The secret key is a random pre-image x of a value y under the SPR function (y is part of the public key); a signature is an encryption c of x together with a proof that the ciphertext c is indeed an encryption of a pre-image of y (the message m is used as label in the proof). The scheme does not achieve fully-leakage resilience.

Comparison to [9] and [51]. The first fully leakage-resilient signature schemes in the standard model, were constructed independently by [9] and [51]. In a nutshell, the signature scheme of Boyle *et al.* [9] instantiates the above paradigm using a *lossy* tag-based public-key encryption scheme and an admissible hash function H [5]. A signature for a message m is an encryption c under the tag $H(m)$ of a pre-image for an SPR function, together with a statistical NIWI argument that the ciphertext c is indeed an encryption of a pre-image of y. The lossy public-key encryption scheme has the property that for many tags the corresponding ciphertexts are lossy (i.e., they do not leak information about the plaintext), while for a few other tags the corresponding ciphertexts are correctly decryptable. The admissible hash function allows to partition lossy tags from non-lossy tags. With noticeable probability all the signature queries will correspond to lossy tags, therefore signatures do not provide any information about the secret key *even if the randomness of the NIWI is leaked*, while the forgery corresponds to a non-lossy tag, and therefore the simulator can extract a pre-image of y that w.h.p. will be different from the one initially sampled.

The signature scheme of [51] can be seen as an alternative instantiation of the same partitioning strategy. The lossy public-key encryption is removed from the construction, by observing that Groth-Sahai NIZKs are either statistically witness indistinguishable or extractable, depending on the choice of the common reference string (CRS). The admissible function is replaced by the Waters hash function [66], that provides a way to map the message space to a CRS. The Waters function allows to show that with noticeable probability all signature queries correspond to a witness indistinguishable CRS while the forgery corresponds to an extractable CRS.

We observe that the approach of [51] cannot work in case of one-more unforgeability, due to the impossibility result of [43]; this is because finding two distinct messages that are mapped into two extractable CRS would imply that the discrete logarithm problem is easy in the group. Using a similar partitioning argument as the one in [9] might work, however the signature size would be proportional to the output of the hash function that grows as $\log^2 \kappa \cdot \log^2 q_s$, where q_s is the number of signature queries. Our technique, instead, allows for a more efficient construction.

Other leakage models. Leakage-resilient signatures are constructed also in other leakage models. The "only computation leaks" model [52], assumes that the only parts of the state subject to leakage are the ones involved in the computation. Leakage-resilient signatures in this model were constructed in [33]. One important limitation of the "only computation leaks" axiom, is that it does not captures important attacks in which inactive values in memory are still

vulnerable [41]. Note that our assumption of perfect erasures might be interpreted as a special case of "only-computation leaks", as a deleted value is never used in a computation again. The model of hard-to-invert leakage [26, 23] (a.k.a. auxiliary input model) generalizes both the bounded and noisy leakage model by allowing the adversary to see computationally hard-to-invert functions of the secret key. The only signature schemes secure in this model are the ones from [32]. In the continual leakage model [25, 11, 44, 37] one can additionally refresh the secret key without changing the verification key of the signature scheme, while the scheme tolerates arbitrary (bounded or noisy) leakage on the entire state between two consecutive invocations of the refreshing algorithms.

A different, but related, line of research recently proposed leakage models that better cope with the perspective of cryptographic engineering (see, e.g., [64, 65, 49]).

Conference version. An abridged version of this paper appeared as [31], which only contains a high-level description of the signature scheme with security against noisy leakage in the non-erasure model (i.e., the scheme $\mathcal{SS}_3$ from this paper). This work is the full extended version of that paper, containing new material (i.e., the schemes $\mathcal{SS}_1$, $\mathcal{SS}_2$, and $\mathcal{SS}_4$) and significantly revised proofs.

1.4 Roadmap

We begin with some notation and basic definitions in Section 2. Our definitions for fully-leakage one-more unforgeability in the bounded leakage model can be found in Section 3. Our generic constructions are given in Section 4 (for the erasure case), Section 5 (for the non-erasure case), and Section 6 (for the BRM). Section 7 discusses how one can instantiate our constructions under standard number-theoretic assumptions, and comments on efficiency. In Section 8 we explain how to leverage our models and constructions to the noisy leakage model. Finally, we state a few open problems and interesting directions for future research in Section 9.

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 $\mathsf{poly}(|x|)$ steps.

Throughout the paper we let κ denote the security parameter. We say that a function $\nu : \mathbb{N} \to [0, 1]$ is negligible in the security parameter κ if $\nu(\kappa) = \kappa^{-\omega(1)}$. For a function $f(\kappa)$ we sometimes write $f(\kappa) \in \mathsf{negl}(\kappa)$ to denote that there exists a negligible function $\nu : \mathbb{N} \to [0, 1]$ such that $f(\kappa) \leq \nu(\kappa)$. For two ensembles $\mathcal{X} = \{X_\kappa\}_{\kappa \in \mathbb{N}}$ and $\mathcal{Y} = \{Y_\kappa\}_{\kappa \in \mathbb{N}}$, we write $\mathcal{X} \equiv \mathcal{Y}$ to denote that $\mathcal{X}$ and $\mathcal{Y}$ are identically distributed, and $\mathcal{X} \approx_s \mathcal{Y}$ (resp., $\mathcal{X} \approx_c \mathcal{Y}$) to denote that $\mathcal{X}$ and $\mathcal{Y}$ are statistically (resp., computationally) indistinguishable.

Vectors and matrices are typeset in boldface. For a vector $\mathbf{v} = (v_1, \ldots, v_n)$ we sometimes write $\mathbf{v}[i]$ for the i-th element of $\mathbf{v}$.

2.2 Random Variables and Min-Entropy

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

We state a lemma from [45]:

Lemma 2.1. *Let X be a random variable with $H := \mathbb{H}_\infty(X)$, and fix $\Delta \in [0, H]$. Let f be a function whose range has size 2^λ, and set $\mathcal{Y} := \left\{y \in \{0,1\}^\lambda |\ \mathbb{H}_\infty(X|f(X) = y) \leqslant H - \Delta\right\}$. Then, we have that $\mathbb{P}[f(X) \in \mathcal{Y}] \leqslant 2^{\lambda-\Delta}$.*

2.3 Commitment Schemes

A (non-interactive) commitment scheme $\mathcal{COM}$ is a tuple of algorithms $(\mathsf{Setup}, \mathsf{Com})$, defined as follows: (1) Algorithm Setup takes as input the security parameter and outputs a verification key ϑ; (2) Algorithm Com takes as input a message $m \in \mathcal{M}$, randomness $r \in \mathcal{R}$, the verification key ϑ 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 $com = \mathsf{Com}(\vartheta, m; r)$.

A commitment scheme has two properties, known as binding and hiding.

Binding Property. Consider the following probability:

$$\mathbb{P}[\mathsf{Com}(\vartheta, m_0; r_0) = \mathsf{Com}(\vartheta, m_1; r_1) :\ \vartheta \leftarrow\!\!\$\ \mathsf{Setup}(1^\kappa); ((m_0, r_0), (m_1, r_1)) \leftarrow\!\!\$\ \mathsf{A}(\vartheta)].$$

A commitment scheme is *computationally* binding in case the above is negligible for all PPT adversaries A. In case the probability is zero, for all even unbounded A, the commitment scheme is called *perfectly* binding.

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

$$\{\vartheta, \mathsf{Com}(\vartheta, m_0)\}_{\kappa\in\mathbb{N}} \approx \{\vartheta, \mathsf{Com}(\vartheta, m_1)\}_{\kappa\in\mathbb{N}},$$

where the two ensembles are considered as random variables over the choice of the randomness to generate $\vartheta \leftarrow\!\!\$\ \mathsf{Setup}(1^\kappa)$ and to compute the commitment. A commitment scheme is *statistically* (resp., *computationally*) hiding in case $\approx$ refers to statistical (resp., computational) indistinguishability. In case the two ensembles are identically distributed, the commitment scheme is called *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 $com^* = \mathsf{Com}(\vartheta, m + m'; r + r')$ and $com'' = \mathsf{Com}(\vartheta, c \cdot m; c \cdot r)$. The above can be generalized to abstract operations over the spaces $\mathcal{M}$, $\mathcal{R}$ and $\mathcal{C}$, but for simplicity, and to be consistent with the concrete instantiation given in this paper, we stick to the above formulation here.

Hybrid commitments. A *hybrid* [14] commitment scheme $\mathcal{COM} = (\mathsf{Setup}, \mathsf{Com}, \mathsf{ESetup}, \mathsf{ECom}, \mathsf{Equiv})$ is a tuple of algorithms specified as follows: (1) $(\mathsf{Setup}, \mathsf{Com})$ is a non-interactive commitment scheme with message space $\mathcal{M}$, randomness space $\mathcal{R}$, and commitment space $\mathcal{C}$; (2) Algorithm ESetup takes as input the security parameter and outputs a pair $(\vartheta, \tau) \leftarrow\!\!\$\ \mathsf{ESetup}(1^\kappa)$; (3) Algorithm ECom takes as input a pair (ϑ, τ) and outputs a pair $(com, r') \leftarrow\!\!\$\ \mathsf{ECom}(\vartheta, \tau)$; (4) Algorithm Equiv takes as input (τ, m, r') and outputs $r \leftarrow\!\!\$\ \mathsf{Equiv}(\tau, m, r')$.

Definition 2.1. *We say that* $\mathcal{COM} = (\mathsf{Setup}, \mathsf{Com}, \mathsf{ESetup}, \mathsf{ECom}, \mathsf{Equiv})$ *is a* hybrid *commitment scheme if the following holds.*

Perfectly Binding. *The tuple* $(\mathsf{Setup}, \mathsf{Com})$ *is a* perfectly binding *commitment scheme.*

Trapdoor Hiding. *The tuple* $(\mathsf{ESetup}, \mathsf{Com}, \mathsf{ECom}, \mathsf{Equiv})$ *is a* trapdoor *commitment scheme, namely for all* $(\vartheta, \tau) \leftarrow\!\!\$\ \mathsf{ESetup}(1^\kappa)$ *and for all* $m \in \mathcal{M}$ *the following probability distributions are indistinguishable:*

$$\left\{ (com, r) \;\middle|\; \begin{array}{c} r \leftarrow\!\!\$\ \mathcal{R}, \\ com := \mathsf{Com}(\vartheta, m; r) \end{array} \right\} \text{ and } \left\{ (com, r) \;\middle|\; \begin{array}{c} (com, r') \leftarrow\!\!\$\ \mathsf{ECom}(\vartheta, \tau), \\ r := \mathsf{Equiv}(\tau, m, r') \end{array} \right\}.$$

Hybridness. *The following probability distributions are computationally indistinguishable:*

$$\{\vartheta |\ (\vartheta, \tau) \leftarrow\!\!\$\ \mathsf{ESetup}(1^\kappa)\} \text{ and } \{\vartheta |\ \vartheta \leftarrow\!\!\$\ \mathsf{Setup}(1^\kappa)\}.$$

We call a verification key *equivocable* (resp. *binding*) if it is generated by the algorithm ESetup (resp. Setup). In the same way a commitment is *equivocable* (resp. *binding*) if it is generated using an equivocable (resp. binding) verification key.

For a hybrid linearly homomorphic commitment scheme we will require the following additional property. Let $(\vartheta, \tau) \leftarrow\!\!\$\ \mathsf{ESetup}(1^\kappa)$, $(com_1, r_1') \leftarrow\!\!\$\ \mathsf{ECom}(\vartheta, \tau)$ and $(com_2, r_2') \leftarrow\!\!\$\ \mathsf{ECom}(\vartheta, \tau)$. Then we can use randomness $r_1' + r_2'$ to equivocate $com_1 \cdot com_2$, and randomness $c \cdot r_1'$ to equivocate com_1^c (for any c).

Consider the following experiment $\mathbf{Exp}^{\text{hiding}}_{\mathcal{COM},\mathsf{A}}(\kappa)$ running with a PPT adversary A and parametrized by the security parameter $\kappa \in \mathbb{N}$:

1. $(m_0, m_1) \leftarrow\!\!\$\ \mathsf{A}(\vartheta)$ where $\vartheta \leftarrow\!\!\$\ \mathsf{Setup}(1^\kappa)$;
2. $com \leftarrow\!\!\$\ \mathsf{Com}(\vartheta, m_b)$ where $b \leftarrow\!\!\$\ \{0,1\}$;
3. Let $b' \leftarrow\!\!\$\ \mathsf{A}(\vartheta, com)$, output 1 iff $b = b'$.

Lemma 2.2 (Adaptive Computational Hiding)**.** *Let* $\mathcal{COM}$ *be a hybrid commitment scheme. For any PPT* A *we have that:*

$$\left| \mathbb{P}\left[\mathbf{Exp}^{\text{hiding}}_{\mathcal{COM},\mathsf{A}}(\kappa) = 1 \right] - 1/2 \right| \in \mathsf{negl}(\kappa).$$

Proof. The trapdoor hiding property implies that for any $(\vartheta, \tau) \leftarrow\!\!\$\ \mathsf{ESetup}(1^\kappa)$ and any $(m_0, m_1) \in \mathcal{M}$ the following two probability distributions are indistinguishable:

$$\{com \mid com \leftarrow\!\!\$\ \mathsf{Com}(\vartheta, m_0)\} \text{ and } \{com \mid com \leftarrow\!\!\$\ \mathsf{Com}(\vartheta, m_1)\}. \tag{1}$$

Since Eq. (1) holds for any ϑ and any m_0, m_1 we can allow the messages to be chosen adaptively after seeing the verification key. Hence, for any $(\vartheta, \tau) \leftarrow\!\!\$\ \mathsf{ESetup}(1^\kappa)$, and for any PPT A, there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ such that:

$$\left| \mathbb{P}\left[b = b' \;\middle|\; \begin{array}{c} (m_0, m_1) \leftarrow\!\!\$\ \mathsf{A}(\vartheta);\ b \leftarrow\!\!\$\ \{0,1\}; \\ com \leftarrow\!\!\$\ \mathsf{Com}(\vartheta, m_b);\ b' \leftarrow\!\!\$\ \mathsf{A}(\vartheta, com); \end{array} \right] - 1/2 \right| \leqslant \nu(\kappa). \tag{2}$$

The hybridness property and Eq. (2) imply the lemma. □

2.4 NIWI and NIZK Arguments

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 $(\mathsf{Init}, \mathsf{Prove}, \mathsf{Ver})$ for $\mathfrak{R}$ is a tuple of algorithms specified as follows: (1) Init takes as input the security parameter and outputs a common reference string $\mathsf{crs} \leftarrow\!\!\$\ \mathsf{Init}(1^\kappa)$; (2) Prove takes as input a pair $(x,w) \in \mathfrak{R}$, and outputs a proof

$\pi \leftarrow\!\!\$\ \mathsf{Prove}(\mathsf{crs}, x, w)$; (3) Ver takes as input a statement x and a proof π, and outputs a bit $b \leftarrow\!\!\$\ \mathsf{Ver}(\mathsf{crs}, x, \pi)$. We require that for all pairs $(x, w) \in \mathfrak{R}$, and all $\mathsf{crs} \leftarrow\!\!\$\ \mathsf{Init}(1^\kappa)$, the following holds: $\mathbb{P}[\mathsf{Ver}(\mathsf{crs}, x, \mathsf{Prove}(\mathsf{crs}, x, w)) = 0] \in \mathsf{negl}(\kappa)$ (over the randomness of the prover and verifier algorithms).

We review the notions of non-interactive witness indistinguishable and zero-knowledge argument systems for a relation $\mathfrak{R}$.

NIWI arguments. A non-interactive witness indistinguishable argument system satisfies two properties known as adaptive soundness and statistical witness indistinguishability.

Definition 2.2 (Adaptive soundness)**.** *Let $\mathcal{NIWI}$ be a non-interactive argument system for a language $\mathfrak{L}$. We say that $\mathcal{NIWI}$ satisfies adaptive soundness, if for all PPT adversaries A we have*

$$\mathbb{P}\left[\mathsf{Ver}(\mathsf{crs}, x, \pi) = 1 \wedge x \notin \mathfrak{L} : \ \mathsf{crs} \leftarrow\!\!\$\ \mathsf{Init}(1^\kappa), (x, \pi) \leftarrow\!\!\$\ \mathsf{A}(1^\kappa, \mathsf{crs})\right] \in \mathsf{negl}(\kappa).$$

Definition 2.3 (Statistical witness indistinguishability)**.** *Let $\mathcal{NIWI}$ be a non-interactive argument system for a relation $\mathfrak{R}$. We say that $\mathcal{NIWI}$ satisfies statistical witness indistinguishability if for any triplet (x, w, w') such that $(x, w) \in \mathfrak{R}$ and $(x, w') \in \mathfrak{R}$, the distributions $\{(\mathsf{crs}, \pi) | \mathsf{crs} \leftarrow\!\!\$\ \mathsf{Init}(1^\kappa), \pi \leftarrow\!\!\$\ \mathsf{Prove}(\mathsf{crs}, x, w)\}$ and $\{(\mathsf{crs}, \pi) |\ \mathsf{crs} \leftarrow\!\!\$\ \mathsf{Init}(1^\kappa), \pi \leftarrow\!\!\$\ \mathsf{Prove}(\mathsf{crs}, x, w')\}$ are statistically indistinguishable.*

NIZK arguments. We will need a non-interactive zero-knowledge argument system satisfying two properties, known as adaptive multi-theorem zero-knowledge and true-simulation extractability [25].

Definition 2.4 (Adaptive multi-theorem zero-knowledge)**.** *Let $\mathcal{NIZK}$ be a non-interactive argument system for a relation $\mathfrak{R}$. We say that $\mathcal{NIZK}$ satisfies adaptive multi-theorem zero-knowledge with labels if the following holds:*

(i) There exists an algorithm $\overline{\mathsf{Init}}$ that outputs a CRS crs, and a simulation trapdoor τ_{sim}.

(ii) There exists a PPT simulator Sim such that, for all PPT adversaries A, we have that

$$\left|\mathbb{P}\left[\mathsf{A}^{\mathsf{Prove}(\mathsf{crs},\cdot)}(\mathsf{crs}) = 1 |\ \mathsf{crs} \leftarrow\!\!\$\ \mathsf{Init}(1^\kappa)\right] - \mathbb{P}\left[\mathsf{A}^{\mathcal{SIM}(\tau_{sim},\cdot)}(\mathsf{crs}) = 1 |\ (\mathsf{crs}, \tau_{sim}) \leftarrow\!\!\$\ \overline{\mathsf{Init}}(1^\kappa)\right]\right|$$

is negligible in κ. The simulation oracle $\mathcal{SIM}(\tau_{sim}, \cdot)$ takes as input a pair (x, w) and a label L, checks if $(x, w) \in \mathfrak{R}$, and, if true, ignores w and outputs a simulated argument $\mathsf{Sim}(\tau_{sim}, (x, L))$, and otherwise outputs $\perp$.

Definition 2.5 (True-simulation extractability)**.** *Let $\mathcal{NIZK}$ be a non-interactive argument system for a relation $\mathfrak{R}$. We say that $\mathcal{NIZK}$ is* true-simulation extractable *(tSE) with labels if the following holds:*

(i) There exists an algorithm $\overline{\mathsf{Init}}$ that outputs a CRS crs, a simulation trapdoor τ_{sim}, and an extraction trapdoor τ_{ext}.

(ii) There exists a PPT algorithm $\mathsf{K}(\tau_{ext}, (x, L), \pi)$ such that, for all PPT adversaries A, we have $\mathbb{P}[\mathsf{A}$ wins$] \in \mathsf{negl}(\kappa)$ in the following game:

- *The challenger runs $(\mathsf{crs}, \tau_{sim}, \tau_{ext}) \leftarrow\!\!\$\ \overline{\mathsf{Init}}(1^\kappa)$, and gives crs to A.*
- *A is given access to the simulation oracle $\mathcal{SIM}$, which it can query adaptively.*
- *A outputs a tuple $((x^*, L^*), \pi^*)$.*
- *The challenger runs $w \leftarrow\!\!\$\ \mathsf{K}(\tau_{ext}, (x^*, L^*), \pi)$. We say that A wins if: (a) the pair (x^*, L^*) was not part of a query to the simulation oracle; (b) $\mathsf{Ver}(\mathsf{crs}, (x^*, L^*), \pi^*) = 1$; and (c) $(x^*, w) \notin \mathfrak{R}$.*

Oblivious sampling. For our application we will need the additional property that the Init algorithm supports oblivious sampling (a.k.a. reference string uniformity in [62]), meaning that the distribution of the common reference string is statistically-close to the uniform distribution. Such property will be satisfied in our concrete instantiations based on the Groth-Sahai proof system [39] (see Section 7).

3 Fully-Leakage 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 a bit $\mathsf{Verify}(vk, (m, \sigma)) \in \{0, 1\}$. We say that $\mathcal{SS}$ satisfies correctness if for all messages $m \in \mathcal{M}$ and for all pairs of keys (vk, sk) generated via KGen, we have that $\mathsf{Verify}(vk, (m, \mathsf{Sign}(sk, m)))$ returns 1 with overwhelming probability over the randomness of the signing algorithm.

In this section we extend the notion of one-more unforgeability [55] to a setting where all intermediate values generated within the lifetime of the system (and not just the secret key) are subject to leakage. As outlined in the introduction, we consider two cases depending whether one is willing to assume perfect erasures or not.

3.1 Definition without Erasures

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

1. Sample $r_0 \in \{0,1\}^*$, run the key generation algorithm to obtain a pair $(vk, sk) := \mathsf{KGen}(1^\kappa; r_0)$, and return vk to A; let $st = \{r_0\}$.
2. The adversary A can adaptively issue signing queries. Upon input a message $m \in \mathcal{M}$, the adversary is given a signature $\sigma := \mathsf{Sign}(sk, m; r)$ computed using fresh coins r. The state is updated to $st := st \cup \{r\}$.
3. The adversary A can adaptively issue leakage queries. Upon input an arbitrary function f, the adversary is given $f(st)$ where st is the current state.
4. Let $\mathcal{Q}$ be the set of signing queries issued by A, and let $\Lambda \in \{0,1\}^\lambda$ be the total amount of information leaked by the adversary. A outputs n pairs $(m_1^*, \sigma_1^*), \ldots, (m_n^*, \sigma_n^*)$.
5. The experiment outputs 1 if and only 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) $n \geq \lfloor \lambda/(\gamma \cdot s) \rfloor + 1$, where $s := |\sigma|$ is the size of a signature.
   (d) $\lambda \leq \ell$ and $|\mathcal{Q}| \leq q_s$.

Definition 3.1 (Fully-leakage one-more unforgeability, no erasures)**.** *We say that $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ is $(\ell, q_s, \gamma, \varepsilon)$-*fully-leakage one-more unforgeable *if for every PPT adversary A asking q_s signature queries we have that $\mathbb{P}[\mathbf{Exp}^{\text{one-more}}_{\mathcal{SS},\mathsf{A}}(\kappa, \ell, q_s, \gamma) = 1] \leq \varepsilon$. Whenever $\varepsilon \in \mathsf{negl}(\kappa)$, $q_s \in \mathsf{poly}(\kappa)$, and γ does not depend on q_s, we simply say that $\mathcal{SS}$ is (ℓ, γ)-fully-leakage one-more unforgeable.*

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)-fully-leakage one-more unforgeability implies standard unforgeability for any adversary asking no leakage ($\lambda = 0$).

As pointed out in [55], the slack 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).

For some of our schemes, the slack parameter will be dependent on the security parameter, and in some case even on the number of signature queries asked by the adversary. We stress, however, that as long as this dependency is polynomial, one-more unforgeability is still meaningful for some applications (cf. [55, Section 5]).

3.2 Definition with Erasures

In the setting of Definition 3.1 the leakage is allowed to depend on the entire randomness produced within the lifetime of the system. A more restricted (yet natural) setting is to assume that the state is erased after each signature is generated. Although perfect erasures might be hard to implement [13], we believe that such weakening might still be useful in practice. Moreover, looking ahead, the ideas behind the schemes we construct assuming perfect erasures (cf. Section 4) form the core for the scheme which does not consider erasures (cf. Section 5).

To define fully-leakage one-more unforgeability with erasures one simply modifies the experiment $\mathbf{Exp}^{\text{one-more}}_{\mathcal{SS},\mathsf{A}}(\kappa, \ell, q_s, \gamma)$ by requiring that the state variable st is always equal to sk. Since this is clearly equivalent to a setting where the adversary asks up to one leakage query after each signature query, we can equivalently[4] give A access to the following machine $\widetilde{\mathsf{Sign}}(sk, \cdot)$:

- Upon input a value $m \in \mathcal{M}$, compute $\sigma \leftarrow\!\!\$\ \mathsf{Sign}(sk, m)$ and return σ. We call such invocation of the signature oracle a *leak-free query*.
- Upon input a pair (m, f), where $m \in \mathcal{M}$ and f is an arbitrary polynomial-time computable function described as a circuit, sample randomness $r \leftarrow\!\!\$\ \{0,1\}^*$, compute $\sigma := \mathsf{Sign}(sk, m; r)$ and return $(\sigma, f(sk, r))$. We call such invocation of the signature oracle a *leaky query*.

A formal definition follows. Given a signature scheme $\mathcal{SS}$, consider the following experiment $\widetilde{\mathbf{Exp}}^{\text{one-more}}_{\mathcal{SS},\mathsf{A}}(\kappa, \ell, q_s, \gamma)$ running with a PPT adversary A and parametrized by the security parameter $\kappa \in \mathbb{N}$, the leakage parameters $\ell \in \mathbb{N}$, and the slack parameter $\gamma := \gamma(\kappa)$:

1. The adversary can issue an initial leakage query. Upon input an arbitrary polynomial-time computable function f_0, sample $r_0 \in \{0,1\}^*$, run the key generation algorithm to obtain a pair $(vk, sk) := \mathsf{KGen}(1^\kappa; r_0)$, and return $(vk, f_0(r_0))$ to A.
2. The adversary A can adaptively access oracle $\widetilde{\mathsf{Sign}}(sk, \cdot)$.
3. Let $\mathcal{Q}$ be the set of signing queries issued by A, and denote with $\Lambda \in \{0,1\}^\lambda$ the total amount of information leaked by the adversary, including the initial leakage $\Lambda_0 \in \{0,1\}^{\lambda_0}$ (with $\lambda \geq \lambda_0$). A outputs n pairs $(m_1^*, \sigma_1^*), \ldots, (m_n^*, \sigma_n^*)$.
4. The experiment outputs 1 if and only if the following conditions are satisfied:

[4] It is, in fact, easy to see that an adversary that makes multiple queries between two consecutive signature queries can be reduced to an adversary in our model.

(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) $n \geq \lfloor \lambda/(\gamma \cdot s) \rfloor + 1$, where $s := |\sigma|$ is the size of a signature.

(d) $\lambda \leq \ell$, and $|\mathcal{Q}| \leq q_s$.

Definition 3.2 (Fully-leakage one-more unforgeability, erasures)**.** *We say that* $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ *is* $(\ell, q_s, \gamma, \varepsilon)$*-fully-leakage one-more unforgeable* with perfect erasures *if for every PPT adversary* A *we have that* $\mathbb{P}[\widetilde{\mathbf{Exp}}^{\text{one-more}}_{\mathcal{SS},\mathsf{A}}(\kappa, \ell, q_s, \gamma) = 1] \leq \varepsilon$*. Whenever* $\varepsilon \in \mathsf{negl}(\kappa)$*,* $q_s \in \mathsf{poly}(\kappa)$*, and* γ *does not depend on* q_s*, we simply say that* $\mathcal{SS}$ *is* (ℓ, γ)*-fully-leakage one-more unforgeable with perfect erasures.*

4 A Scheme with Erasures

4.1 The Basic Construction

We give a construction of a fully-leakage one-more unforgeable signature scheme (cf. Definition 3.2) based on the following building blocks:

- A statistical non-interactive witness-indistinguishable argument system $\mathcal{NIWI} = (\mathsf{Init}, \mathsf{Prove}, \mathsf{Ver})$ (cf. Section 2.4).
- A trapdoor hiding, linearly homomorphic commitment $\mathcal{COM}_1 = (\mathsf{ESetup}_1, \mathsf{Com}_1, \mathsf{ECom}_1, \mathsf{Equiv}_1)$ equipped with an obliviously sampleable key generation algorithm Setup and with message space equal to $\mathbb{F}^\mu$ (for a finite field $\mathbb{F}$ and a parameter $\mu \in \mathbb{N}$) and randomness space equal to $\mathbb{F}$ (cf. Section 2.3). Let ρ be defined as $\rho := \frac{\log |\mathbb{F}^\mu|}{\log(|\mathbb{F}^\mu| \times |\mathbb{F}|)} = \frac{\mu}{\mu+1}$.
- A hybrid, linearly homomorphic commitment scheme $\mathcal{COM}_2 = (\mathsf{Setup}_2, \mathsf{Com}_2, \mathsf{ESetup}_2, \mathsf{ECom}_2, \mathsf{Equiv}_2)$, with message space equal to $\mathbb{F}^\mu$ and randomness space equal to $\mathbb{F}^\nu$ for a parameter $\nu \geqslant \mu$.
- A pseudorandom generator (PRG) $G : \mathcal{X} \to \mathcal{Y}$.[5]

Our scheme $\mathcal{SS}_1 = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ has message space[6] equal to $\mathbb{F}$and is described below:

Key Generation. Let $t, d, \mu, \nu \in \mathbb{N}$ be parameters. Run $\mathsf{crs} \leftarrow\!\!\$\ \mathsf{Init}(1^\kappa)$, sample a uniform $y \in \mathcal{Y}$ from the range of G, run $\vartheta \leftarrow\!\!\$\ \mathsf{Setup}_1(1^\kappa)$, and $\vartheta_i \leftarrow\!\!\$\ \mathsf{Setup}_2(1^\kappa)$ for all $i \in [t]$. Sample $\boldsymbol{\Delta} \leftarrow\!\!\$\ (\mathbb{F}^\mu)^{d+1}$ and $\mathbf{r} = (r_0, \ldots, r_d) \leftarrow\!\!\$\ \mathbb{F}^{d+1}$, and compute commitments $com_i = \mathsf{Com}_1(\vartheta, \boldsymbol{\delta}_i; r_i)$ for $i \in [0, d]$, where $\boldsymbol{\delta}_i \in \mathbb{F}^\mu$ is the i-th column of $\boldsymbol{\Delta}$. Output

$$sk = (\boldsymbol{\Delta}, \mathbf{r}) \qquad vk = (\mathsf{crs}, y, \vartheta, \{\vartheta_i\}_{i=1}^t, \{com_i\}_{i=0}^d).$$

Signature. For $j \in [\mu]$, let $\delta_j(X)$ be the degree-d polynomial having as coefficients the elements in the j-th row of $\boldsymbol{\Delta}$, i.e., $\delta_j(X) := \sum_{i=0}^d \delta_{j,i} \cdot X^i$; similarly let $r(X)$ be the degree-d polynomial having as coefficients the elements of $\mathbf{r}$, i.e., $r(X) := \sum_{i=0}^d r_i \cdot X^i$. Define

$$\boldsymbol{\Delta}(X) := \begin{pmatrix} \delta_1(X) \\ \vdots \\ \delta_\mu(X) \end{pmatrix}.$$

[5] A PRG is a deterministic, efficiently computable, function such that the ensemble $\{G(x) |\ x \in \{0,1\}^\kappa\}_{\kappa \in \mathbb{N}}$ is pseudorandom.

[6] To sign arbitrary messages in $\mathcal{M} = \{0,1\}^*$, one can simply hash the message via a collision-resistant hash function with range $\mathbb{F}$.

Consider the following polynomial-time relation:

$$\mathfrak{R} := \left\{ (\vartheta, \vartheta', \tilde{com}, com); (\tilde{m}, \tilde{r}, r) \;\middle|\; \begin{array}{l} \tilde{com} = \mathsf{Com}_1(\vartheta, \tilde{m}; \tilde{r}) \\ com = \mathsf{Com}_2(\vartheta', \tilde{m}; r) \end{array} \right\} .$$

together with the polynomial-time OR-relation

$$\mathfrak{R}_{OR} := \{(\alpha, y);\ (\beta, x) \,|(\alpha, \beta) \in \mathfrak{R} \ \vee\ G(x) = y\} .$$

To sign a message $m \in \mathbb{F}$ compute $\tilde{m} = \boldsymbol{\Delta}(m)$ and $\tilde{r} = r(m)$, and let $\tilde{com} := \mathsf{Com}_1(\vartheta, \tilde{m}; \tilde{r})$. Notice 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. Pick a random index $j \leftarrow\!\!\$\, [t]$, and compute $com := \mathsf{Com}_2(\vartheta_j, \tilde{m}; r)$ where $r \leftarrow\!\!\$\, \mathbb{F}^{\nu}$. Using crs as common reference string, generate a NIWI argument π for $(\vartheta, \vartheta_j, \tilde{com}, com, y) \in \mathfrak{L}_{\mathfrak{R}_{OR}}$, the language generated by the above relation $\mathfrak{R}_{OR}$. Output $\sigma = (\pi, j, com)$.

Verification. Given a pair (m, σ), parse σ as (π, j, com). Output the same as:

$$\mathsf{Ver}(\mathsf{crs}, \pi, (\vartheta, \vartheta_j, \prod_{i=0}^{d} (com_i)^{m^i}, com, y)).$$

Let us first argue that the signature scheme satisfies the correctness property. This follows from the correctness of the NIWI argument system, and from the fact that $\mathcal{COM}$ is linearly homomorphic (cf. Section 2.3), as

$$\tilde{com} = \prod_{i=0}^{d} (com_i)^{m^i} = \prod_{i=0}^{d} \mathsf{Com}_1(\vartheta, \boldsymbol{\delta}_i \cdot m^i; r_i \cdot m^i) = \mathsf{Com}_1\big(\vartheta, \underbrace{\sum_{i=0}^{d} \boldsymbol{\delta}_i \cdot m^i}_{\tilde{m}}; \underbrace{\sum_{i=0}^{d} r_i \cdot m^i}_{\tilde{r}}\big).$$

Theorem 4.1. *Let $\mu \in \mathbb{N}$, and let $\mathbb{F}$ be a finite field of size $\log|\mathbb{F}| = \kappa$ for security parameter $\kappa \in \mathbb{N}$. For any $0 < \xi \leqslant \frac{\mu}{\mu+1}$ and $t = O(\kappa)$, whenever $\gamma = O(1/\kappa)$ the above signature scheme is $((\frac{\mu}{\mu+1} - \xi)|sk|, \gamma)$-fully-leakage one-more unforgeable with perfect erasures.*

Notice that the slack parameter γ is not optimal, as $\gamma = O(1/\kappa)$. However, as pointed out in Section 3, this is fine for some applications. The proof of Theorem 4.1 proceeds in two steps. Let A be a PPT adversary breaking fully-leakage one-more unforgeability with probability $\varepsilon(\kappa)$. In the first step we define a series of hybrid experiments that are indistinguishable from the original experiment defining security of the signature scheme. The purpose of these hybrids is to reach a state in which signature queries reveal (information theoretically) as little information as possible on the secret matrix $\boldsymbol{\Delta}$. In the second step we analyse the uncertainty of $\boldsymbol{\Delta}$ given the adversary's view; in particular we show two facts: (1) The conditional min-entropy of $\boldsymbol{\Delta}$ is high with high probability; (2) There exists a predictor able to recover $\boldsymbol{\Delta}$ with probability depending on $\varepsilon(\kappa)$. Finally one observes that fact (2) contradicts fact (1) and, whenever the advantage is noticeable, the probability of fact (2) is high, which cannot be true, since also the probability of fact (1) is, and thus the scheme has to be secure.

The main technical challenge to overcome is that although simulated signatures should not leak information about the secret key, the forgeries output by the adversary need to. It easy to see that these two requirements are in contrast. To overcome this difficulty we construct the simulator in a way that, on average, simulated signatures do not provide information about $\boldsymbol{\Delta}$, meaning that sometimes the simulator will have to retrieve and reveal some information about the secret key. We obtain a contradiction as soon as the forgeries reveal more information about the secret key than the information that the adversary has leaked. Setting the slack parameter to $O(1/\kappa)$ provides us with enough signatures to overcome the bound set by the leakage parameter and reach a contradiction.

4.2 Proof of Theorem 4.1

Let A be an adversary making $q = \mathsf{poly}(\kappa)$ *leaky* signature queries, such that

$$\mathbb{P}[\widetilde{\mathbf{Exp}}^{\text{one-more}}_{\mathcal{SS}_1,\mathsf{A}}(\kappa, \ell, q_s, \gamma) = 1] = \varepsilon(\kappa).$$

Experiment $\mathcal{H}_0$: This experiment is exactly the same as the original experiment defining fully-leakage one-more unforgeability of $\mathcal{SS}_1$. Note that the initial state (after key generation), is equal to $st := ((\mathbf{\Delta}, \mathbf{r}), \mathsf{crs}, \vartheta, \{\vartheta_i\}_{i=1}^t)$. (Recall that we assume oblivious sampling for both the NIWI and the commitment scheme.) This state is used to answer the initial leakage query f_0; afterwards everything but $(\mathbf{\Delta}, \mathbf{r})$ is erased. Additionally at each invocation of oracle $\widetilde{\mathsf{Sign}}(sk, \cdot)$ corresponding to a *leaky query* (m, f), the experiment produces a signature $\sigma = (\pi, j, com)$ following the signing algorithm, and defines the current state to be $st := ((\mathbf{\Delta}, \mathbf{r}), r, r_{niwi})$, where r is the randomness used to generate the commitment com and r_{niwi} is the randomness used to generate the argument π.

Denote with $((m_1^*, (\pi_1^*, j_1^*, com_1^*)), \ldots, (m_n^*, (\pi_n^*, j_n^*, com_n^*)))$ the set of forgeries returned by the adversary at the end of the experiment. Let Forge_0 be the event that $\mathcal{H}_0$ returns 1, so that $\mathbb{P}[\mathsf{Forge}_0] = \varepsilon$. Define False_0 to be the event that at least one of the proofs contained in the adversary's forgeries is relative to a false statement, i.e., False_0 is verified if in $\mathcal{H}_0$ there exists some index $i \in [n]$ for which com_i^* is a well-formed commitment to a value $m_i' \neq \mathbf{\Delta}(m_i^*)$. Let $\varepsilon_0 := \mathbb{P}[\mathsf{Forge}_0 \wedge \neg\mathsf{False}_0]$.

Claim 4.1. $\varepsilon - \varepsilon_0 \in \mathsf{negl}(\kappa)$.

Proof. By adaptive computational soundness of the NIWI argument system, we must have that $\mathbb{P}[\mathsf{False}_0] \in \mathsf{negl}(\kappa)$. In fact, from an adversary A provoking False_0, we can easily construct an adversary B breaking adaptive soundness by simply emulating the entire experiment for A and outputting one of the n proofs π_i^* contained in A's forgeries (chosen uniformly at random). In case $\mathbb{P}[\mathsf{False}_0]$ is non-negligible, B breaks soundness with noticeable advantage, so False_0 must happen with negligible probability. Hence, there exists a negligible function $\nu_0 : \mathbb{N} \to [0, 1]$ such that

$$\begin{aligned}\varepsilon_0 = \mathbb{P}\left[\mathsf{Forge}_0 \wedge \neg\mathsf{False}_0\right] &= \mathbb{P}\left[\mathsf{Forge}_0\right] - \mathbb{P}\left[\mathsf{Forge}_0 \wedge \mathsf{False}_0\right] \\ &\geq \mathbb{P}\left[\mathsf{Forge}_0\right] - \mathbb{P}\left[\mathsf{False}_0\right] \geq \varepsilon - \nu_0(\kappa),\end{aligned}$$

as desired. □

We now define a series of hybrid experiments. For each hybrid $\mathcal{H}_i$, we write ε_i for the probability of the event $\mathsf{Forge}_i \wedge \neg\mathsf{False}_i$, and View_i for the view of the adversary at the end of the experiment. The first sequence of hybrids are made to switch the public key of the signature scheme to a simulated public key. The simulated public key is generated together with a trapdoor, and moreover does not reveal any information about the signing key.

Experiment $\mathcal{H}_{1.a}$: This experiment is the same as $\mathcal{H}_0$, except that during key generation the verification key for the commitment scheme $\mathcal{COM}_1$ is computed using the equivocable setup algorithm ESetup_1, a random index $i^* \in [t]$ is sampled, and all verification keys for the commitment scheme $\mathcal{COM}_2$ but the i^*-th are computed using the equivocable setup algorithm ESetup_2 (whereas ϑ_{i^*} is computed as before). More precisely,

$$\begin{cases} (\vartheta, \tau) \leftarrow\!\!\$\ \mathsf{ESetup}_1(1^\kappa) & \\ (\vartheta_i, \tau_i) \leftarrow\!\!\$\ \mathsf{ESetup}_2(1^\kappa) & \text{for } i \neq i^* \\ \vartheta_i \leftarrow\!\!\$\ \mathsf{Setup}_2(1^\kappa) & \text{for } i = i^*. \end{cases}$$

Note that from the point of view of the adversary, the initial state remains equal to $st := ((\boldsymbol{\Delta}, \mathbf{r}), \mathsf{crs}, \vartheta, \{\vartheta_i\}_{i=1}^t)$.

Claim 4.2. $\varepsilon_0 - \varepsilon_{1.a} \in \mathsf{negl}(\kappa)$.

Proof. First note the distributions of Setup_1 and ESetup_1 are the same. The claim follows by a standard hybrid argument relying on the hybridness property of the commitment scheme. □

Experiment $\mathcal{H}_{1.b}$: This experiment is the same as the previous hybrid, except that now the commitments $\{com_i\}_{i=0}^d$ to the columns $\boldsymbol{\delta}_i$ are replaced by equivocable commitments, i.e. $(com_i, r'_i) \leftarrow\!\!\$\ \mathsf{ECom}_1(\vartheta, \tau)$ for all $i \in [0, d]$. Notice that the actual randomness r_i, used to produce com_i in $\mathcal{H}_{1.a}$, can be recovered efficiently as a function of the coefficients $\boldsymbol{\delta}_i$ and the fake randomness r'_i, as $r_i(\boldsymbol{\Delta}) := \mathsf{Equiv}_1(\tau, \boldsymbol{\delta}_i, r'_i)$. Given r_i the signature computation is identical in $\mathcal{H}_{1.a}$ and $\mathcal{H}_{1.b}$; in the following we write $\mathbf{r}(\boldsymbol{\Delta})$ for the vector $(r_0(\boldsymbol{\Delta}), \ldots, r_d(\boldsymbol{\Delta}))$.

Claim 4.3. $\varepsilon_{1.a} - \varepsilon_{1.b} \in \mathsf{negl}(\kappa)$.

Proof. The trapdoor hiding property of the commitment scheme implies that the distribution of each pair (com_i, r_i) in the two hybrids are statistically close. The claim follows by a standard hybrid argument. □

Experiment $\mathcal{H}_{1.c}$: This experiment is identical to the previous hybrid, except that the value y contained in the verification key is now computed by sampling a random $x \in \mathcal{X}$ and by outputting $y := G(x)$ (instead of sampling $y \in \mathcal{Y}$ uniformly at random).

Claim 4.4. $\varepsilon_{1.b} - \varepsilon_{1.c} \in \mathsf{negl}(\kappa)$.

Proof. It follows from the fact that G is a secure pseudorandom generator. □

The next sequence of hybrids is made in order to replace all signatures returned by the signing oracle to simulated signatures that, on average, do not reveal information about the secret key. Looking ahead, the trapdoor information that allows the above transition to go through consists of the tuple $(i^*, \tau, \{\tau_i\}_{i \neq i^*}, \{r'_i\}_{i=0}^d, x)$, i.e. the index i^* corresponding to the perfectly binding verification key, the trapdoors for all other verification keys, the randomness r'_i used to equivocate the values com_i, and the seed x of the pseudorandom generator.

Experiment $\mathcal{H}_{2.a}$: This hybrid differs from the previous one in the way *leak-free* signature queries are dealt with. Specifically, for each query $m \in \mathbb{F}$ the proof π is computed using x as witness for $y = G(x)$ instead of $\beta = (\tilde{m}, \tilde{r}, r)$ as witness for $\alpha = (\vartheta, \vartheta_j, \tilde{com}, com) \in \mathcal{L}_{\mathfrak{R}}$.

Claim 4.5. $\varepsilon_{1.c} - \varepsilon_{2.a} \in \mathsf{negl}(\kappa)$.

Proof. Follows from the statistical witness indistinguishability property of the argument system, via a standard hybrid argument on the number of *leak-free* signature queries. □

Experiment $\mathcal{H}_{2.b}$: For each *leak-free* signature query of the form $m \in \mathbb{F}$, the experiment checks whether the sampled index $j \leftarrow\!\!\$\ [t]$ is such that $j = i^*$. If this is the case, such query is answered exactly as in the previous hybrid, and otherwise the value com in the signature σ is computed as $(com, r') \leftarrow\!\!\$\ \mathsf{ECom}_2(\vartheta_j, \tau_j)$.

Claim 4.6. $\varepsilon_{2.a} - \varepsilon_{2.b} \in \mathsf{negl}(\kappa)$.

Proof. Let $\sigma = (\pi, j, com)$ be the answer to a leak-free signature query $m \in \mathbb{F}$. The distribution of the proof π and of the index j is identical in the two hybrids. The trapdoor hiding property implies that the distribution of the commitment *com* in hybrid $\mathcal{H}_{2.b}$ is statistically close to the distribution of *com* in hybrid $\mathcal{H}_{2.a}$. The claim follows. □

Experiment $\mathcal{H}_{2.c}$: This experiment behaves in the same way as the previous hybrid, with the exception that for each *leak-free* query such that $j = i^*$ the value *com* is replaced with a commitment to zero, i.e. $com := \mathsf{Com}_2(\vartheta_{i^*}, 0^\mu; r)$ for $r \leftarrow\!\!\$\ \mathbb{F}^\nu$.

Claim 4.7. $\varepsilon_{2.b} - \varepsilon_{2.c} \in \mathsf{negl}(\kappa)$.

Proof. Recall that there are three types of signature queries handled by the two hybrids $\mathcal{H}_{2.b}$ and $\mathcal{H}_{2.c}$. Specifically:

1. Both hybrids handle in the same way *leaky* signature queries.
2. In case of a *leak-free* signature query on a message m both hybrids randomly select the index $j \leftarrow\!\!\$\ [t]$ and:
   (a) if $j \neq i^*$, both hybrids handle the signature in the same way;
   (b) if $j = i^*$, hybrid $\mathcal{H}_{2.b}$ computes *com* as an honestly generated commitment to $\mathbf{\Delta}(m)$, whereas hybrid $\mathcal{H}_{2.c}$ computes *com* by committing to zero. We call such a query *binding*.

Let $q' \in \mathsf{poly}(\kappa)$ be the total number of *leak-free* signature queries and let $K \in [q']$ be a random variable representing the number of such queries that are binding. Note that in case no signature query happens to be binding the two hybrids are identically distributed, and thus there is nothing to prove. On the other hand, we show that for any possible realization $k \in [q']$ of the random variable K the two hybrids are computationally close. This implies the claim.

Denote with $\mathcal{J} = \{j_1, \ldots, j_k\} \subseteq [q']$ the set of indexes corresponding to the binding signature queries. For an index $0 \leq w \leq k$, consider the following series of hybrids experiments. Hybrid $\mathcal{H}^w_{2.b}$ treats the first w binding signature queries (i.e., the ones up to index j_w) as $\mathcal{H}_{2.b}$ does, whereas the remaining queries (i.e., the ones going from index j_{w+1} to j_k) are treated as in $\mathcal{H}_{2.c}$. Note that $\mathcal{H}_{2.c} \equiv \mathcal{H}^0_{2.b}$, and $\mathcal{H}_{2.b} \equiv \mathcal{H}^k_{2.b}$.

By contradiction, assume that there exists an efficient distinguisher D that for some $w \in [0, k]$ can distinguish the distribution of $\mathcal{H}^w_{2.b}$ and $\mathcal{H}^{w+1}_{2.b}$ with non-negligible probability. Consider the following adversary B (using D) attacking the adaptive computational hiding property of the hybrid commitment scheme (cf. Lemma 2.2).

- At the beginning B receives a verification key ϑ^* and sets-up the verification key for the signature scheme with the same distribution as in $\mathcal{H}_{2.b}$ and $\mathcal{H}_{2.c}$, and setting $\vartheta_{i^*} = \vartheta^*$.
- All *leaky* signature queries, and all *non-binding* signature queries are treated in the same way as in $\mathcal{H}_{2.b}$ and $\mathcal{H}_{2.c}$.
- For the first $w - 1$ binding signature queries (corresponding to indexes $j_1, \ldots, j_{w-1}$), B computes *com* as in $\mathcal{H}_{2.b}$.
- Let m be the message corresponding to the w-th binding signature query. At this point B outputs $(0^\mu, \mathbf{\Delta}(m))$ and sets $com = com^*$, where com^* is the challenge commitment sent to B.
- For all remaining binding signature queries (corresponding to indexes $j_{w+1}, \ldots, j_k$), B computes *com* as in $\mathcal{H}_{2.c}$.
- In case D outputs a set of forgeries $(m^*_1, \sigma^*_1), \ldots, (m^*_n, \sigma^*_n)$, B performs the same checks as done in $\mathcal{H}_{2.b}$ and $\mathcal{H}_{2.c}$.
- Finally, B outputs whatever D does.

Note that the verification key simulated by B has the correct distribution; in fact, all public keys are set to be equivocable (as in both $\mathcal{H}_{2.b}$ and $\mathcal{H}_{2.c}$) but the one corresponding to coordinate i^*, which is taken from the reduction (and thus has the right distribution). Additionally, in case com^* is a commitment to $\mathbf{\Delta}(m)$ the reduction perfectly emulates the view of $\mathcal{H}^w_{2.b}$, whereas in case com^* is a commitment to 0^μ the reduction perfectly emulates the view of $\mathcal{H}^{w+1}_{2.b}$. Thus, B breaks the adaptive computational hiding property with non-negligible probability contradicting Lemma 2.2. This concludes the proof.

□

Experiment $\mathcal{H}_{2.d}$: In this experiment we modify the way *leaky* signature queries are answered. Given such a query (m, f), the experiment samples the index $j \leftarrow\!\!\$\ [t]$ as before and, in case $j \neq i^*$, computes the commitment com as $(com, r') \leftarrow\!\!\$\ \mathsf{ECom}_2(\vartheta_j, \tau_j)$. Notice that the randomness r used to compute com in the previous hybrid can be obtained as a function of the matrix $\mathbf{\Delta}$ by running the equivocation algorithm. In particular:

$$r(\mathbf{\Delta}) := \begin{cases} \mathsf{Equiv}_2(\tau_j, \mathbf{\Delta}(m), r') & \text{if } j \neq i^* \\ r \leftarrow\!\!\$\ \mathbb{F}^\nu & \text{if } j = i^*. \end{cases}$$

The state used to reply the current leaky query is set to $st = \{(\mathbf{\Delta}, \mathbf{r}), r(\mathbf{\Delta}), r_{niwi}\}$, where r_{niwi} is the randomness used to generate the NIWI argument π.

Claim 4.8. $\varepsilon_{2.c} - \varepsilon_{2.d} \in \mathsf{negl}(\kappa)$.

Proof. Let us consider the view of the adversary in the two hybrids, consisting of the verification key, and the answers to both leaky and leak-free signature queries. The two hybrids differ only in the way leaky queries are treated.

Consider one such query. In case $j = i^*$ the two hybrids distributions are identical; in case $j \neq i^*$ the trapdoor hiding property implies that the distribution of the pair (com, r) in $\mathcal{H}_{2.d}$ is statistically close to the distribution of the same pair in $\mathcal{H}_{2.c}$. Note that in both experiments, the verification key vk, the argument π, the leakage Λ are computed as a function of $\mathbf{\Delta}$, $r(\mathbf{\Delta})$ and independent randomness. It follows that the answers to each leaky query are statistically close in the two experiments. By a standard hybrid argument, it must be that the joint distributions $((\mathbf{\Delta}, \mathbf{r}(\mathbf{\Delta})), \mathsf{View}_{2.c})$ and $((\mathbf{\Delta}, \mathbf{r}(\mathbf{\Delta})), \mathsf{View}_{2.d})$ are statistically close. The claim follows. □

From now on the proof is purely information theoretic, so in particular the next experiments might no longer be efficient. Before defining the next hybrid, we state a lemma implicit in [9].

Lemma 4.1. *Let $\mathfrak{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ be an NP relation, and consider a statistical NIWI argument system for $\mathfrak{R}$. There exists a (possibly not efficient) reconstruction algorithm* Rand *that given as input a proof π for an instance x under a witness w, another witness w' for x, and the randomness r_{niwi} used to generate π, outputs a value r'_{niwi} such that $\pi = \mathsf{Prove}(\mathsf{crs}, x, w'; r'_{niwi})$ (with overwhelming probability over the choice of r_{niwi}). Additionally, the distribution of r'_{niwi} is statistically close to the uniform distribution.*

Proof. For any $(x, w) \in \mathfrak{R}$, and $\mathsf{crs} \leftarrow\!\!\$\ \mathsf{Init}(1^\kappa)$, let $\Pi_w := \{\pi |\ \pi = \mathsf{Prove}(\mathsf{crs}, x, w; r_{niwi})\}$. The statistical witness indistinguishability property implies:

$$\mathbb{P}_{\pi \leftarrow\!\!\$\ \mathsf{Prove}(\mathsf{crs}, x, w)} [\pi \notin \Pi_{w'}] \in \mathsf{negl}(\kappa).$$

This is because otherwise the event $\pi \in \Pi_{w'}$ can be used to distinguish the ensembles Π_w and $\Pi_{w'}$. In case the above condition is satisfied, algorithm Rand samples r'_{niwi} from the weighted distribution $\{r |\ \pi = \mathsf{Prove}(\mathsf{crs}, x, w'; r)\}$ (otherwise it aborts). □

Experiment $\mathcal{H}_{2.e}$: This experiment is identical to the previous hybrid, except that (whenever possible) it uses a different witness β' to compute the argument corresponding to a *leaky* signature query. In particular, given such a query (m, f), the experiment samples $j \leftarrow\!\!\$\, [t]$ and, in case $j \neq i^*$, it generates the argument π by running

$$\mathsf{Prove}(\mathsf{crs}, \underbrace{(\vartheta, \vartheta_j, \tilde{com}, com)}_{\alpha}, \underbrace{(0, \mathsf{Equiv}_1(\tau, 0^\mu, r'(m)), \mathsf{Equiv}_2(\tau_j, 0^\mu, r')}_{\beta'}; r'_{niwi}),$$

where $r'(m) = \sum_{i=0}^{d} r'_i \cdot m^i$ is computed using the randomness $\{r'_i\}_{i=0}^{d}$. Notice that the randomness r_{niwi} used to generate the NIWI argument in the previous experiment can be sampled (inefficiently) as a function of the witness $\beta := (\mathbf{\Delta}(m), \mathsf{Equiv}_1(\tau, \mathbf{\Delta}(m), r'(m)), \mathsf{Equiv}_2(\tau_j, \mathbf{\Delta}(m), r'))$ and r'_{niwi}. In particular:

$$r_{niwi}(\mathbf{\Delta}) := \begin{cases} r_{niwi} \text{ (as sampled by the prover)} & \text{if } j = i^* \\ \mathsf{Rand}(\pi, r'_{niwi}, \beta) & \text{if } j \neq i^*, \end{cases}$$

where Rand is the reconstruction algorithm of Lemma 4.1. The state used to answer the leakage query f is set to $st(\mathbf{\Delta}) = ((\mathbf{\Delta}, \mathbf{r}(\mathbf{\Delta})), r(\mathbf{\Delta}), r_{niwi}(\mathbf{\Delta}))$.

Claim 4.9. $\varepsilon_{2.d} - \varepsilon_{2.e} \in \mathsf{negl}(\kappa)$.

Proof. The linear homomorphic property of the hybrid commitment scheme (cf. Section 2.3) ensures that the value $r'(m)$ is the right randomness to equivocate $\tilde{com}$.

By statistical witness indistinguishability, for each leaky query such that $j \neq i^*$ the distribution of the argument π is statistically close in the two hybrids. The same holds for the reconstructed state, by Lemma 4.1. It follows that $((\mathbf{\Delta}, \mathbf{r}(\mathbf{\Delta})), \mathsf{View}_{2.d})$ and $((\mathbf{\Delta}, \mathbf{r}(\mathbf{\Delta})), \mathsf{View}_{2.e})$ are statistically close, which concludes the proof. □

The next hybrid that we define has no direct access to the matrix $\mathbf{\Delta}$, but instead depends on a leakage oracle $\mathcal{O}_{\mathbf{\Delta}}(\cdot)$ which takes as input a function f and returns $f(\mathbf{\Delta})$.

Experiment $\mathcal{H}_3^{\mathcal{O}_{\mathbf{\Delta}}(\cdot)}$: This experiment is the same as the previous hybrid, with the difference that $\mathbf{\Delta}$ is not sampled by the hybrid as part of the signing key, but can instead be accessed via $\mathcal{O}_{\mathbf{\Delta}}(\cdot)$. In particular, all *leak-free* signature queries and *leaky* signature queries where $j \neq i^*$, are handled as in $\mathcal{H}_{2.e}$. For a *leaky* signature query (m, f) where the index j happens to be equal to i^*, the experiment defines the leakage function $f'_{m,(r'_0,\ldots,r'_d),r',r'_{niwi}} := (f(st(\mathbf{\Delta})), \mathbf{\Delta}(m))$ and queries $\mathcal{O}_{\mathbf{\Delta}}(\cdot)$ on f'.[7] We call such query a *bad* query and we let Bad denote the corresponding event. Given the value $\mathbf{\Delta}(m)$, the signature σ can be computed as in $\mathcal{H}_{2.e}$.

Claim 4.10. $\varepsilon_3 = \varepsilon_{2.e}$.

Proof. The only difference between the two hybrids stems from the fact that $\mathbf{\Delta}$ is only accessed via a leakage oracle in $\mathcal{H}_3$. Note that in $\mathcal{H}_{2.e}$ the matrix $\mathbf{\Delta}$ is needed only to answer leaky signature queries; in case of a bad query, $\mathcal{H}_3$ accesses the leakage oracle to additionally obtain $\mathbf{\Delta}(m)$ in addition to the usual leakage on the state. Given $\mathbf{\Delta}(m)$, the signature computation is identical in the two hybrids. □

[7] Note that the function hard-wires the message m, and the randomness needed to reconstruct the current state as a function of $\mathbf{\Delta}$ (as specified in the previous hybrids).

In what follows, let View_3 be the entire view of experiment $\mathcal{H}_3$; this includes the verification key vk, and the answers to all leaky and leak-free signature queries. The two lemmas below characterize the conditional min-entropy of $\mathbf{\Delta}$ given View_3. The first lemma says that, with high probability over the randomness in the experiment, the only information revealed on $\mathbf{\Delta}$ comes from the leakage and the bad queries. Recall that we assume the adversary A makes exactly q *leaky* queries.

Lemma 4.2. *For any $\beta > 0$ we have that*

$$\mathbb{P}\left[\mathbb{H}_\infty\left(\mathbf{\Delta}|\ \mathsf{View}_3 = v\right) \geqslant |\mathbf{\Delta}| - (2eq/t)\mu\log|\mathbb{F}| - \beta - \lambda\right] > 1 - 2^{-\beta} - 2^{-2eq/t}, \tag{3}$$

where the probability is taken over the randomness of the experiment.

Proof. Let q_s be the total number of signature queries made by the adversary. We write $\mathcal{L} \subseteq [q_s]$ for the set of indexes corresponding to the leaky signature queries. Furthermore, we let $\mathcal{Z} := \{i \in \mathcal{L} \wedge \mathsf{Bad}_i\}$ be the set of indexes corresponding to the bad queries. Note that the cardinality Z of $\mathcal{Z}$ is a random variable, depending on the event that the index $j \leftarrow\!\!\$\ [t]$ (sampled independently for each leaky query) happens to hit the index i^*.

The random variable View_3 consists of the simulated verification key, the answers to all leak-free signature queries and the answers to all leaky signature queries made by the adversary. Let $\mathsf{View}'_3 := (\Lambda_i)_{i\in\mathcal{L}}||(\mathbf{\Delta}(m_i))_{i\in\mathcal{Z}}$ the total information leaked in $\mathcal{H}_3$. Recall that in $\mathcal{H}_3$ both the verification key, the answers to leak-free queries and the signatures corresponding to leaky queries where the event Bad does not happen, are completely independent on $\mathbf{\Delta}$. Hence,

$$\mathbb{H}_\infty\left(\mathbf{\Delta}|\ \mathsf{View}_3 = v\ \right) = \mathbb{H}_\infty\left(\mathbf{\Delta}|\ \mathsf{View}'_3 = v'\ \right),$$

where for any realization of the adversary's view v, we write v' for the "striped-off" view that considers only the total information retrieved via the leakage oracle.

Observe that $|\mathcal{Z}| = Z = \sum_{i\in[q_s]} \mathsf{Bad}_i$, and in particular $|\mathsf{View}'_3| \leqslant \lambda + Z\mu\log|\mathbb{F}|$. Furthermore, note that the number of *leaky* queries is equal to q, so that $\mathbb{E}[Z] = q/t$. Since the events Bad_i are independent, we can apply a Chernoff bound and write:

$$\mathbb{P}\left[Z > 2eq/t\right] < 2^{-2eq/t}.$$

Setting $\lambda' := \lambda + (2eq/t)\mu\log|\mathbb{F}|$ and $\Delta := \lambda' + \beta$, we obtain

$$\begin{aligned}
&\mathbb{P}\left[\mathbb{H}_\infty\left(\mathbf{\Delta}|\mathsf{View}'_3 = v'\right) \geqslant |\mathbf{\Delta}| - \Delta\right] \\
&\quad\geqslant \mathbb{P}\left[\mathbb{H}_\infty\left(\mathbf{\Delta}|\mathsf{View}'_3 = v'\right) \geqslant |\mathbf{\Delta}| - \Delta \ \middle|\ Z \leqslant 2eq/t\right](1 - 2^{-2eq/t}) = \\
&\quad= \mathbb{P}\left[\mathbb{H}_\infty\left(\mathbf{\Delta}|\mathsf{View}'_3 = v'\right) \geqslant |\mathbf{\Delta}| - \Delta \ \middle|\ |\mathsf{View}'_3| \leqslant \lambda'\right](1 - 2^{-2eq/t}).
\end{aligned} \tag{4}$$

Applying Lemma 2.1 to the right hand side of the last inequality, we have:

$$\mathbb{P}\left[\mathbb{H}_\infty\left(\mathbf{\Delta}|\mathsf{View}'_3 = v'\right) \geqslant |\mathbf{\Delta}| - \Delta \ \middle|\ |\mathsf{View}'_3| \leqslant \lambda'\right] \geqslant 1 - 2^{\lambda'-\Delta} = 1 - 2^{-\beta}. \tag{5}$$

The lemma follows by combining inequality (4) and (5). □

Lemma 4.3. *For any adversary A such that $\mathbb{P}[\widetilde{\mathbf{Exp}}^{\text{one-more}}_{\mathcal{SS}_1,\mathsf{A}}(\kappa,\ell,q_s,\gamma) = 1] = \varepsilon$, there exists a constant $c > 0$ such that:*

$$\mathbb{E}\left[\mathbb{H}_\infty\left(\mathbf{\Delta}|\ \mathsf{View}_3 = v\right)\right] \leqslant \left(d + 1 - \frac{n - 1 + q}{t}\right)\mu\log|\mathbb{F}| - c\log\varepsilon(\kappa),$$

where the expectation is taken over the randomness of the experiment.

Proof. To prove the lemma we define a predictor strategy P (running A) that outputs $\boldsymbol{\Delta}$ with "high enough" probability.

Let $\mathcal{K}$ be the set of indexes such that $i \in \mathcal{K}$ if and only if the i-th forged signature σ_i^* is of the form $(\pi_i^*, i^*, \tilde{com}_i^*)$. W.l.o.g. let us assume that $\mathcal{K} := \{1, \ldots, K\}$ for a random variable $K \in [n]$. The predictor P proceeds as follows:

1. At the beginning P interacts with A, playing the role of the challenger in $\mathcal{H}_3$. Let $\{m_i\}_{i \in \mathcal{Z}}$ be the set of messages corresponding to the *bad* leaky signature queries, i.e. to the queries for which $\mathcal{H}_3$ retrieved the vector $\boldsymbol{\Delta}(m_i)$ via the leakage oracle $\mathcal{O}_{\boldsymbol{\Delta}}(\cdot)$.
2. Whenever A outputs a set of forgeries $(m_1^*, \sigma_1^*), \ldots, (m_n^*, \sigma_n^*)$, P finds, for all $i \in [K]$, the unique vector $\boldsymbol{\Delta}(m_i^*)$ such that $com_i^* = \mathsf{Com}_2(\vartheta_{i^*}, \boldsymbol{\Delta}(m_i^*); r_i^*)$ (for some randomness $r_i^* \in \mathbb{F}^\nu$).
3. For each $j \in [\mu]$, P solves the following linear system:

$$\begin{pmatrix} 1 & m_1 & \ldots & m_1^d \\ & & \ddots & \\ 1 & m_Z & \ldots & m_Z^d \\ 1 & m_1^* & \ldots & {m_1^*}^d \\ & & \ddots & \\ 1 & m_K^* & \ldots & {m_K^*}^d \\ 1 & \bar{m}_1 & \ldots & \bar{m}_1^d \\ & & \ddots & \\ 1 & \bar{m}_{d+1-K-Z} & \ldots & \bar{m}_{d+1-Z-K}^d \end{pmatrix} \cdot \begin{pmatrix} \delta_{j,0} \\ \vdots \\ \delta_{j,d} \end{pmatrix} = \begin{pmatrix} \delta_j(m_1) \\ \vdots \\ \delta_j(m_Z) \\ \delta_j(m_1^*) \\ \vdots \\ \delta_j(m_K^*) \\ y_{j,1} \\ \vdots \\ y_{j,d+1-Z-K} \end{pmatrix}, \tag{6}$$

   where $\delta_j(X)$ is the polynomial defined by the coefficients in the j-th row of $\boldsymbol{\Delta}$. The values $y_1, \ldots, y_{d+1-Z-K}$ are chosen uniformly at random from $\mathbb{F}$, whereas the values $\bar{m}_1, \ldots, \bar{m}_{d+1-Z-K}$ are chosen uniformly in $\mathbb{F} \setminus (\{m_i\}_{i \in [Z]} \cup \{m_i^*\}_{i \in [K]})$.
4. Output the matrix $\boldsymbol{\Delta}'$ having $(\delta_{j,i})_{j \in [\mu], i \in [0,d]}$ as elements.

Recall that the common reference string corresponding to index i^* is perfectly binding. Hence P always succeeds in finding the values $\{\boldsymbol{\Delta}(m_i^*)\}_{i \in [K]}$, and these values are unique. Moreover, since we are conditioning on False_3 not happening, the extracted values contain the evaluation of the polynomials $\delta_1(X), \ldots, \delta_\mu(X)$ at m_i^*. Also note that the system of Eq. (6) is full-rank.

We conclude that the matrix $\boldsymbol{\Delta}'$ is equal to $\boldsymbol{\Delta}$, provided that P guessed correctly the values $y_{j,1}, \ldots, y_{j,d-Z-K}$ for all $j \in [\mu]$, and that the forgeries output by A are such that $\mathsf{Forge}_3 \wedge \neg \mathsf{False}_3$ happens. The probability of the first event is equal to $|\mathbb{F}|^{-\mu(d+1-Z-K)}$. The probability of the second event is equal to ε_3; combining Claims 4.1—4.10 there exists a constant c such that $\varepsilon_3(\kappa) \geq \varepsilon(\kappa)^c$. Thus,

$$\mathbb{H}_\infty\left(\boldsymbol{\Delta} \middle| \, \mathsf{View}_3 = v\right) \leq \mu(d + 1 - Z - K) \log |\mathbb{F}| - c \log \varepsilon(\kappa).$$

The lemma now follows by noticing that $\mathbb{E}[K] > \frac{n-1}{t}$, which implies

$$\mathbb{E}[-Z - K] \leq -\frac{n - 1 + q}{t}.$$

□

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. We start by computing the leakage bound. To this end, let us set the parameters such that in Lemma 4.2 the probability that the conditional min-entropy of $\mathbf{\Delta}$ given the view in the last hybrid is at least 1. Let $\beta > 0$, for concreteness say $\beta = 2$; furthermore let $t := \frac{2e\rho q}{\xi(d+1)} - 1$ for any $0 < \xi \leq \rho$. Lemma 4.2 requires that:

$$\begin{aligned}\lambda &\leqslant |\mathbf{\Delta}| - (2eq/t)\mu \log |\mathbb{F}| - 3 \\ &= |\mathbf{\Delta}| - (\xi/\rho(d+1) + 1)\mu \log |\mathbb{F}| - 3,\end{aligned}$$

which is implied by:

$$\begin{aligned}\lambda &\leqslant |\mathbf{\Delta}| - (\xi/\rho)(d+1)\mu \log |\mathbb{F}| = \\ &= (1 - \xi/\rho)|\mathbf{\Delta}| = (\rho - \xi)|sk|.\end{aligned}$$

W.l.o.g. let us assume $\lambda = (\rho - \xi)|sk|$; since $q \leq \lambda$ we get

$$t = \frac{2e\rho q}{\xi(d+1)} - 1 \leqslant \frac{2e\rho\lambda}{\xi(d+1)} = \frac{2e\rho(1-\xi/\rho)(d+1)\mu\kappa}{\xi(d+1)} = \frac{2e\rho(1-\xi/\rho)\mu}{\xi}\kappa = O(\kappa).$$

Lemma 4.3 implies:

$$\mathbb{P}\left[\mathbb{H}_\infty\left(\mathbf{\Delta} \middle| \ \mathsf{View}_3 = v\right) \leqslant \frac{\left(d + 1 - \frac{n-1+q}{t}\right)\mu \log |\mathbb{F}| - c \log \varepsilon(\kappa)}{1 - 2^{-\beta} - 2^{-2eq/t}}\right] \geqslant 2^{-\beta} + 2^{-2eq/t}. \tag{7}$$

Notice that Eq. (3) and Eq. (7) give, respectively, a lower bound and an upper bound on the conditional min-entropy of the random variable $\mathbf{\Delta}$ given the entire view at the end of the experiment. We show that, whenever $\varepsilon(\kappa)$ is non-negligible, by setting the parameters as in the statement of the theorem the lower bound is strictly bigger than the upper bound. It follows that the event of Eq. (3) implies the negation of the event of Eq. (7). We obtain a contradiction by noticing that the sum of the two probabilities is strictly bigger than 1.

Hence, we need to show that the following holds:

$$\frac{\left(d + 1 - \frac{n-1+q}{t}\right)\mu \log |\mathbb{F}| - c \log \varepsilon(\kappa)}{1 - 2^{-\beta} - 2^{-2eq/t}} < (d+1)\mu \log |\mathbb{F}| - (2eq/t)\mu \log |\mathbb{F}| - 2 - \lambda. \tag{8}$$

Substituting $\log |\mathbb{F}| = \kappa$, and rearranging, we obtain:

$$\frac{(n-1)\mu\kappa}{t} > (2^{-\beta} + 2^{-2eq/t} - 1) \cdot ((d + 1 - 2eq/t)\mu\kappa - 2 - \lambda) + (d + 1 - q/t)\mu\kappa - c \log \epsilon(\kappa).$$

For the last inequality to hold it suffices that

$$\frac{(n-1)\mu\kappa}{t} > (d + 1 - q/t)\mu\kappa + (2e-1)q/t + 4 + 2\lambda - c \log \varepsilon.$$

Recall that $q/t \geqslant \frac{\xi(d+1)}{2e\rho}(d+1)$ and $(d+1) = \frac{\lambda}{\mu\kappa(\rho-\xi)}$, hence, for the last inequality to hold, it suffices that

$$\frac{(n-1)\mu\kappa}{t} > \left(2 + \frac{1 - \xi/2e\rho}{\rho - \xi} + \frac{2e-1}{2e\rho}\right)\lambda + 4 - c \log \varepsilon(\kappa).$$

Thus, in order for Eq. (8) to hold, it suffices that

$$\frac{n\kappa}{t} = \Omega\left(\lambda - \log \varepsilon(\kappa)\right).$$

If $\varepsilon(\kappa)$ is non-negligible, recalling that $t = O(\kappa)$ and $\lambda = \Omega(\kappa)$, we get that n is linear in λ. Since, by definition, $n \geq \lfloor \frac{\lambda}{\gamma \cdot s} \rfloor + 1$ and a signature consists of a constant number of group elements, it suffices to set the slack parameter γ as $\gamma(\kappa) = O(1/\kappa)$ in order for Eq. (8) to hold. □

4.3 A Variant

The construction of Section 4.1 implicitly defines an *unbounded simulation sound* NIZK. In fact, the OR-proof technique resembles the unbounded simulation sound NIZK argument systems given in [39, 38, 12]. The main drawback, as noticed in [25, 48], is that the arguments for the relation $\mathfrak{R}_{OR}$ instantiated using the Groth-Sahai proof system [40] need to use quadratic equations, and thus can be significantly less efficient than the arguments for $\mathfrak{R}$.

In this section we show how to avoid the expensive OR-proof trick. The technical tools we use are:

- A trapdoor hiding, linearly homomorphic commitment scheme $\mathcal{COM}_1$ as in the scheme from Section 4.1.
- A hybrid commitment $\mathcal{COM}_2$ as in the scheme from Section 4.1, with the additional property that the public key space is a group $\mathbb{G}$ of cardinality p with $\log p = O(\kappa)$, and the equivocable keys form a subgroup $\mathbb{H}$ of $\mathbb{G}$ (with cardinality p'). We also require that: (a) for any two elements $g_1 \in \mathbb{G} \setminus \mathbb{H}, g_2 \in \mathbb{H}$ the element $g_1 \cdot g_2 \in \mathbb{G} \setminus \mathbb{H}$, and (b) the ESetup_2 algorithm produces an equivocable key that is uniformly distributed in $\mathbb{H}$.
- A statistical NIWI argument system $\mathcal{NIWI} = (\mathsf{Init}_1, \mathsf{Prove}_1, \mathsf{Ver}_1)$ for the same relation $\mathfrak{R}_1 := \mathfrak{R}$ of the construction from Section 4.1.
- A true-simulation extractable NIZK argument system $\mathcal{NIZK} = (\mathsf{Init}_2, \mathsf{Prove}_2, \mathsf{Ver}_2)$ (see Section 2.4) supporting labels, and with obvious sampling of the common reference string, for the relation

$$\mathfrak{R}_2 := \left\{ (\vartheta, \vartheta_1, \vartheta_2); (\alpha 1, \alpha_2) \;\middle|\; \vartheta = \vartheta_1^{\alpha_1} \cdot \vartheta_2^{\alpha_2} \;\right\} \subseteq \mathbb{G}^3 \times \mathbb{Z}_p^2. \tag{9}$$

  For brevity, given a vector $\mathbf{g} \in \mathbb{G}^n$ and $\boldsymbol{\alpha} \in \mathbb{Z}_p^n$ we write $\langle \mathbf{g}, \boldsymbol{\alpha} \rangle = \prod_{i=1}^n g_i^{\alpha_i}$.

Our scheme $\mathcal{SS}_2 = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ has message space equal to $\mathbb{F}$ and is described below:

Key Generation. Let $t, d, \mu, \nu \in \mathbb{N}$ be parameters. Run $\mathsf{crs}_1 \leftarrow\!\!\$\ \mathsf{Init}_1(1^\kappa)$ and $\mathsf{crs}_2 \leftarrow\!\!\$\ \mathsf{Init}_2(1^\kappa)$, and sample the values ϑ, $\{\vartheta_i\}_{i=1}^t$, $\boldsymbol{\Delta} \in (\mathbb{F}^\mu)^{d+1}$, $\mathbf{r} \in \mathbb{F}^{d+1}$, and $\{com_i\}_{i=1}^d$ as in the scheme from Section 4.1. Output

$$sk = (\boldsymbol{\Delta}, \mathbf{r}) \qquad vk = (\mathsf{crs}_1, \mathsf{crs}_2, \vartheta, \{\vartheta_i\}_{i=1}^t, \{com_i\}_{i=0}^d).$$

Signature. For $j \in [\mu]$, let $\delta_j(X)$, $r(X)$, and $\boldsymbol{\Delta}(X)$ be defined as in the scheme from Section 4.1. Consider the following polynomial-time relation:

$$\mathfrak{R}_1 := \left\{ (\vartheta, \vartheta', \tilde{com}, com); (\tilde{m}, \tilde{r}, r) \;\middle|\; \begin{array}{l} \tilde{com} = \mathsf{Com}_1(\vartheta, \tilde{m}; \tilde{r}) \\ com = \mathsf{Com}_2(\vartheta', \tilde{m}; r) \end{array} \right\}.$$

To sign a message $m \in \mathbb{F}$ compute $\tilde{m} = \boldsymbol{\Delta}(m)$ and $\tilde{r} = r(m)$, and let $\tilde{com} := \mathsf{Com}_1(\vartheta, \tilde{m}; \tilde{r})$. Notice 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. Then proceed as follows:

1. Pick random indexes $j_1, j_2 \leftarrow\!\!\$\ [t]$ such that $j_1 \neq j_2$, sample $\alpha_1, \alpha_2 \leftarrow\!\!\$\ \mathbb{Z}_p$ and compute $\vartheta' := \vartheta_{j_1}^{\alpha_1} \cdot \vartheta_{j_2}^{\alpha_2}$ and $com := \mathsf{Com}_2(\vartheta', \tilde{m}; r)$ where $r \leftarrow\!\!\$\ \mathbb{F}^\nu$.
2. Using crs_1 as common reference string, generate a NIWI argument π_1 for $(\vartheta, \vartheta', \tilde{com}, com) \in \mathfrak{L}_{\mathfrak{R}_1}$.
3. Using crs_2 as common reference string, generate a NIZK argument π_2 for $(\vartheta', \vartheta_{j_1}, \vartheta_{j_2}) \in \mathfrak{L}_{\mathfrak{R}_2}$ using m as a label.

4. Output $\sigma = (j_1, j_2, \vartheta', com, \pi_1, \pi_2)$.

Verification Given a pair (m, σ), parse σ as $(j_1, j_2, \vartheta', com, \pi_1, \pi_2)$. Output:

$$\mathsf{Ver}_1(\mathsf{crs}_1, \pi_1, (\vartheta, \vartheta', \prod_{i=1}^{d} com_i^{m^i}, com)) \quad \wedge \quad \mathsf{Ver}_2(\mathsf{crs}_2, \pi_2, m, (\vartheta', \vartheta_{j_1}, \vartheta_{j_2})).$$

Theorem 4.2. *Let $\mu \in \mathbb{N}$, and let $\mathbb{F}$ be a finite field of size $\log |\mathbb{F}| = \kappa$ for security parameter $\kappa \in \mathbb{N}$. For any constant $0 < \xi \leqslant \frac{\mu}{\mu+1}$ and $t = O(\kappa)$, whenever $\gamma = O(1/\kappa)$ the above signature scheme is $((\frac{\mu}{\mu+1} - \xi)|sk|, \gamma)$-fully-leakage one-more unforgeable with perfect erasures.*

The main difference with the proof of Theorem 4.1 is in the way we simulate leak-free queries. In the previous proof the simulator knows a value y, pre-image of x under the PRG G, and therefore it can produce a valid argument for the OR-relation and simulate correctly leak-free queries thanks to the WI property of $\mathcal{NIWI}$. Note, however, that whenever the sampled index $j \leftarrow\!\!\$\, [t]$ is such that $j \neq i^*$ (i.e., both the verification keys for the commitment scheme are equivocable) a valid NIWI argument can be produced by opening the commitments to 0.

Unfortunately the above strategy cannot be always used by the simulator, as the index j is part of the signature (and thus known to the adversary). In other words, although the simulator knows where the equivocable keys are, it cannot exploit this extra knowledge, as it needs to sample j using the same distribution used by the real signer.

The idea of re-randomizing the verification keys each time we generate a signature, allows the simulator to exploit the above extra knowledge. In fact, whenever the index j corresponding to the binding verification key is chosen, the simulator can force the re-randomized verification key ϑ' to always be equivocable. The zero-knowledge and the hybridness property ensure that the view of the adversary is computationally indistinguishable from the view in the real experiment; using the message m as label avoids the simple "cut-and-paste" attack in which an adversary re-uses a pair (ϑ', π_2) from a leak-free signature query in its forgeries.[8]

4.4 Proof Sketch of Theorem 4.2

The proof of Theorem 4.2 uses a hybrid argument, along the same lines of the proof of Theorem 4.1. Since many of the hybrids are identical, we only describe the steps that are different in full details.

Let A be an adversary making $q = \mathsf{poly}(\kappa)$ *leaky* signature queries, with advantage ε in the fully-leakage one-more unforgeability experiment for $\mathcal{SS}_2$. Let $\mathcal{H}_0$, $\mathcal{H}_{1.a}$, $\mathcal{H}_{1.b}$, Forge_0 be the same as in the proof of Theorem 4.1. Let False_0^1 be the same as False_0, and define $\varepsilon_i := \mathbb{P}[\mathsf{Forge}_i \wedge \neg \mathsf{False}_i^1]$. A reasoning similar to the one in Claim 4.1–4.3 shows that $\varepsilon - \varepsilon_{1.b} \in \mathsf{negl}(\kappa)$.

Experiment $\mathcal{H}_{2.a}$: This hybrid differs from the previous one in the way *leak-free* signature queries are dealt with. Specifically, during key generation the common reference string crs_2 is sampled by running $(\mathsf{crs}_2, \tau_{sim}, \tau_{ext}) \leftarrow\!\!\$\, \overline{\mathsf{Init}}(1^\kappa)$ which additionally returns a simulation trapdoor τ_{sim} and an extraction trapdoor τ_{ext}. Moreover, for each query $m \in \mathbb{F}$ the NIZK argument π_2 is computed by running $\pi_2 \leftarrow\!\!\$\, \mathsf{Sim}(\tau_{sim}, (m, (\vartheta', \vartheta_{j_1}, \vartheta_{j_2})))$.

Let $\pi_{2,i}^*$ be the NIZK argument contained in the i-th forgery returned by the adversary. Also, let $(\alpha_{1,i}^*, \alpha_{2,i}^*) \leftarrow\!\!\$\, \mathsf{K}(\tau_{ext}, m_i^*, \pi_{2,i}^*)$ be the witness extracted from such argument. Define $\mathsf{False}_{2.a}^2$

[8] In this case the predictor of Lemma 4.3 does not work.

to be the event that there exists an index $i \in [n]$ such that the extracted witness $(\alpha^*_{1,i}, \alpha^*_{2,i})$ is not valid, i.e.,

$$\vartheta'^* \neq \left\langle (\vartheta_{j_1^*}, \vartheta_{j_2^*}), (\alpha^*_{1,i}, \alpha^*_{2,i}) \right\rangle .$$

For the hybrid $\mathcal{H}_{2.a}$, and for each of the hybrids below, we write ε'_i for the probability of the event $\mathsf{Forge}_i \wedge \neg\mathsf{False}^1_i \wedge \neg\mathsf{False}^2_i$.

Claim 4.11. $\varepsilon_{1.b} - \varepsilon'_{2.a} \in \mathsf{negl}(\kappa)$.

Proof. Note that, by true-simulation extractability, we have that $\mathbb{P}[\mathsf{False}^2_{2.a}] \in \mathsf{negl}(\kappa)$ which implies $\varepsilon'_{2.a} \geqslant \varepsilon_{2.a} - \mathsf{negl}(\kappa)$. Furthermore, by the adaptive multi-theorem zero-knowledge property of the NIZK argument system, we have that the views $\mathsf{View}_{2.a}$ and $\mathsf{View}_{1.b}$ are computationally indistinguishable, and thus $\varepsilon_{1.b} - \varepsilon_{2.a} \in \mathsf{negl}(\kappa)$ which implies the claim. □

Experiment $\mathcal{H}_{2.b}$: The same as the previous hybrid, but now for each *leak-free* signature query of the form $m \in \mathbb{F}$, in case one of the two random indexes (say j_2) is equal to i^* then set $\alpha_2 := 0$ and compute $\vartheta' = \vartheta_{j_1}^{\alpha_1}$ where $\alpha_1 \leftarrow\!\!\$\ \mathbb{Z}_{p'}$.

Claim 4.12. $\varepsilon'_{2.a} - \varepsilon'_{2.b} \in \mathsf{negl}(\kappa)$.

Proof. Note that for each *leak-free* signature query the verification key ϑ' is uniformly distributed in $\mathbb{H}$, and thus it has the same distribution as a verification key output by ESetup_2. The claim follows by the hybridness property of the commitment scheme. □

Experiment $\mathcal{H}_{2.c}$: The same as the previous hybrid, but now for each *leak-free* signature query of the form $m \in \mathbb{F}$, the value *com* of the signature σ is computed as $(com, r') \leftarrow\!\!\$\ \mathsf{ECom}_2(\vartheta', \tau')$.

Claim 4.13. $\varepsilon'_{2.b} - \varepsilon'_{2.c} \in \mathsf{negl}(\kappa)$.

Proof. Follows by a hybrid argument relying on the trapdoor hiding property of the hybrid commitment scheme. □

Hybrids $\mathcal{H}_{2.d}$ and $\mathcal{H}_{2.e}$ are identical to the corresponding hybrids in the proof of Theorem 4.1. Putting it all together we have shown $\varepsilon - \varepsilon'_{2.e} \in \mathsf{negl}(\kappa)$. One can show that the equivalents of Lemma 4.2 and Lemma 4.3 hold true with exactly the same parameters. The only difference is in the way we show that $\mathbb{E}[K] \geq \frac{n-1}{t}$ (in the proof of Lemma 4.3). In particular we have to evaluate (in expectation) the number of forged signatures for which the commitment is binding. Towards this, for a value $i \in [n]$, let $\mathsf{Binding}^i_{2.e}$ be the event that the verification key contained in the j-th forged signature of hybrid $\mathcal{H}_{2.e}$ is an element of $\mathbb{G} \setminus \mathbb{H}$. It is easy to see that, for all $i \in [n]$, there exists a negligible function $\nu : \mathbb{N} \to [0, 1]$ such that

$$\mathbb{P}[\mathsf{Binding}^i_{2.e}] = \mathbb{P}[\mathsf{Binding}^i_{2.c}] \geq \mathbb{P}[\mathsf{Binding}^i_{2.b}] - \nu(\kappa),$$

where the inequality comes from the hybridness property of the commitment scheme. Furthermore, $\mathbb{P}[\mathsf{Binding}^i_{2.b}] = \mathbb{P}[\mathsf{Binding}^i_{2.a}]$ so we are left to bound the probability of $\mathsf{Binding}^i_{2.a}$. We have that, for a negligible function $\nu' : \mathbb{N} \to [0, 1]$,

$$\begin{aligned}
\mathbb{P}[\mathsf{Binding}^i_{2.a}] &\geqslant \mathbb{P}[\mathsf{Binding}^i_{2.a} |\ \neg\mathsf{False}^2_{2.a}] \cdot \mathbb{P}[\neg\mathsf{False}^2_{2.a}] \\
&\geqslant \mathbb{P}[\mathsf{Binding}^i_{2.a} |\ \neg\mathsf{False}^2_{2.a}] - \nu'(\kappa) && (10) \\
&\geqslant \mathbb{P}[\underbrace{\exists k \in \{1, 2\} : j_k = i^* \wedge \alpha^*_{k,i} \neq 0}_{\mathsf{Hit}^i_{2.a}}] - \nu'(\kappa), && (11)
\end{aligned}$$

where i^* is the index corresponding to the binding verification key ϑ_{i^*}. Eq. (10) follows by true-simulation extractability of the NIZK argument system.[9] Eq. (11) follows by the fact that, given that the extracted witness $(\alpha^*_{1,i}, \alpha^*_{2,i})$ is valid, the key ϑ' is binding only if the event $\mathsf{Hit}^i_{2.a}$ is verified.

One observes that $\mathbb{P}[\mathsf{Hit}^i_{2.a}] = \mathbb{P}[\mathsf{Hit}^i_{1.a}]$. Furthermore, by the hybridness property of the commitment scheme, there exists a negligible function $\nu'' : \mathbb{N} \to [0,1]$ such that

$$\begin{aligned}\mathbb{P}[\mathsf{Hit}^i_{1.a}] &\geqslant \mathbb{P}_{\substack{\mathcal{H}_0 \\ i^* \leftarrow\$ [t]}}[\exists k \in \{1,2\} : j_k = i^* \wedge \alpha^*_{k,i} \neq 0] - \nu''(\kappa) \\ &\geq \frac{1}{2}\left(\frac{1}{t} + \frac{1}{t-1}\right) - \nu''(\kappa) \geqslant \frac{1}{t} - \nu''(\kappa).\end{aligned}$$

where the last inequality follows by the fact that the random index i^* is sampled independently of the randomness used in hybrid $\mathcal{H}_0$. This concludes the proof sketch. □

5 A Scheme without Erasures

In this section we construct a signature scheme that is fully-leakage one-more unforgeable, without assuming secure erasures (cf. Definition 3.1). The drawback of the scheme is that the graceful degradation provided is not tight, as the slack parameter γ depends linearly on the number q_s of signature queries asked by the adversary. As mentioned in Section 3.1, since q_s is polynomial in the security parameter, this flavour of one-more unforgeability is still suitable for some applications.

Our scheme, and its security analysis, is similar to the constructions of Section 4. One additional difficulty is that we cannot distinguish anymore between leaky and leak-free signature queries, as now a leakage query might depend on the entire state of the signer (which is never erased).

To deal with this difficulty, we define a new simulation strategy that only rarely requires to access the signing key in order to simulate signature queries; moreover the probability of accessing the signing key can be controlled by the simulator, such that, on average, the conditional min-entropy of the signing key remains high. This feature allows the simulation to go trough.

Before coming to the scheme, we abstract away a special type of hybrid commitment scheme whose properties will turn useful in the security proof. We call this tool a *secret sharing* hybrid commitment, and we define it in Section 5.1. The signature scheme is described in Section 5.2, and its security analysis can be found in Section 5.3.

5.1 *Secret Sharing* Hybrid Commitment

Let $\mathbb{F}$ be a finite field. For parameters $\mu, \nu \in \mathbb{N}$, let $\mathcal{COM} = (\mathsf{Setup}, \mathsf{Com}, \mathsf{ESetup}, \mathsf{ECom}, \mathsf{Equiv})$ be a hybrid, linearly homomorphic commitment scheme with message space equal to $\mathbb{F}^\mu$ and randomness space equal to $\mathbb{F}^\nu$. Fix parameters $p := \log \kappa$ and $t := \kappa$, for security parameter $\kappa \in \mathbb{N}$. A *secret sharing* hybrid commitment scheme is a tuple of algorithms $\mathcal{COM}_{ss} = (\mathsf{Setup}_{ss}, \mathsf{Com}_{ss}, \mathsf{ESetup}_{ss}, \mathsf{ECom}_{ss}, \mathsf{Equiv}_{ss})$ specified as follow.

Algorithm $\mathsf{Setup}_{ss}(1^\kappa)$**:** For all $i \in [t]$ run $\vartheta_i \leftarrow\$ \mathsf{Setup}(1^\kappa)$ and output the verification key $\bar{\vartheta} := (\vartheta_1, \ldots, \vartheta_t)$.

Algorithm $\mathsf{ESetup}_{ss}(1^\kappa)$**:** For all $i \in [t]$ generate $(\vartheta_i, \tau_i) \leftarrow\$ \mathsf{ESetup}(1^\kappa)$ and output the verification key $\bar{\vartheta} := (\vartheta_1, \ldots, \vartheta_t)$ and the trapdoor key $\bar{\tau} := (\tau_1, \ldots, \tau_t)$.

[9] The forged messages were not queried to the signature oracle which means that for each forged signature (m^*, σ^*) no simulated argument was given for the pair $((\vartheta'^*, \vartheta_{j_1^*}, \vartheta_{j_2^*}), m^*)$.

Algorithm $\mathsf{Com}_{ss}(\bar{\vartheta}, m)$**:** Given a verification key $\bar{\vartheta}$ and a message $m \in \mathbb{F}^{\mu}$ do the following:

1. Choose random shares $s_1, \ldots, s_{p-1} \leftarrow\!\!\$\ \mathbb{F}^{\mu}$ and set $s_p := m - \sum_{i=1}^{p-1} s_i$;
2. Choose indexes $\mathbf{j} := (j_1, \ldots, j_p) \leftarrow\!\!\$\ [t]^p$ and for each $i \in [p]$ compute $com_i \leftarrow\!\!\$\ \mathsf{Com}(\vartheta_{j_i}, s_i; r_i)$ where $r_i \leftarrow\!\!\$\ \mathbb{F}^{\nu}$;
3. Output $c\bar{o}m := (\mathbf{j}, \mathbf{com})$ where $\mathbf{com} = (com_1, \ldots, com_p)$, and define the randomness (i.e., the opening) as $\bar{r} := (\mathbf{j}, \mathbf{s}, \mathbf{r})$ where $\mathbf{s} = (s_1, \ldots, s_{p-1})$ and $\mathbf{r} = (r_1, \ldots, r_p)$.

Algorithm $\mathsf{ECom}_{ss}(\bar{\vartheta}, \bar{\tau})$**:** Given a verification key $\bar{\vartheta}$ and the trapdoor $\bar{\tau}$, do the following:

1. Choose indexes $\mathbf{j} = (j_1, \ldots, j_p) \leftarrow\!\!\$\ [t]^p$;
2. Run $(com_p, r'_p) \leftarrow\!\!\$\ \mathsf{ECom}(\vartheta_{j_p}, \tau_{j_p})$ and, for each $i \in [p-1]$, compute $com_i \leftarrow\!\!\$\ \mathsf{Com}(\vartheta_{j_i}, s_i; r_i)$ where $s_i \in \mathbb{F}^{\mu}$ and $r_i \leftarrow\!\!\$\ \mathbb{F}^{\nu}$;
3. Output commitment $c\bar{o}m := (\mathbf{j}, \mathbf{com})$ where $\mathbf{com} = (com_1, \ldots, com_p)$, and randomness $\bar{r}' := (\mathbf{j}, \mathbf{s}, \mathbf{r}')$ where $\mathbf{s} = (s_1, \ldots, s_{p-1})$ and $\mathbf{r}' = (r_1, \ldots, r_{p-1}, r'_p)$.

Algorithm $\mathsf{Equiv}_{ss}(\bar{\tau}, m, \bar{r}')$**:** Given a trapdoor $\bar{\tau}$, a message m and randomness $\bar{r}'$, do the following:

1. Parse the randomness as $\bar{r}' = (\mathbf{j}, \mathbf{s}, \mathbf{r}')$, where $\mathbf{r}' = (r_1, \ldots, r_{p-1}, r'_p)$, and the trapdoor as $\bar{\tau} = (\tau_1, \ldots, \tau_t)$;
2. Compute $s_p := m - \sum_{i=1}^{p-1} s_i$, and run $r_p := \mathsf{Equiv}(\vartheta_{j_p}, s_p, r'_p)$;
3. Output randomness $\bar{r} := (\mathbf{j}, \mathbf{s}, \mathbf{r})$, for $\mathbf{r} = (r_1, \ldots, r_{p-1}, r_p)$.

It is easy to see that $\mathcal{COM}_{ss}$ is a hybrid linearly homomorphic commitment scheme with message space $\mathbb{F}^{\mu}$ and randomness space equal to $\mathcal{R}_{ss} := [t]^p \times \mathbb{F}^{(p-1)\mu} \times \mathbb{F}^{p\nu}$.

Probable binding commitment. Let $\mathcal{COM}_{ss}$ be a secret sharing hybrid commitment. We equip $\mathcal{COM}_{ss}$ with a special tuple of algorithms that allow, once instantiated the verification key, to generate commitments that are perfectly binding with some fixed probability. The algorithms are specified below.

Algorithm $\mathsf{pbSetup}(1^{\kappa}, c)$**:** Given the security parameter κ, and a constant $c \in \mathbb{N}$ such that $1 \leqslant c \leqslant \log \kappa$, output a verification key $\bar{\vartheta} := (\vartheta_1, \ldots, \vartheta_t)$ and a trapdoor $\bar{\tau} := (\tau_1, \ldots, \tau_t)$ defined as follows:

1. Choose I^* random subset of $[t]$ with cardinality $n := 2^{-c}t$ and for all $i \in [t]$ let:

$$\begin{cases} \vartheta_i \leftarrow\!\!\$\ \mathsf{Setup}(1^{\kappa}) & \text{if } i \in I^* \\ (\vartheta_i, \tau_i) \leftarrow\!\!\$\ \mathsf{ESetup}(1^{\kappa}) & \text{else;} \end{cases}$$

2. Let $\bar{\tau} := \left(I^*, \{\tau_i\}_{i \in [t] \setminus I^*}\right)$.

Algorithm $\mathsf{pbECom}(\bar{\vartheta}, \bar{\tau})$**:** Choose indexes $\mathbf{j} := (j_1, \ldots, j_p) \leftarrow\!\!\$\ [t]^p$; if $\mathbf{j}$ is a subset of I^* return $\perp$. Otherwise w.l.o.g. let j_p be the index not in I^*; the algorithm is identical to ECom_{ss}.

Algorithm $\mathsf{pbEquiv}(\bar{\tau}, m, \bar{r}')$**:** Parse $\bar{r}' = (\mathbf{j}, \mathbf{s}, \mathbf{r}')$; in case $\mathbf{j}$ is a subset of I^* return $\perp$. Otherwise the algorithm is identical to Equiv_{ss}.

Note that whenever pbECom and $\mathsf{pbEquiv}$ do not return $\perp$, the trapdoor key $\bar{\tau}$ contains the trapdoor information τ_{j_p} which is needed by the algorithms ECom_{ss} and Equiv_{ss}.

We say that a commitment $c\bar{o}m = (\mathbf{j}, \mathbf{com})$ is *binding* iff $\mathbf{j} \subseteq I^*$.

Lemma 5.1. *Let* $\mathcal{COM}_{ss}$ *and* $(\mathsf{pbSetup}, \mathsf{pbECom}, \mathsf{pbEquiv})$ *be defined as above. For any* $c \in \mathbb{N}$ *such that* $0 < c \leqslant \log \kappa$ *the following holds.*

(a) The following probability distributions are computationally indistinguishable:

$$\left\{\bar{\vartheta} |\ (\bar{\vartheta}, \bar{\tau}) \leftarrow\!\!\$\ \mathsf{pbSetup}(1^\kappa)\right\} \quad \textit{and} \quad \left\{\bar{\vartheta} |\ \bar{\vartheta} \leftarrow\!\!\$\ \mathsf{Setup}_{ss}(1^\kappa)\right\}.$$

(b) For all $(\bar{\vartheta}, \bar{\tau}) \leftarrow\!\!\$\ \mathsf{pbSetup}(1^\kappa)$ and for all $m \in \mathbb{F}^\mu$ the following probability distributions are statistically indistinguishable:

$$\left\{ (c\bar{o}m, \bar{r}) \;\middle|\; \begin{array}{rcl} \bar{r} & \leftarrow\!\!\$ & \mathcal{R}_{ss}, \\ c\bar{o}m & := & \mathsf{Com}_{ss}(\bar{\vartheta}, m; \bar{r}), \\ \multicolumn{3}{c}{c\bar{o}m \textit{ is not binding}} \end{array} \right\}$$

$$\left\{ (c\bar{o}m, \bar{r}) \;\middle|\; \begin{array}{rcl} (c\bar{o}m, \bar{r}') & \leftarrow\!\!\$ & \mathsf{pbECom}(\bar{\vartheta}, \bar{\tau}), \\ \bar{r} & := & \mathsf{pbEquiv}(\bar{\tau}, m, \bar{r}') \\ \multicolumn{3}{c}{\mathsf{pbECom} \textit{ doesn't return } \perp} \end{array} \right\}$$

(c) For all PPT adversaries A there exists a negligible function $\nu : \mathbb{N} \to [0,1]$ such that the following holds:

$$\left| \mathbb{P}\left[c\bar{o}m \text{ is binding } s.t. \begin{array}{l} (\bar{\vartheta}, \bar{\tau}) \leftarrow\!\!\$\ \mathsf{pbSetup}(1^\kappa, c); \\ (c\bar{o}m, m, \bar{r}) \leftarrow\!\!\$\ \mathsf{A}(\bar{\vartheta}); \\ c\bar{o}m = \mathsf{Com}_{ss}(\bar{\vartheta}, m, \bar{r}) \end{array} \right] - \frac{1}{\kappa^c} \right| \leqslant \nu(\kappa).$$

(d) For any $(\bar{\vartheta}, \bar{\tau}) \leftarrow\!\!\$\ \mathsf{pbSetup}(1^\kappa, c)$ the probability over the randomness $\bar{r}'$ that $\mathsf{pbECom}(\bar{\vartheta}; \bar{r}')$ aborts is $1/\kappa^c$.

Proof. Properties (a) and (b) follow, respectively, from the hybridness and the trapdoor hiding property of $\mathcal{COM}_{ss}$. To prove property (c), consider the following probability:

$$\mathbb{P}\left[c\bar{o}m \text{ is binding } s.t. \begin{array}{l} \bar{\vartheta} \leftarrow\!\!\$\ \mathsf{Setup}_{ss}(1^\kappa),\ I^* \leftarrow\!\!\$\ \binom{[t]}{n}; \\ (c\bar{o}m, m, \bar{r}) \leftarrow\!\!\$\ \mathsf{A}(\bar{\vartheta}); \\ c\bar{o}m = \mathsf{Com}_{ss}(\bar{\vartheta}, m; \bar{r}) \end{array} \right].$$

Recall that $c\bar{o}m$ is binding iff $\mathbf{j} \subseteq I^*$. Since I^* and the verification key $\bar{\vartheta}$ are uniformly and independently sampled, the above probability is upper bounded by

$$\binom{n}{t}^p \;=\; 2^{-c \log \kappa} \;=\; \frac{1}{\kappa^c}. \tag{12}$$

Property (c) easily follows by the previous equation and Property (a).

Property (d) follows directly from Eq. (12), as the algorithm ECom_{ss} chooses the indexes $\mathbf{j}$ uniformly at random. □

5.2 The Signature Scheme

Let $\mathcal{COM} := \mathcal{COM}_1$ be a trapdoor hiding, linearly homomorphic commitment scheme (like the one used in the construction from Section 4.1). Let $\mathcal{COM}_2$ be a hybrid, linearly homomorphic commitment scheme, and $\mathcal{COM}_{ss}$ be the corresponding secret sharing hybrid commitment as described in Section 5.1. We modify the scheme $\mathcal{SS}_1$ from Section 4.1 in two ways: (i) we strip off the OR-construction; and (ii) we replace the commitment *com* with a secret sharing hybrid commitment. Our scheme $\mathcal{SS}_3 = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ has message space equal to $\mathbb{F}$ and is described below:

Key Generation. Let $t, d, \mu \in \mathbb{N}$ be parameters. Run $\mathsf{crs} \leftarrow\!\!\$\ \mathsf{Init}(1^\kappa)$, sample $\vartheta \leftarrow\!\!\$\ \mathsf{Setup}(1^\kappa)$, and $\bar{\vartheta} := (\vartheta_1, \ldots, \vartheta_t) \leftarrow\!\!\$\ \mathsf{Setup}_{ss}(1^\kappa)$. Sample $\boldsymbol{\Delta} \leftarrow\!\!\$\ (\mathbb{F}^\mu)^{d+1}$ and $\mathbf{r} = (r_0, \ldots, r_d) \leftarrow\!\!\$\ \mathbb{F}^{d+1}$, and compute commitments $com_i = \mathsf{Com}(\vartheta, \boldsymbol{\delta}_i; r_i)$ for $i \in [0, d]$, where $\boldsymbol{\delta}_i \in \mathbb{F}^\mu$ is the i-th column of $\boldsymbol{\Delta}$. Output

$$sk = (\boldsymbol{\Delta}, \mathbf{r}) \qquad vk = (\mathsf{crs}, \vartheta, \{\vartheta_i\}_{i=1}^t, \{com_i\}_{i=0}^d).$$

Signature. For $j \in [\mu]$, let $\delta_j(X)$, $r(X)$, and $\boldsymbol{\Delta}(X)$ be defined as in the scheme from Section 4.1. Consider the following polynomial-time relation:

$$\mathfrak{R} := \left\{ (\vartheta, \bar{\vartheta}, \tilde{com}, \bar{com}); (\tilde{m}, \tilde{r}, \bar{r}) \;\middle|\; \begin{array}{c} com = \mathsf{Com}(\vartheta, \tilde{m}; \tilde{r}) \\ \bar{com} = \mathsf{Com}_{ss}(\bar{\vartheta}, \tilde{m}; \bar{r}) \end{array} \right\}.$$

To sign a message $m \in \mathbb{F}$ compute $\tilde{m} = \boldsymbol{\Delta}(m)$ and $\tilde{r} = r(m)$, and let $\tilde{com} := \mathsf{Com}(\vartheta, \tilde{m}; \tilde{r})$ and $\bar{com} := \mathsf{Com}_{ss}(\bar{\vartheta}, \tilde{m}; \bar{r})$ where $\bar{r} \leftarrow\!\!\$\ \mathcal{R}_{ss}$. Using crs as common reference string, generate a NIWI argument π for $(\vartheta, \bar{\vartheta}, \tilde{com}, com) \in \mathfrak{L}_{\mathfrak{R}}$, the language generated by the above relation $\mathfrak{R}$. Output $\sigma = (\pi, \bar{com})$.

Verification. Given a pair (m, σ), parse σ as $\sigma = (\pi, \bar{com})$. Output the same as:

$$\mathsf{Ver}(\mathsf{crs}, \pi, (\vartheta, \bar{\vartheta}, \tilde{com}, \bar{com})).$$

Theorem 5.1. *Let $\mu \in \mathbb{N}$, and let $\mathbb{F}$ be a finite field of size $\log |\mathbb{F}| = \kappa$ for security parameter $\kappa \in \mathbb{N}$. For any constant $0 < \xi \leqslant \mu/(\mu+1)$ and $t = O(\kappa)$ the above signature scheme is $((\frac{\mu}{\mu+1} - \xi)|sk|, q_s, O(\frac{1}{q_s \cdot \log \kappa}), \mathsf{negl}(\kappa))$-fully-leakage one-more unforgeable.*

Notice that, when μ grows, the leakage rate asymptotically goes to 1. The slack parameter $\gamma(\kappa, q_s)$ is linear in the number of signature queries, but still polynomial in the security parameter as $q_s = \mathsf{poly}(\kappa)$. As shown in [55, Section 5] this is good enough for some applications.

5.3 Proof Sketch

The proof of Theorem 5.1 requires an hybrid argument, along the same lines of the proof of Theorem 4.1. Since many of the hybrids are identical, we only describe the steps that are different in full details.

Let A be an adversary asking $q_s = \kappa^c$ (for some constant c) signature queries, such that

$$\mathbb{P}[\mathbf{Exp}^{\text{one-more}}_{\mathcal{SS}_3, \mathsf{A}}(\kappa, \ell, q_s, \gamma) = 1] = \varepsilon(\kappa).$$

Experiment $\mathcal{H}_0$: This experiment is exactly the same as the original experiment defining fully-leakage one-more unforgeability of $\mathcal{SS}_3$. Note that the initial state (after key generation), is equal to $st := ((\boldsymbol{\Delta}, \mathbf{r}), \mathsf{crs}, \vartheta, \bar{\vartheta})$. (Recall that we assume oblivious sampling for both the NIWI and the commitment scheme.) At each invocation of the signing oracle $\mathsf{Sign}(sk, \cdot)$ upon input a message m, the experiment produces a signature $\sigma = (\pi, \bar{com})$ following the signing algorithm, and appends the randomness used to generate σ to the current state, i.e. $st := st \cup (\bar{r}, r_{niwi})$, where $\bar{r}$ is the randomness used to generate the commitment $\bar{com}$ and r_{niwi} is the randomness used to generate the argument π.

Denote with $((m_1^*, (\pi_1^*, com_1^*)), \ldots, (m_n^*, (\pi_n^*, com_n^*)))$ the set of forgeries returned by the adversary at the end of the experiment. Let Forge_0 and False_0 be defined as in the proof of Theorem 4.1, and define $\varepsilon_0 := \mathbb{P}[\mathsf{Forge}_0 \wedge \neg \mathsf{False}_0]$.

Claim 5.1. $\varepsilon - \varepsilon_0 \in \mathsf{negl}(\kappa)$.

Proof. The proof is similar to the proof of Claim 4.1 and is therefore omitted. □

For each hybrid $\mathcal{H}_i$, described below, we write ε_i for the probability of the event $\mathsf{Forge}_i \wedge \neg\mathsf{False}_i$, and View_i for the view of the adversary at the end of the experiment.

Experiment $\mathcal{H}_{1.a}$: This experiment is the same as $\mathcal{H}_0$, except that the verification key is computed using the probable binding key generation. Namely, we run $(\bar{\vartheta}, \bar{\tau}) \leftarrow\!\!\$\ \mathsf{pbSetup}(1^\kappa, c)$ (recall that the number of signature queries made by A is $q_s = \kappa^c$). Note that from the point of view of the adversary, the initial state remains equal to $st := ((\mathbf{\Delta}, \mathbf{r}), \mathsf{crs}, \vartheta, \bar{\vartheta})$.

Claim 5.2. $\varepsilon_0 - \varepsilon_{1.a} \in \mathsf{negl}(\kappa)$.

Proof. Follows readily from Property (a) of Lemma 5.1. □

Experiment $\mathcal{H}_{1.b}$: Identical to the same hybrid in the proof of Theorem 4.1.

Claim 5.3. $\varepsilon_{1.a} - \varepsilon_{1.b} \in \mathsf{negl}(\kappa)$.

Proof. Similar to the proof of Claim 4.3, and therefore omitted. □

Experiment $\mathcal{H}_{2.d}$: In this experiment we modify the way signature queries are answered. Given such a query m, the experiment runs $\mathsf{pbECom}(\bar{\vartheta})$; in case the algorithm returns $\perp$ the query is answered as before. Otherwise let $(c\bar{o}m, \bar{r}')$ be the output of $\mathsf{pbECom}(\bar{\vartheta})$; the answer to the signature query uses $c\bar{o}m$ as commitment. Observe that in the latter case the randomness $\bar{r}$ used to compute $c\bar{o}m$ in the previous hybrid can be obtained as a function of the matrix $\mathbf{\Delta}$ by running the equivocation algorithm. In particular:

$$\bar{r}(\mathbf{\Delta}) := \begin{cases} \bar{r} \leftarrow\!\!\$\ \mathcal{R}_{ss} & \text{if } \mathsf{ECom}_{ss}(\bar{\vartheta}) \text{ returns } \perp \\ \mathsf{Equiv}_{ss}(\bar{\tau}, \mathbf{\Delta}(m), \bar{r}') & \text{otherwise.} \end{cases}$$

The state is set to $st := st \cup \{(\mathbf{\Delta}, \mathbf{r}), \bar{r}(\mathbf{\Delta}), r_{niwi}\}$, where r_{niwi} is the randomness used to generate the NIWI argument π.

Claim 5.4. $\varepsilon_{2.b} - \varepsilon_{2.d} \in \mathsf{negl}(\kappa)$.

Proof. Follows readily from Property (b) of Lemma 5.1. □

Experiment $\mathcal{H}_{2.e}$: This experiment is identical to the previous hybrid, except that (whenever possible) it uses a different witness to compute the argument corresponding to a signature query. In particular, given such a query $m \in \mathbb{F}$ for which $\mathsf{pbECom}(\bar{\vartheta})$ does not return $\perp$, it generates the argument π by running

$$\mathsf{Prove}(\mathsf{crs}, \underbrace{(\vartheta, \bar{\vartheta}, c\tilde{o}m, c\bar{o}m)}_{\alpha}, \underbrace{(0, \mathsf{Equiv}(\tau, 0^\mu, r'(m)), \mathsf{pbEquiv}(\bar{\tau}, 0^\mu, \bar{r}')}_{\beta'}; r'_{niwi}),$$

where $r'(m) = \sum_{i=0}^{d} r'_i \cdot m^i$ is computed using the randomness $\{r'_i\}_{i=0}^{d}$. Notice that the randomness r_{niwi} used to generate the NIWI argument in the previous experiment can be sampled (inefficiently) as a function of the witness $\beta := (\mathbf{\Delta}(m), \mathsf{Equiv}(\tau, \mathbf{\Delta}(m), r'(m))$, $\mathsf{pbEquiv}(\bar{\tau}, \mathbf{\Delta}(m), \bar{r}')$ and r'_{niwi}. In particular:

$$r_{niwi}(\mathbf{\Delta}) := \begin{cases} r_{niwi} \text{ (as sampled by the prover)} & \text{if } \mathsf{pbECom}(\bar{\vartheta}) \text{ returns } \perp \\ \mathsf{Rand}(\pi, r'_{niwi}, \beta) & \text{otherwise,} \end{cases}$$

where Rand is the reconstruction algorithm of Lemma 4.1. The state is updated as in $st(\mathbf{\Delta}) = st \cup \{r(\mathbf{\Delta}), r_{niwi}(\mathbf{\Delta})\}$.

Claim 5.5. $\varepsilon_{2.d} - \varepsilon_{2.e} \in \mathsf{negl}(\kappa)$.

Proof. Similar to the proof of Claim 4.9, and therefore omitted. □

Experiment $\mathcal{H}_3^{\mathcal{O}_{\mathbf{\Delta}}(\cdot)}$: This experiment is the same as the previous hybrid, with the difference that $\mathbf{\Delta}$ is not sampled by the hybrid as part of the signing key, but can instead be accessed via $\mathcal{O}_{\mathbf{\Delta}}(\cdot)$. Note that each signature query $m \in \mathbb{F}$ for which $\mathsf{pbECom}(\bar{\vartheta})$ happens to not return $\perp$ can be answered without knowing $\mathbf{\Delta}$. Moreover:

- For each signature query $m \in \mathbb{F}$ where $\mathsf{pbECom}(\bar{\vartheta})$ returns $\perp$, the experiment defines the leakage function $f'_m := \mathbf{\Delta}(m)$ and queries $\mathcal{O}_{\mathbf{\Delta}}(\cdot)$ on f'. We call such query a *bad* query and we let Bad denote the corresponding event. Given the value $\mathbf{\Delta}(m)$, the signature σ can be computed as in $\mathcal{H}_{2.e}$.
- For each leakage query f the experiment defines a function f'' and queries $\mathcal{O}_{\mathbf{\Delta}}(\cdot)$ on f''. The function f'' hard-wires all values necessary to reconstruct the current state $st = st(\mathbf{\Delta})$; this includes the trapdoor information needed in order to reconstruct $\mathbf{r}$, and all random coins used to simulate previous signature queries.

Claim 5.6. $\varepsilon_3 = \varepsilon_{2.e}$.

Proof. The only difference between the two hybrids stems from the fact that $\mathbf{\Delta}$ is only accessed via a leakage oracle in $\mathcal{H}_3$. Note that in $\mathcal{H}_{2.e}$ the matrix $\mathbf{\Delta}$ is needed only to answer bad signature queries and leakage queries. In case of a bad query, $\mathcal{H}_3$ accesses the leakage oracle to additionally obtain $\mathbf{\Delta}(m)$ via f'_m; given $\mathbf{\Delta}(m)$ the signature computation is identical in the two hybrids. The distribution of the leakage is also identical in the two hybrids, as the function f'' perfectly reconstructs the current state $st(\mathbf{\Delta})$. □

Lemma 5.2. *For any $\beta > 0$ we have that*

$$\mathbb{P}\left[\mathbb{H}_\infty\left(\mathbf{\Delta} \middle|\ \mathsf{View}_3 = v\right) \geqslant |\mathbf{\Delta}| - 2e\mu \log|\mathbb{F}| - \beta - \lambda\right] \geqslant 2^{-\beta} + 2^{-2e}, \tag{13}$$

where the probability is taken over the randomness of the experiment.

Proof. The proof is a straightforward adaptation of the proof of Lemma 4.2. The only difference is in the computation of the probability of the events Bad_i, which is now upper bounded by $1/\kappa^c = 1/q_s$ by Property (d) of Lemma 5.1. Note that this gives $\mathbb{E}[Z] = 1$. □

Lemma 5.3. *For any adversary A such that $\mathbb{P}[\mathbf{Exp}^{\text{one-more}}_{\mathcal{SS}_1,\mathsf{A}}(\kappa,\ell,q_s,\gamma) = 1] = \varepsilon$, there exists a constant $c' > 0$ such that:*

$$\mathbb{E}\left[\mathbb{H}_\infty\left(\mathbf{\Delta} \middle|\ \mathsf{View}_3 = v\right)\right] \leqslant \left(d + 1 - \frac{n}{q_s}\right)\mu \log|\mathbb{F}| - c' \log \varepsilon(\kappa),$$

where the expectation is taken over the randomness of the experiment.

Proof. The proof is a straightforward adaptation of the proof of Lemma 4.3. The only difference is in the computation of $\mathbb{E}[X + Z]$. Applying Property (c) and Property (d) of Lemma 4.2, we get $\mathbb{E}[X + Z] \geq n(1/q_s - \mathsf{negl}(\kappa)) + 1 \geq n/q_s$. □

Given the two lemmas above the proof of Theorem 5.1 can be derived following the same strategy as for the proof of Theorem 4.1. In particular, one can verify that to obtain a contradiction we need that:

$$\frac{n\kappa}{q_s} = \Omega(\lambda - \log \varepsilon(\kappa)). \tag{14}$$

Whenever $\varepsilon(\kappa)$ is noticeable, since $n \geq \lfloor \frac{\lambda}{\gamma \cdot s} \rfloor + 1$, and $s = \Theta(\kappa \log \kappa)$, in order for Eq. (14) to hold it suffices to set $\gamma = O(\frac{1}{q_s \cdot \log \kappa})$. This concludes the proof sketch. □

6 A Scheme in the Bounded-Retrieval Model

In this section we describe a fully-leakage one-more unforgeable signature scheme in the Bounded Retrieval Model (BRM) [17, 29, 30]. The BRM imposes additional efficiency constraints on the size of the public key and the computational time of the signer and verifier algorithm. The scheme is in the random oracle model [4]. To define fully-leakage one-more unforgeability in the BRM, we let the key generation, signature, and verification algorithms additionally be parametrized by the leakage parameter ℓ.

Definition 6.1 (Fully-leakage one-more unforgeability in the BRM)**.** *A signature scheme* $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ *is* (ℓ, γ)-fully-leakage one-more unforgeable *in the BRM if the following conditions are met:*

- *The verification key size, signature size, signing-time and verification-time (and the number of secret-key bits read during signature computation) are independent of the leakage-bound* ℓ*. More formally, there exist polynomials* vksize, sgsize, sigT, verT, *such that, for any polynomial* ℓ*, and any* $(vk, sk) \leftarrow_\$ \mathsf{KGen}(1^\kappa, 1^{\ell(\kappa)})$, $m \in \mathcal{M}$, $\sigma \leftarrow_\$ \mathsf{Sign}(sk, m)$*, we have that:*
  - *Verification key size is* $|vk| = O(\mathsf{vksize}(\kappa))$*, signature size is* $|\sigma| = O(\mathsf{sgsize}(\kappa, |m|))$.
  - *Run-time of* $\mathsf{Sign}(sk, m)$ *and the number of bits of* sk *accessed is* $O(\mathsf{sigT}(\kappa, |m|))$.
  - *Run-time of* $\mathsf{Verify}(vk, m, \sigma)$ *is* $O(\mathsf{verT}(\kappa, |m|))$.
- *The scheme* $\mathcal{SS}$ *is* (ℓ, γ)*-fully-leakage one-more unforgeable according to Definition 3.1.*

6.1 Ingredients

Let $\mathbb{F}$ be a finite field with additive identity 0. For two parameter d and t, with $d > t$, let us define a distribution $\mathsf{Sparse}(\mathbb{F}, d, t)$ by sampling a vector $\mathbf{c} = (c_1, \ldots, c_d)$ such that for all $i \in [d]$: (i) $c_i \leftarrow_\$ \mathbb{F}$ with probability t/d, and (ii) $c_i = 0$ with probability $1 - t/d$. We call t the locality parameter.

We will rely on the following lemma.

Lemma 6.1. *Let* $\kappa = \log |\mathbb{F}|$*, for security parameter* $\kappa \in \mathbb{N}$*. For any subspace* $A \subset \mathbb{F}^d$ *of dimension less then* d *the probability that a vector* $\mathbf{c}$ *randomly chosen over* $\mathsf{Sparse}(\mathbb{F}, d, t)$ *lies in the subspace* A *is at most* $\frac{1}{|\mathbb{F}|} + e^{-t}$*.*

Proof. Note that the bigger A is, the more likely $\mathbf{c} \in A$, therefore the worst case is $dim(A) = d - 1$. If $\mathbf{c} \in A$ then there exist $\alpha_1, \ldots, \alpha_d \in \mathbb{F}$ with at least one coordinate i s.t. $\alpha_i \neq 0$, such that $\mathbf{c} \in A$ if and only if $E_d = 1$ where E_d is the event defined as

$$E_d := \begin{cases} 1 & \text{if } \sum_{i=0}^{d} \alpha_i \cdot c_i = 0 \\ 0 & \text{otherwise.} \end{cases}$$

For any $d' \leq d$ we have:

$$\begin{aligned} \mathbb{P}[E_{d'}] &\leqslant \sum_{x \in \mathbb{F}} \mathbb{P}[c_{d'} = x]\, \mathbb{P}\left[\sum_{i=1}^{d'-1} \alpha_i \cdot c_i = -\alpha_{d'} \cdot x\right] \\ &\leqslant \mathbb{P}[c_{d'} = 0]\, \mathbb{P}[E_{d'-1}] + \sum_{x \in \mathbb{F} \setminus \{0\}} \mathbb{P}[c_{d'} = x] \cdot \mathbb{P}\left[\sum_{i=1}^{d'-1} \alpha_i \cdot c_i = -\alpha_{d'} \cdot x\right] \end{aligned}$$

$$\leqslant \left(\frac{t\cdot 2^{-\kappa}}{d}+1-\frac{t}{d}\right)\mathbb{P}\left[E_{d'-1}\right]+\frac{t\cdot 2^{-\kappa}}{d}\cdot\sum_{x\in\mathbb{F}\setminus\{0\}}\mathbb{P}\left[\sum_{i=1}^{d'-1}\alpha_i\cdot c_i=-\alpha_{d'}\cdot x\right]$$
$$\leqslant \left(\frac{t\cdot 2^{-\kappa}}{d}+1-\frac{t}{d}\right)\mathbb{P}\left[E_{d'-1}\right]+\frac{t\cdot 2^{-\kappa}}{d}\cdot\left(1-\mathbb{P}\left[E_{d'-1}\right]\right)$$
$$\leqslant \left(1-\frac{t}{d}\right)\mathbb{P}\left[E_{d'-1}\right]+\frac{t\cdot 2^{-\kappa}}{d}.$$

From the last inequality, by setting $d' := d$, it follows that

$$\begin{aligned}\mathbb{P}\left[E_d\right] &\leqslant \left(1-\frac{t}{d}\right)^d\mathbb{P}\left[E_0\right]+\frac{t\cdot 2^{-\kappa}}{d}\left(\sum_{i=0}^{d-1}\left(1-\frac{t}{d}\right)^i\right)\\ &\leqslant e^{-t}\cdot\mathbb{P}\left[E_0\right]+\frac{t\cdot 2^{-\kappa}}{d}\left(\frac{1-(1-\frac{t}{d})^d}{t/d}\right)\\ &\leqslant e^{-t}+\frac{1}{|\mathbb{F}|}.\end{aligned}$$

□

Our construction in the BRM is based on the following ingredients:

- Two random oracles $H_0 : \{0,1\}^* \to \{0,1\}^*$ and $H_1^{d,t} : \{0,1\}^* \to \mathsf{Sparse}(\mathbb{F}, d, t)$.[10]
- A perfectly hiding, linearly homomorphic commitment scheme $\mathcal{COM} = (\mathsf{Setup}, \mathsf{Com})$, with message space equal to $\mathbb{F}^\mu$ and randomness space equal to $\mathbb{F}$ (for a finite field $\mathbb{F}$ and a parameter $\mu \in \mathbb{N}$).
- A perfect zero-knowledge proof of knowledge system $\mathcal{NIZK} = (\mathsf{Prove}, \mathsf{Ver})$ with an online extractor (w.r.t. the random oracle H_0), for proving knowledge of the opening of a commitment.

We review the definition of a NIZK PoK with an online extractor below (see, e.g., [58, 36]).

Definition 6.2 (Online Extractability)**.** *We say that $\mathcal{NIZK} = (\mathsf{Prove}, \mathsf{Ver})$ is a NIZK PoK for a relation $\mathfrak{R}$ (w.r.t. a random oracle H) if the following holds:*

Perfect Zero-Knowledge. *There exist a pair of PPT algorithms $\mathsf{Sim} = (\mathsf{Sim}_0, \mathsf{Sim}_1)$ (a.k.a. the zero-knowledge simulator), such that for any pair of PPT distinguishers $\mathsf{D} = (\mathsf{D}_0, \mathsf{D}_1)$ the following distributions are identical:*

- *Let $(x, w, st_{\mathsf{D}}) \leftarrow\!\!\$\ \mathsf{D}_0^H(1^\kappa)$, and $\pi \leftarrow\!\!\$\ \mathsf{Prove}^H(x, w)$ if $(x, w) \in \mathfrak{R}$ and $\pi := \perp$ otherwise. Output $\mathsf{D}_1^H(\pi, st_{\mathsf{D}})$.*
- *Let $(H_0, st_{\mathsf{Sim}}) \leftarrow\!\!\$\ \mathsf{Sim}_0(1^\kappa)$, $(x, w, st_{\mathsf{D}}) \leftarrow\!\!\$\ D_0^{H_0}(1^\kappa)$, and $(H_1, \pi) \leftarrow\!\!\$\ \mathsf{Sim}_1(st_{\mathsf{Sim}}, x, \mathtt{yes})$ if $(x, w) \in \mathfrak{R}$ and $(H_1, \pi) \leftarrow\!\!\$\ \mathsf{Sim}_1(st_{\mathsf{Sim}}, x, \mathtt{no})$ otherwise. Output $\mathsf{D}_1^{H_1}(\pi, st_{\mathsf{D}})$.*

Online Extractability. *There exist a probabilistic polynomial-time algorithm K (a.k.a. the online extractor), such that the following holds for any algorithm A. Let $(x, \pi) \leftarrow\!\!\$\ \mathsf{A}^H(1^\kappa)$ and $\mathcal{Q}_H(\mathsf{A})$ be the sequence of queries of A to H and H's answers. Let $w \leftarrow\!\!\$\ \mathsf{K}(x, \pi, \mathcal{Q}_H(\mathsf{A}))$. Then, as a function of κ,*

$$\mathbb{P}\left[(x, w) \notin \mathfrak{R} \wedge \mathsf{Ver}^H(x, \pi) = 1\right] \in \mathsf{negl}(\kappa).$$

[10] The two random oracles can be derived by a single random oracle H, e.g., by letting $H_0 := H(0, \cdot)$ and $H_1 := H(1, \cdot)$.

6.2 Basic Construction

We first describe a scheme that meets partially the efficiency requirements of the BRM; namely, the signing and verification algorithms have running time independent of the leakage parameter ℓ, but the length of the verification key still depends on ℓ. Later we adopt the technique of Alwen *et al.* [2] to obtain a full-fledged scheme in the BRM.

Our scheme $\mathcal{SS}'_4 = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ has $\mathcal{M} = \{0,1\}^*$ and is described below:

Key Generation. Let $t, d, \mu \in \mathbb{N}$ be parameters; choose d to be large enough such that $\ell(\kappa) < d\mu \cdot \kappa$. Run $\vartheta \leftarrow\!\!\$\ \mathsf{Setup}(1^\kappa)$. Sample $\mathbf{\Delta} \leftarrow\!\!\$\ \mathbb{F}^{\mu \times d}$ and $\mathbf{r} = (r_1, \ldots, r_d) \leftarrow\!\!\$\ \mathbb{F}^d$, and compute commitments $com_i = \mathsf{Com}(\vartheta, \boldsymbol{\delta}_i; r_i)$ for $i \in [d]$, where $\boldsymbol{\delta}_i \in \mathbb{F}^\mu$ is the i-th column of $\mathbf{\Delta}$. Output

$$sk = (\mathbf{\Delta}, \mathbf{r}) \qquad vk = (\vartheta, \{com_i\}_{i=1}^d).$$

Signature. Consider the following polynomial-time relation:

$$\mathfrak{R} := \{(\vartheta, \tilde{com}); (\tilde{m}, \tilde{r}) \mid \tilde{com} = \mathsf{Com}(\vartheta, \tilde{m}; \tilde{r})\}.$$

To sign a message $m \in \{0,1\}^*$, let $\mathbf{c} := \mathcal{H}_1^{d,t}(m)$ and compute $\mathbf{\Delta}(m) := \mathbf{\Delta} \cdot \mathbf{c}^\mathsf{T}$ and $r(m) := \mathbf{r} \cdot \mathbf{c}^\mathsf{T}$. Generate a proof $\pi \leftarrow\!\!\$\ \mathsf{Prove}^{H_0}((\vartheta, \tilde{com}), (\mathbf{\Delta}(m), r(m))$, where $\tilde{com} := \mathsf{Com}(\vartheta, \mathbf{\Delta}(m); r(m))$. Output $\sigma := \pi$.

Verification. Given a pair (m, σ), parse σ as $\sigma = \pi$ and let $\mathbf{c} := H_1^{d,t}(m) = (c_1, \ldots, c_d)$. Output the same as:

$$\mathsf{Ver}^{H_0}(\pi, (\vartheta, \prod_{i=1}^{d} (com_i)^{c_i})).$$

Notice that for $t = O(\kappa)$ the efficiency constraint on the running time of signer and verifier are met only *on average*. If the signature scheme needs to be *always* local, the signer can just re-sign the message whenever the number of non-zero locations in $\mathbf{c}$ is not in the range $\{(1 \pm \varepsilon)t\}$, for a constant ε.

Theorem 6.1. *Let $\mathbb{F}$ be a finite field such that $\log|\mathbb{F}| = \kappa$, for security parameter $\kappa \in \mathbb{N}$. For any $\xi = \omega(\log \kappa)$, whenever $\gamma = O(1)$ the above signature scheme is $((\frac{\mu}{\mu+1})|sk| - \xi, \gamma)$-fully-leakage one-more unforgeable in the random oracle model.*

We reduce to the computational binding property of the commitment scheme. The reduction simulates the entire experiment by sampling a legitimate secret key $(\mathbf{\Delta}, \mathbf{r})$. Whenever the adversary outputs its forgeries $(m_1^*, \pi_1^*), \ldots, (m_n^*, \pi_n^*)$, the reduction chooses an index $i \leftarrow\!\!\$\ [n]$ uniformly at random and extracts the proof π_i^* obtaining a valid witness $(\tilde{m}_i^*, \tilde{r}_i^*)$. Finally the reduction outputs $(\tilde{com}_i, (\tilde{m}_i^*, \tilde{r}_i^*), (\mathbf{\Delta}(m_i^*), r(m_i^*))$.

It remains to argue that the conditional average min-entropy of $\mathbf{\Delta}$ given the view of the adversary is high. By a counting argument, this implies that there exists an index $i \in [n]$ for which the min-entropy of the witness sitting behind each of the forged signatures is larger than 1, so that the reduction breaks the computational binding property with noticeable probability.

Proof. Let A be an adversary in the fully-leakage one-more unforgeability experiment. Define with Forge the event that A makes the experiment output 1, and assume $\mathbb{P}[\mathsf{Forge}] = \varepsilon$.

Consider the following reduction algorithm B (based on A), attacking the computational binding property of $\mathcal{COM}$. Adversary B works as follows:

1. At the beginning B receives the public key $\vartheta \leftarrow\!\!\$\ \mathsf{Setup}(1^\kappa)$ for the commitment scheme. Hence, it generates a legitimate signing/verification key pair (sk, vk) for $\mathcal{SS}'_4$, where $sk = (\mathbf{\Delta}, \mathbf{r})$ and $vk = (\vartheta, \{com_i\}_{i=1}^d)$.

2. B takes care of A's queries to the random oracle, and keeps a list $\mathcal{Q}_H(\mathsf{A})$ of such queries and the relative answers.
3. Given the above setup, B answers signature queries and leakage queries asked by A in the natural way. Every time a signature is computed, the state is updated as $st := st \cup \{r_{nizk}\}$, where r_{nizk} is the randomness required to generate each signature.
4. Eventually A returns a set of forgeries $(m_1^*, \pi_1^*), \ldots, (m_n^*, \pi_n^*)$. At this point B runs $\mathsf{K}(\vartheta, \tilde{com}_i^*, \mathcal{Q}_H(\mathsf{A}))$ for all $i \in [n]$, where $\tilde{com}_i^* = \mathsf{Com}(\vartheta, \mathbf{\Delta}(m_i^*), r(m_i^*))$, obtaining a witness $(\tilde{m}_i^*, \tilde{r}_i^*)_{i\in[n]}$. In case there exists an index $i \in [n]$ such that $(\tilde{com}_i^*, (\tilde{m}_i^*, \tilde{r}_i^*)) \notin \mathfrak{R}$, the reduction B aborts.
5. B finds an index i such that $(\mathbf{\Delta}(m_i^*), r(m_i^*)) \neq (\tilde{m}_i^*, \tilde{r}_i^*)$. If no such index is found, B aborts. Otherwise B outputs $(\tilde{com}_i^*, (\tilde{m}_i^*, \tilde{r}_i^*), (\mathbf{\Delta}(m_i^*), r(m_i^*)))$.

We say that B *wins* if it breaks the computational binding property of $\mathcal{COM}$. Let Abort_1 (resp. Abort_2) be the event that B aborts in step 4 (resp. step 5); define $\mathsf{Abort} := \mathsf{Abort}_1 \vee \mathsf{Abort}_2$. By definition of B:

$$\begin{aligned}\mathbb{P}\left[\mathsf{B}\text{ wins}\right] &= \mathbb{P}[\mathsf{Forge} \wedge \neg\mathsf{Abort}_1 \wedge \neg\mathsf{Abort}_2] \\ &\geq \mathbb{P}[\mathsf{Forge} \wedge \neg\mathsf{Abort}_1] - \mathbb{P}[\mathsf{Abort}_2] \\ &\geq \mathbb{P}[\mathsf{Forge}] - \mathbb{P}[\mathsf{Forge} \wedge \mathsf{Abort}_1] - \mathbb{P}[\mathsf{Abort}_2].\end{aligned}$$

Claim 6.1. $\mathbb{P}[\mathsf{Forge} \wedge \mathsf{Abort}_1] \in \mathsf{negl}(\kappa)$.

Proof. Event Abort_1 holds true in case there is at least one index $i \in [n]$ such that K fails to extract a valid witness for $\tilde{com}_i^*$. However, since Forge occurs, all proofs π_i^* are valid and thus there is at least one index $i \in [n]$ for which:

$$\mathsf{Ver}(\vartheta, \tilde{com}_i^*, \pi_i^*) = 1 \wedge ((\vartheta, \tilde{com}_i^*), (\tilde{m}_i^*, \tilde{r}_i^*)) \notin \mathfrak{R},$$

which would contradict simulation extractability of the NIZK argument system. We conclude that $\mathbb{P}[\mathsf{Forge} \wedge \mathsf{Abort}_1]$ must be negligible. □

Claim 6.2. $\mathbb{P}[\mathsf{Abort}_2] \in \mathsf{negl}(\kappa)$.

Proof. Event Abort_2 holds true if for all $i \in [n]$ we have that $(\mathbf{\Delta}(m_i^*), r(m_i^*)) = (\tilde{m}_i^*, \tilde{r}_i^*)$ for all values extracted by B. Consider the matrix $\mathbf{M}$ for which the i-th column is the vector $\mathbf{c}_i = \mathcal{H}_1^{d,t}(m_i^*)$. Let $\mathsf{FullRank}$ be the event that $\mathbf{M}$ is full rank, and suppose that A makes q_s signature queries. Note that the view of A is a random variable View consisting of: (i) the verification key vk (ii) the leakage Λ, (iii) the signatures $(m_1, \sigma_1), \ldots, (m_{q_s}, \sigma_{q_s})$, (iv) the set $\mathcal{Q}_{\mathcal{H}}(\mathsf{A})$ of random oracle calls and corresponding answers. We can write:

$$\mathbb{P}[\mathsf{Abort}_2] \leqslant \mathbb{P}[\neg\mathsf{FullRank}] + 2^{-\widetilde{\mathbb{H}}_\infty(\mathbf{\Delta M}, \mathbf{rM} | \mathsf{View}; \mathsf{FullRank})},$$

where $\widetilde{\mathbb{H}}_\infty(X|Y;E)$ is the average conditional min-entropy of the random variable X given the random variable Y and conditioned on the event E.

Let q_h be the number of random oracle queries made by A, we claim that:

$$\mathbb{P}[\neg\mathsf{FullRank}] \leqslant q_h \cdot \left(\frac{1}{|\mathbb{F}|} + e^{-t}\right).$$

This is because the first $d-1$ columns of $\mathbf{M}$ form a subspace of dimension less then d, and for any subspace $A \subset \mathbb{F}^d$ of dimension less then d the probability that a vector $\mathbf{c}$ randomly chosen

over $\mathsf{Sparse}(\mathbb{F}, d, t)$ lies in the subspace A is at most $\frac{1}{|\mathbb{F}|} + e^{-t}$ (by Lemma 6.1), and so the above inequality follows by the union bound. Hence,

$$\begin{aligned}
\widetilde{\mathbb{H}}_\infty(\mathbf{\Delta M}, \mathbf{rM} |\ \mathsf{View}; \mathsf{FullRank}) &= \widetilde{\mathbb{H}}_\infty(\mathbf{\Delta}, \mathbf{r} |\ vk, \Lambda, \{m_i, \sigma_i\}_{i=1}^{q_s}, \mathcal{Q}_{\mathcal{H}}(\mathsf{A})) && (15)\\
&= \widetilde{\mathbb{H}}_\infty(sk |\ vk, \Lambda, \{m_i, \sigma_i\}_{i=1}^{q_s})\\
&= \widetilde{\mathbb{H}}_\infty(sk |\ vk, \Lambda) && (16)\\
&\geqslant \widetilde{\mathbb{H}}_\infty(sk |\ vk) - \lambda && (17)\\
&\geqslant \rho|sk| - \lambda, && (18)
\end{aligned}$$

where Eq. (15) holds by the fact that $\mathbf{M}$ is full rank, Eq. (16) follows by perfect zero-knowledge of $\mathcal{NIZK}$, Eq. (17) follows by the chain rule on the average conditional min-entropy (see, e.g., [28, Lemma 2.2]), and Eq. (18) holds by the fact that sk is uniform over $\mathbb{F}^{d(\mu+1)}$. We conclude that, for negligible function $\nu : \mathbb{N} \to [0, 1]$,

$$\mathbb{P}[\mathsf{Abort}_2] \leqslant \nu(\kappa) + 2^{\rho|sk|-\lambda} \leqslant \nu(\kappa) + 2^{-\omega(\log \kappa)} \in \mathsf{negl}(\kappa),$$

as desired. □

Combining Claim 6.1 and Claim 6.2 we get that, for a negligible function $\nu' : \mathbb{N} \to [0, 1]$, $\mathbb{P}[\mathsf{B} \text{ wins}] \geqslant \varepsilon - \nu'(\kappa)$, and thus ε must be negligible, finishing the proof. □

6.3 The Full-Fledged Scheme

In order to reduce the size of the verification key in the scheme from the previous section, we rephrase the idea of Alwen et al. [2, Section 4.3] using the notion of homomorphic linear authenticators introduced in [3]. Let $\mathbb{F}$ be a field equipped with a function $\mathsf{Combine}(\mathbf{c}, \mathbf{f})$ that takes as input two vector $\mathbf{c} \in \mathbb{Z}^d$ and $\mathbf{f} \in \mathbb{F}^d$, and outputs a field element, concretely, we can think of it as the scalar multiplication $\sum_i f_i \cdot c_i$ or the scalar multiplication "in the exponents" $\prod_i f_i^{c_i}$.

A public-key homomorphic linear authenticator for a field $\mathbb{F}$ equipped with a function $\mathsf{Combine}(\mathbf{c}, \mathbf{f})$ is a tuple of four PPT algorithms $\mathcal{HLA} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Auth}, \mathsf{Ver})$ such that:

- $(PK, SK) \leftarrow\!\!\$\ \mathsf{KGen}(1^\kappa)$ is a probabilistic algorithm used to setup the scheme. It takes as input the security parameter and the field description $\mathbb{F}$ (where $\log|\mathbb{F}| = \kappa$), and outputs a public and private key pair (PK, SK).
- $(\mathbf{tag}, st) \leftarrow\!\!\$\ \mathsf{Tag}(SK, \mathbf{f})$ is a probabilistic algorithm that is run in order to tag a vector. It takes as input a secret key SK and a vector $\mathbf{f} \in \mathbb{F}^d$, and outputs a vector of tags $\mathbf{tag} \in \mathbb{F}^d$ and state information st.
- $\tau := \mathsf{Auth}(PK, \mathbf{f}, \mathbf{tag}, \mathbf{c})$ is a deterministic algorithm that is run to generate tags. It takes as input a public key PK, a vector $\mathbf{f} \in \mathbb{F}^d$, a tag vector $\mathbf{tag}$, and a challenge vector $\mathbf{c} \in \mathbb{Z}^d$, and outputs a tag τ.
- $b := \mathsf{Ver}(PK, st, \gamma, \mathbf{c}, \tau)$ is a deterministic algorithm that is used to verify a tag. It takes as input a public key PK, state information st, an element $\gamma \in \mathbb{F}$, a challenge vector $\mathbf{c} \in \mathbb{Z}^d$, and a tag τ. It outputs a bit b, where $b = 1$ indicates acceptance and $b = 0$ indicates rejection.

For correctness, we require that for all $\kappa \in \mathbb{N}$, all (PK, SK) output by $\mathsf{KGen}(1^\kappa)$, all $\mathbf{f} \in \mathbb{F}^d$, all $(\mathbf{tag}, st)$ output by $\mathsf{Tag}(SK, \mathbf{f})$, and all $\mathbf{c} \in \mathbb{Z}^d$, it holds that

$$\mathsf{Ver}\left(PK, st, \mathsf{Combine}(\mathbf{c}, \mathbf{f}), \mathbf{c}, \mathsf{Auth}(PK, \mathbf{f}, \mathbf{tag}, \mathbf{c})\right) = 1.$$

Definition 6.3 (Unforgeability for public-key HLAs)**.** *Let* A *be an adversary for the following experiment:*

1. *The challenger computes* $(PK, SK) \leftarrow\!\!\$\ \mathsf{KGen}(1^\kappa)$.
2. *Given* PK *and oracle access to* $\mathsf{Tag}(SK, \cdot)$*, adversary* A *outputs a vector* $\mathbf{f} \in \mathbb{F}^d$.
3. *The challenger tags* $\mathbf{f}$ *by computing* $(\mathbf{tag}, st) \leftarrow\!\!\$\ \mathsf{Tag}(SK, \mathbf{f})$.
4. *Given* $\mathbf{tag}$ *and* st*, the adversary* A *outputs a challenge vector* $\mathbf{c} \in \mathbb{Z}^d$*, an element* $\gamma' \in \mathbb{F}$*, and a tag* τ.
5. *The adversary succeeds if* $\gamma' \neq \mathsf{Combine}(\mathbf{c}, \mathbf{f})$ *and* $\mathsf{Ver}(PK, st, \gamma', \mathbf{c}, \tau) = 1$.

We say that $\mathcal{HLA}$ *is unforgeable if the success probability of every PPT adversary* A *in the above experiment is negligible.*

We are now ready to describe our signature scheme $\mathcal{SS}_4 = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$, based on our construction from the previous section $\mathcal{SS}'_4 = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ and a public-key homomorphic linear authenticator $\mathcal{HLA} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Auth}, \mathsf{Ver})$.

Key Generation. Run $(vk', sk') \leftarrow\!\!\$\ \mathcal{SS}'_4.\mathsf{KGen}(1^\kappa)$ and $(PK, SK) \leftarrow\!\!\$\ \mathcal{HLA}.\mathsf{KGen}(1^\kappa)$; parse vk' as $(\vartheta, \{com_i\}_{i=1}^d)$ and define $\mathbf{f} = (com_1, \ldots, com_d)$. Generate $(\mathbf{tag}, st) \leftarrow\!\!\$\ \mathsf{Tag}(SK, \mathbf{f})$ and output:

$$sk := (sk', \mathbf{tag}) \qquad vk := (\vartheta, PK, st).$$

Signature. Run $\pi \leftarrow\!\!\$\ \mathcal{SS}'_4.\mathsf{Sign}(sk, m)$, and compute $\tilde{com} = \prod_{i=1}^d com_i^{c_i}$ and $\tau := \mathsf{Auth}(PK, \mathbf{f}, \mathbf{tag}, \mathbf{c})$ where $\mathbf{c} = (c_1, \ldots, c_d) = \mathcal{H}_1^{d,t}(m)$. Output $\sigma := (\tilde{com}, \tau, \pi)$.

Verification Given a pair (m, σ), parse σ as $(\tilde{com}, \tau, \pi)$. Compute $\mathbf{c} := \mathcal{H}_1^{d,t}(m)$ and output:

$$\mathcal{HLA}.\mathsf{Ver}(PK, st, \tilde{com}, \mathbf{c}, \tau) \wedge \mathsf{Ver}^{\mathcal{H}_0}((\mathsf{crs}, \tilde{com}), \pi).$$

Theorem 6.2. *Let* $\mathbb{F}$ *be a finite field such that* $\log|\mathbb{F}| = \kappa$*, for security parameter* $\kappa \in \mathbb{N}$*. For any* $\xi = \omega(\log \kappa)$*, whenever* $\gamma = O(1)$ *the above signature scheme is* $((\frac{\mu}{\mu+2})|sk| - \xi, \gamma)$*-fully-leakage one-more unforgeable in the bounded retrieval model under the condition that the adversary does not leak from the randomness used during key generation.*

Note that the scheme does not allow leakage on the coins of the key generation; these coins have to be securely erased (together with SK). However, after key generation is over no more erasures are needed.

Proof. Let A be an adversary in the fully-leakage one-more unforgeability experiment for $\mathcal{SS}_4$.

We construct an adversary B that either breaks fully-leakage one-more unforgeability of $\mathcal{SS}'_4$ (contradicting Theorem 6.1) or the unforgeability of $\mathcal{HLA}$. Adversary B plays both games at the same time, with challenger C_1 (for $\mathcal{SS}'_4$) and C_2 (for $\mathcal{HLA}$). A description of B follows:

1. At the beginning B receives as input the verification key $vk' = (\vartheta, \{com_i\}_{i=1}^d)$ (from C_1), and the public key for the HLA PK (from C_2). Recall that there is no leakage during the key generation phase. Thus, B queries oracle $\mathsf{Tag}(SK, \cdot)$ upon input the value $\mathbf{f} = (com_1, \ldots, com_d)$, obtaining a pair $(\mathbf{tag}, st)$, and returns $vk = (\vartheta, PK, st)$ to A.
2. Upon input a signature query m from A, adversary B forwards this query to C_1 receiving back a signature π. It then computes $\mathbf{c} = (c_1, \ldots, c_d) = \mathcal{H}_1^{d,t}(m)$, $\tilde{com} = \prod_{i=1}^d com_i^{c_i}$, $\tau = \mathsf{Auth}(PK, \mathbf{f}, \mathbf{tag}, \mathbf{c})$, and returns $(\tilde{com}, \tau, \pi)$ to A.
3. Upon input a leakage query $f(\cdot)$ from A, adversary B defines a function $f'(\cdot) := f(\mathbf{tag}, \cdot)$ and forwards such query to C_1. Note that this is a perfect simulation, as the random coins for signature generation consists only of the coins for the signing algorithm of $\mathcal{SS}'_4$.

4. Eventually A returns n forgeries $(m_1^*, (\tilde{com}_1^*, \tau_1^*, \pi_1^*), \ldots, m_n^*, (\tilde{com}_n^*, \tau_n^*, \pi_n^*))$. At this point B checks, for $\mathbf{c}_i^* = \mathcal{H}_1^{d,t}(m_i^*)$, that there exists an index $i \in [n]$ such that

$$\mathsf{Ver}(PK, st, \tilde{com}_i^*, \mathbf{c}_i^*, \tau_i^*) = 1 \wedge \tilde{com}_i^* \neq \mathsf{Combine}(\mathbf{c}_i^*, \mathbf{f}). \tag{19}$$

In case such index is found, B outputs $(\tilde{com}_i^*, \mathbf{c}_i^*, \tau_i^*)$ to C_2. Otherwise, B outputs the forgeries $(m_1^*, \pi_1^*, \ldots, m_n^*, \pi_n^*)$ to C_1.

Let Fresh be the event that Eq. (19) holds for some $i \in [n]$. Moreover, let Forge be the event that A wins in the fully-leakage one-more unforgeability experiment. We can write:

$$\mathbb{P}[\mathsf{Forge}] = \mathbb{P}[\mathsf{Forge} \wedge \mathsf{Fresh}] + \mathbb{P}[\mathsf{Forge} \wedge \neg\mathsf{Fresh}].$$

The two claims below conclude the proof.

Claim 6.3. $\mathbb{P}[\mathsf{Forge} \wedge \mathsf{Fresh}] \in \mathsf{negl}(\kappa)$.

Proof. One can easily see that in case A provokes both Forge and Fresh, adversary B is successful against C_2 contradicting unforgeability of the HLA. This is because $(\tilde{com}_i^*, \mathbf{c}_i^*, \tau_i^*)$ is verified correctly, and moreover $\tilde{com}_i^*$ is a fresh value different than $\mathsf{Combine}(\mathbf{c}_i^*, \mathbf{f})$. □

Claim 6.4. $\mathbb{P}[\mathsf{Forge} \wedge \neg\mathsf{Fresh}] \in \mathsf{negl}(\kappa)$.

Proof. One can easily see that in case A provokes both Forge and $\neg\mathsf{Fresh}$, the forgeries output by B against C_1 are all accepting and with the right distribution, as, in particular, $\tilde{com}_i^* = \mathsf{Combine}(\mathbf{c}_i^*, \mathbf{f}) = \prod_{j=1}^{d} com_j^{\mathbf{c}_i[j]}$. This contradicts fully-leakage one-more unforgeability of $\mathcal{SS}'_4$. □

□

7 Instantiations

In the following let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of prime order q, supporting oblivious sampling of group elements, and let $\mathsf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a non-degenerate efficiently computable bilinear map. Set the field $\mathbb{F}$ to be $\mathbb{Z}_q$. For ease of notation, given three vectors $\mathbf{g}, \mathbf{h} \in \mathbb{G}^n$ and $\boldsymbol{\alpha} \in \mathbb{F}^n$ and an element $\alpha \in \mathbb{F}$ we write $\langle \mathbf{g}, \boldsymbol{\alpha} \rangle$ for the group element $\prod_i g_i^{\alpha_i}$ (as already done in Section 4.3), $\mathbf{g} \cdot \mathbf{h}$ for the vector $(g_1 \cdot h_1, \ldots, g_n \cdot h_n)$, $\mathbf{g}^{\boldsymbol{\alpha}}$ for the vector $(g_1^{\alpha_1}, \ldots, g_n^{\alpha_n})$, $\mathbf{g}^{\alpha}$ for the vector $(g_1^{\alpha}, \ldots, g_n^{\alpha})$ and $\hat{\mathsf{e}}(\mathbf{g}, h)$ for $(\mathsf{e}(g_1, h), \ldots, \mathsf{e}(g_n, h))$.

Definition 7.1 (DLIN assumption [6])**.** *Let $g_1, g_2, g_3 \leftarrow_\$ \mathbb{G}$ and $r, s, z \leftarrow_\$ \mathbb{F}$, the Decisional Linear Assumption states that the following two distributions are computationally indistinguishable:*

$$(\mathbb{G}, g_1, g_2, g_3, g_1^r, g_2^s, g_3^{r+s}) \quad \text{and} \quad (\mathbb{G}, g_1, g_2, g_3, g_1^r, g_2^s, g_3^z).$$

Trapdoor hiding commitment. We use the generalized Pedersen commitment [59]. Algorithm ESetup outputs $\vartheta = \mathbf{g}^{\mathbf{r}}$ for random $\mathbf{r} \leftarrow_\$ \mathbb{F}^{\mu+1}$, and sets the trapdoor to $\tau := \mathbf{r}$. The commitment to an element $m \in \mathbb{F}^{\mu}$ using randomness $\mathbf{r} \leftarrow_\$ \mathbb{F}$ is computed as $c = \mathsf{Com}(\vartheta, m; r) := \langle \vartheta, (m, r) \rangle$.

Note that Pedersen commitment is linearly homomorphic: For $com_1 = \mathsf{Com}(m_1; r_1)$ and $com_2 = \mathsf{Com}(m_2; r_2)$ it holds that

$$com_1 \cdot com_2 = \langle \vartheta, (m_1, r_1) + (m_2, r_2) \rangle = \mathsf{Com}(m_1 + m_2; r_1 + r_2).$$

Moreover, for all constants $c \in \mathbb{F}$ we have that $com_1^c = \langle \vartheta, (m_1, r) \rangle^c = \mathsf{Com}(c \cdot m_1; c \cdot r_1)$. Algorithm Setup obliviously samples an element from $\mathbb{G}^{\mu+1}$; algorithm ECom picks $r' \leftarrow_\$ \mathbb{F}$ and returns $com = g^r$; algorithm Equiv upon input a message $m \in \mathbb{F}^\mu$, trapdoor $\tau = \mathbf{r}$, and randomness r', returns $r := r' - \sum_i r_i m_i$.

Hybrid commitment. We use the homomorphic commitment of [40], that is secure under the DLIN assumption. The trapdoor hiding property is derived applying the general transformation of Damgaard and Nielsen [21].

Let $\psi : \mathbb{F}^2 \to \mathbb{F}^3$ be such that $\psi(r_1, r_2) = (r_1, r_2, r_1 + r_2)$. The algorithms depend on global parameters $(\mathbb{G}, \mathbf{g})$, where $\mathbf{g} \leftarrow_\$ \mathbb{G}^3$. Algorithm Setup chooses $\bar{\mathbf{r}} \leftarrow_\$ \mathbb{F}^3$, and sets the verification key as $\vartheta := \mathbf{g}^{\bar{\mathbf{r}}}$; note that oblivious sampling requires that the group $\mathbb{G}$ itself supports oblivious sampling. Algorithm ESetup chooses $\bar{\mathbf{r}} \leftarrow_\$ \mathbb{F}^2$, and sets the verification and trapdoor keys as $\vartheta := \mathbf{g}^{\psi(\bar{\mathbf{r}})}$ and $\tau := \bar{\mathbf{r}}$. The hybridness property follows by the DLIN assumption.

To commit to a value $m \in \mathbb{F}^\mu$, algorithm Com takes the verification key $\vartheta = \mathbf{h}$, chooses randomness $\mathbf{r}_1, \ldots, \mathbf{r}_\mu \leftarrow_\$ \mathbb{F}^2$, and outputs:

$$com := (\mathbf{h}^{m_1} \cdot \mathbf{g}^{\psi(\mathbf{r}_1)}, \ldots, \mathbf{h}^{m_\mu} \cdot \mathbf{g}^{\psi(\mathbf{r}_\mu)}).$$

Algorithm ECom takes the verification key $\vartheta = (\mathbf{g}^{\psi(\bar{\mathbf{r}}_1)}, \ldots, \mathbf{g}^{\psi(\bar{\mathbf{r}}_\mu)})$ and trapdoor key $\tau = \bar{\mathbf{r}}$, chooses randomness $\mathbf{r'}_1, \ldots, \mathbf{r'}_\mu \leftarrow_\$ \mathbb{F}^2$, and outputs:

$$com := (\mathbf{g}^{\psi(\mathbf{r}_1)}, \ldots, \mathbf{g}^{\psi(\mathbf{r}_\mu)}) \text{ and } \mathbf{r}' := (\mathbf{r'}_1, \ldots, \mathbf{r'}_\mu).$$

To equivocate algorithm Equiv takes as input $\mathbf{r}'$ and a trapdoor key $\tau = \bar{\mathbf{r}}$, and returns $\mathbf{r}_i := \mathbf{r}'_i - m_i \cdot \bar{\mathbf{r}}$ for each $i \in [\mu]$. We note that for any message m, any i, and any verification key ϑ, the distribution of $\mathbf{r}_i$ is a bijective function of $\mathbf{r}'_i$, therefore $\mathbf{r}_i$ is uniformly distributed on $\mathbb{F}^2$ as desired.

In Section 4.2 we require the extra property that given the randomness $\mathbf{r}'_1$ and $\mathbf{r}'_2$ for two equivocal commitments com_1 and com_2 (under verification key ϑ, with trapdoor information $\tau = \bar{\mathbf{r}}$), the randomness $\mathbf{r}'_c := \alpha \cdot \mathbf{r}'_a + \cdot \mathbf{r}'_b$ can equivocate the equivocal commitment $com_3 := com_1^\alpha \cdot com_2$. This is indeed the case, as

$$\begin{aligned}
com_3 &= \left(\mathbf{g}^{\psi(\mathbf{r}'_{a,1})}, \ldots, \mathbf{g}^{\psi(\mathbf{r}'_{a,\mu})}\right)^\alpha \cdot \left(\mathbf{g}^{\psi(\mathbf{r}'_{b,1})}, \ldots, \mathbf{g}^{\psi(\mathbf{r}'_{b,\mu})}\right) = \\
&= \left(\mathbf{g}^{\alpha \cdot \psi(\mathbf{r}'_{a,1}) + \psi(\mathbf{r}'_{b,1})}, \ldots, \mathbf{g}^{\alpha \cdot \psi(\mathbf{r}'_{a,\mu}) + \psi(\mathbf{r}'_{b,\mu})}\right) = \\
&= \left(\mathbf{g}^{\psi(\alpha \cdot \mathbf{r}'_{a,1} + \mathbf{r}'_{b,1})}, \ldots, \mathbf{g}^{\psi(\alpha \cdot \mathbf{r}'_{a,\mu} + \mathbf{r}'_{b,\mu})}\right) = \\
&= \left(\mathbf{g}^{\psi(\mathbf{r}'_{c,1})}, \ldots, \mathbf{g}^{\psi(\mathbf{r}'_{c,\mu})}\right).
\end{aligned}$$

In Section 4.3 we require the following extra properties: (i) $\mathbb{H} = \{\mathbf{g}^{\psi(\mathbf{r})} : \mathbf{r} \in \mathbb{F}^2\}$ forms a proper subgroup of $\mathbb{G} = \{\mathbf{g}^{\mathbf{r}} : \mathbf{r} \in \mathbb{F}^3\}$; (ii) given $\vartheta_1 \in \mathbb{G} \setminus \mathbb{H}$ and $\vartheta_2 \in \mathbb{H}$, the product $\vartheta_1 \cdot \vartheta_2$ (defined as the element-wise multiplication) is in $\mathbb{G} \setminus \mathbb{H}$; (iii) $\mathbf{g}_1^{\psi(\mathbf{r})}$ is uniformly distributed in $\mathbb{H}$, for $\mathbf{r} \leftarrow_\$ \mathbb{F}^2$. The above properties hold by simple inspection.

NIWI arguments. We instantiate the NIWI argument system using the powerful Groth-Sahai proofs system [40]. Using the so-called simulated common reference string key generator, the system is a statistical NIWI argument system.[11] As pointed out in [9], the Init algorithm

[11] Strictly speaking, to instantiate the NIWI argument using [40] one would have to slightly adapt Definition 2.3 in order to account for the fact that the witness indistinguishability property only holds when the CRS is generated using an alternative initialization algorithm $\overline{\mathsf{Init}}$ (producing a computationally indistinguishable distribution); this is easily taken into account in the analysis by defining a further hybrid in the proofs (see Theorem 4.1, Theorem 4.2, and Theorem 5.1), where Init is replaced by $\overline{\mathsf{Init}}$.

admits oblivious sampling. Following [40], the number of group elements for the common reference string is 12 group elements. The relation $\mathfrak{R}$ in Section 4.1 can be expressed using $3\mu + 1$ equations with $3\mu + 1$ variables; the relation $\mathfrak{R}$ in Section 5.2 can be expressed using $(3\mu + 1)\log\kappa + 1$ equations with $4\mu\log\kappa + 1$ variables. According to Libert [48] we have:

- A NIWI argument for the scheme $\mathcal{SS}_1$ requires $O(\mu)$ group elements, while the number of pairings operations for verification is $\Omega(\mu^2)$; the exact number of group elements is comparable with the OR-construction of [12]. The difference is that [12] employs the strong one-time signature of [38], which requires 2 group elements for the public key; instead our PRG (based on DLIN) uses 3 group elements. According to [48], we estimate that the NIWI argument of scheme $\mathcal{SS}_1$ requires approximately $24\mu + 90$ group elements.
- A NIWI argument for the scheme $\mathcal{SS}_2$ requires $15\mu + 5$ group elements, while the number of pairings operations for verification is $27\mu^2 + 36\mu + 12$.
- A NIWI argument for the scheme $\mathcal{SS}_3$ requires $(12\log\kappa + 4)\mu + 4\mu + 5$ group elements, while the number of pairings operations for verification is $\Omega((\mu\log\kappa)^2)$.

NIZK arguments. We use the tSE-NIZK based on DLIN from Dodis [25]. Their instantiation uses a multi-message version of the Cramer-Shoup encryption scheme [16], and the Groth-Sahai proof system. One can check that Cramer-Shoup admits oblivious sampling of the public key.

Roughly, the argument system encrypts the witness and proves that the encrypted value is a valid witness for the relation. For the scheme $\mathcal{SS}_2$ presented in Section 4.3 the witness space for the relation $\mathfrak{R}_2$ (cf. Eq. (9)) is $\mathbb{F}^2$; however the Cramer-Shoup encryption scheme has message space $\mathbb{G}^n$ for some n.

In order to solve the above mismatch, we instantiate the tSE-NIZK with the following relation:

$$\mathfrak{R}_2' := \left\{ (\vartheta, \vartheta_1, \vartheta_2); (g^{\alpha_1}, g^{\alpha_2}) \;\middle|\; \hat{\mathsf{e}}(\vartheta, g) = \hat{\mathsf{e}}(\vartheta_1, g^{\alpha_1}) \cdot \hat{\mathsf{e}}(\vartheta_2, g^{\alpha_2}) \right\} \subseteq (\mathbb{G}^3)^3 \times \mathbb{G}^2.$$

Following [25], the size of a tSE-NIZK argument is of 124 group elements and 6 elements in $\mathbb{F}$, while the common reference string size is of 18 group elements.

Instantiating the scheme in the BRM. We use Fischlin's trasformation [36] for the NIZK with online extraction. To obtain reasonable security parameters the size of a proof needs to be $\Omega(\log\kappa)$ times the prover's transcript in the Sigma protocol. Hence, by using the generalized Pedersen commitment scheme and the corresponding Sigma-protocol for proving knowledge of an opening, we obtain that the size of a signature is $\log\kappa$ group elements and $(\mu + 1)\log\kappa$ elements in $\mathbb{F}$.

For the homomorphic linear authenticator we use the scheme implicitly defined by Shacham and Waters in [63], based on the BLS signature scheme of Boneh *et al.* [7].

8 Extension to Noisy Leakage

Our definitions of Section 3 are in the bounded leakage model, where the adversary is allowed to query a leakage oracle with arbitrary polynomial-time functions, provided that the total amount of leakage is smaller than a global parameter ℓ. In this section, we explain how to extend our definitions and some of our results to a more general setting (that strictly implies the case of bounded leakage) where the leakage functions can have a much longer output, as long as the entropy of the secret key does not decrease by too much.

Following [24], we define a notion of a function being ℓ-leaky.

Definition 8.1 (ℓ-leaky function)**.** *A (possibly randomized) function $f : \{0,1\}^* \to \{0,1\}^*$ is ℓ-leaky, if for all $\kappa \in \mathbb{N}$ we have that $\widetilde{\mathbb{H}}_\infty(U_\kappa | f(U_\kappa)) \geq \kappa - \ell$, where U_κ is the uniform distribution over $\{0,1\}^\kappa$.*

As shown by [24, Lemma L.3] the above notion composes nicely, meaning that if two functions are respectively ℓ_1- and ℓ_2-leaky, their concatenation is $(\ell_1 + \ell_2)$-leaky.

We explain how to adapt Definition 3.1 to the setting of noisy-leakage. Essentially we allow the adversary to query the leakage oracle on arbitrary polynomial-time functions, provided that each function is ℓ_i-leaky and $\sum_i \ell_i \leq \ell$, where ℓ is the leakage parameter. We stress that one-more unforgeability still requires that an adversary should produce a number of forgeries strictly larger than the ones he could have leaked (up-to the slack factor γ), however the number of leaked signatures now might depend on the amount of information that a signature actually carries. We distinguish two cases, depending on whether the signature algorithm (statistically) reveals partial information on the secret key or not. In the first case, the parameter n in the winning condition is related to the leakage parameter ℓ, since a forgery is de-facto a leaky function of the secret; in the second case we need to be more pessimistic, and let the parameter n be related to the actual leakage λ performed by the adversary.

Definition 8.2 (Fully-leakage one-more unforgeability, noisy case)**.** *We say that a signature scheme $\mathcal{SS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Verify})$ is (ℓ, γ)-fully-leakage one-more unforgeable w.r.t. noisy leakage if it satisfies Definition 3.1 with the following modification to the security experiment:*

5. *The experiment outputs* 1 *if and only if conditions (a)-(b) are met and moreover:*

   (c) *Let* Sign *be $\tilde{s}$-leaky and $s := |\sigma|$ be the size of a signature, then:*

$$\begin{cases} n \geq \lfloor \lambda/(\gamma \cdot s) \rfloor + 1 & \text{if } \tilde{s} = 0 \\ n \geq \lfloor \ell/(\gamma \cdot \tilde{s}) \rfloor + 1 & \text{else} \end{cases} \tag{20}$$

   where λ is the total length of the leakage.

   (d) *Each function submitted to the leakage oracle is ℓ_i-leaky, $\sum_i \ell_i \leq \ell$, and $|\mathcal{Q}| \leq q_s$.*

The definitions for the case of perfect erasures, and in the BRM can be updated accordingly. Note that the above formulation offers a graceful degradation of security also for schemes where the secret key is potentially shorter than a signature, and signatures do not constrain the secret key. In fact, for such schemes, leaking a signature is allowed, but does not directly constitute a security breach as the adversary is required to produce at least one more forgery.

It is easy to see that the above formulation strictly implies our definitions in the bounded leakage model. In fact, an ℓ-bit output function is in particular ℓ-leaky.

The theorem below shows that our signature scheme $\mathcal{SS}_3$ (cf. Section 5) and our schemes $\mathcal{SS}'_4$ and $\mathcal{SS}_4$ (cf. Section 6) are secure also w.r.t. noisy leakage. Unfortunately we are not able to prove a similar statement for the schemes $\mathcal{SS}_1$ and $\mathcal{SS}_2$ (cf. Section 4). The reason for this is that in the erasure case the simulator could be forced to leak too much information, as the number of leaky signature queries is not anymore upper-bounded by λ, but is instead an arbitrary polynomial. In such a case, we cannot upper bound the probability of the events Bad_i in the proof of Lemma 4.2. This issue does not apply to the non-erasure case, as the simulator can arbitrarily control the probability of the events Bad_i.[12]

Theorem 8.1. *The statement of Theorem 5.1, Theorem 6.1, and Theorem 6.2 also hold w.r.t. noisy-leakage security, with the same leakage and efficiency parameters.*

[12] We do not have a concrete attack on the schemes.

8.1 Proof Sketch of Theorem 8.1

We will rely on the following technical lemma (see [28, Lemma 2.2] and [24, Lemma L.3]).

Lemma 8.1. *Let X be a random variable over some domain $\mathcal{X}$. The following holds:*

1. *For any random variable $Y \in \mathcal{Y}$, and for any $\beta > 0$, we have that*

$$\mathbb{P}\left[\ \mathbb{H}_\infty(X|Y=y) \geq \widetilde{\mathbb{H}}_\infty(X|Y) - \beta\ \right] \geq 1 - 2^{-\beta}$$

   (where the probability is over the choice of y).

2. *For all sequences of ℓ_i-leaky functions $f_i : \{0,1\}^* \to \{0,1\}^*$, such that $\sum_i \ell_i \leq \ell$, we have that $\widetilde{\mathbb{H}}_\infty(X|f_1(X), f_2(X), \ldots,) \geq \mathbb{H}_\infty(X) - \ell$.*

We start with the statement of Theorem 5.1. We need to show the following twist of Lemma 4.2, where the bound now depends on ℓ (the amount of min-entropy left in the secret key) instead of λ (the total length of the output of the leakage functions).

Lemma 8.2. *For any $\beta > 0$ we have that*

$$\mathbb{P}\left[\mathbb{H}_\infty\left(\mathbf{\Delta}|\ \mathsf{View}_3 = v\right) \geqslant |\mathbf{\Delta}| - (1+B)\mu\log|\mathbb{F}| - \ell - \beta\ \right] \geqslant 1 - 2^{-\beta} - 2^{-2e}. \tag{21}$$

Proof. Let $\mathsf{View}'_3 := (f_i((\mathbf{\Delta}, \mathbf{r}), st_i))_{i\in[q_s]}||(\mathbf{\Delta}(m_i))_{i\in\mathcal{Z}}$ the total information leaked in $\mathcal{H}_3$, where $\mathcal{Z}$, as before, is the set of bad queries that force the simulator to leak from $\mathbf{\Delta}$. For any adversary's view v let v' the "*stripped-off*" view where we consider only the total information leaked via the leakage oracle. We have,

$$\mathbb{H}_\infty\left(\mathbf{\Delta}|\ \mathsf{View}_3 = v\ \right) = \mathbb{H}_\infty\left(\mathbf{\Delta}|\ \mathsf{View}'_3 = v'\ \right).$$

We condition on the number of bad queries $Z = |\mathcal{Z}|$ being less than $2e$, then we apply Property (1) of Lemma 8.1:

$$\mathbb{P}\left[\mathbb{H}_\infty\left(\mathbf{\Delta}|\ \mathsf{View}'_3 = v'\right)\ \geqslant \widetilde{\mathbb{H}}_\infty\left(\mathbf{\Delta}|\ \mathsf{View}'_3\right) - \beta\ \middle|\ Z \leqslant 2e\right] \geqslant 1 - 2^{-\beta}.$$

Now we can apply Property (2) of Lemma 8.1, using the fact that the function $f(\mathbf{\Delta}) := (\mathbf{\Delta}(m_i))_{i\in\mathcal{Z}}$ is $(Z\mu\log|\mathbb{F}|)$-leaky:

$$\mathbb{P}\left[\mathbb{H}_\infty\left(\mathbf{\Delta}|\ \mathsf{View}'_3 = v'\right) \geqslant |\mathbf{\Delta}| - (2e\mu\log|\mathbb{F}| + \ell) - \beta\ \middle|\ Z \leqslant 2e\right] \geqslant 1 - 2^{-\beta}.$$

Since $\mathbb{P}[Z \leqslant 2e] > 1 - 2^{-2e}$, by the Chernoff bound, we obtain the statement of the lemma. □

To conclude the proof, one can verify that, similarly to Theorem 5.1, in order to obtain a contradiction we need that:

$$\frac{n\kappa}{q_s} = \Omega(\ell - \log\varepsilon(\kappa)). \tag{22}$$

For $\varepsilon(\kappa)$ noticeable, since $n \geq \lfloor\frac{\ell}{\gamma\cdot\kappa}\rfloor + 1$ (as the signing algorithm is κ-leaky), it suffices to set $\gamma = O(\frac{1}{q_s})$ in order for Eq. (22) to hold.

We turn to the proof of the noisy-variant of Theorem 6.1 and Theorem 6.2. It suffices to prove Theorem 6.1, as the other proof is identical. The proof proceeds as before, except that now Eq. (17) follows by a direct application of Property (2) of Lemma 8.1 (and not by the chain rule of average conditional min-entropy). The signature algorithm is 0-leaky, so condition (c) in the definition of fully-leakage one-more unforgeability stays the same (see Definition 8.2). The proof follows.

9 Conclusions and Open Problems

We have shown new constructions of fully leakage-resilient signatures in the bounded and noisy leakage model. Our schemes enjoy a graceful degradation of security in situations where the size of a single signature is independent of the size of the secret key. This notion is still meaningful for security, and moreover allows to obtain shorter signature tolerating a $1 - o(1)$ fraction of leakage on the secret key.

Our main schemes are efficient, in the standard model, and can be instantiated under fairly standard cryptographic assumptions. We have also built a scheme in the BRM, relying on a random oracle.

We conclude by highlighting a few questions left open by our work. As for the standard model constructions, it would be interesting to construct more efficient schemes, both in the erasure and non-erasure model, with optimal slack parameter $\gamma = O(1)$. As for the scheme in the BRM, it is a challenging open question whether it is possible to remove the random oracle. Also it would be interesting to make the proof work without assuming online extractability for the NIZK (e.g., using the Fiat-Shamir transform [35]). This case requires a more careful analysis, dealing with rewinding an adversary that outputs n forgeries, and seems to require a more general version of the forking lemma [60].

Finally, it remains open to generalize our schemes to the setting of continuous and hard-to-invert leakage (possibly without relying on hardware assumptions, like the Floppy model of [2, 19, 20]).

References

[1] Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrieval model. In *EUROCRYPT*, pages 113–134, 2010.

[2] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *CRYPTO*, pages 36–54, 2009.

[3] Giuseppe Ateniese, Seny Kamara, and Jonathan Katz. Proofs of storage from homomorphic identification protocols. In *ASIACRYPT*, pages 319–333, 2009.

[4] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS*, pages 62–73, 1993.

[5] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *CRYPTO*, pages 443–459, 2004.

[6] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO*, pages 41–55, 2004.

[7] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In *ASIACRYPT*, pages 514–532, 2001.

[8] Elette Boyle, Shafi Goldwasser, Abhishek Jain, and Yael Tauman Kalai. Multiparty computation secure against continual memory leakage. In *ACM STOC*, pages 1235–1254, 2012.

[9] Elette Boyle, Gil Segev, and Daniel Wichs. Fully leakage-resilient signatures. In *EUROCRYPT*, pages 89–108, 2011.

[10] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *CRYPTO*, pages 1–20, 2010.

[11] 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 *IEEE FOCS*, pages 501–510, 2010.

[12] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *EUROCRYPT*, pages 351–368, 2009.

[13] Ran Canetti, Dror Eiger, Shafi Goldwasser, and Dah-Yoh Lim. How to protect yourself without perfect shredding. In *ICALP*, pages 511–523, 2008.

[14] Dario Catalano and Ivan Visconti. Hybrid trapdoor commitments and their applications. In *ICALP*, pages 298–310, 2005.

[15] Sherman S. M. Chow, Yevgeniy Dodis, Yannis Rouselakis, and Brent Waters. Practical leakage-resilient identity-based encryption from simple assumptions. In *ACM CCS*, pages 152–161, 2010.

[16] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.

[17] Giovanni Di Crescenzo, Richard J. Lipton, and Shabsi Walfish. Perfectly secure password protocols in the bounded retrieval model. In *TCC*, pages 225–244, 2006.

[18] Özgür Dagdelen and Daniele Venturi. A second look at Fischlin's transformation. In *AFRICACRYPT*, pages 356–376, 2014.

[19] Ivan Damgaard, Sebastian Faust, Pratyay Mukherjee, and Daniele Venturi. Bounded tamper resilience: How to go beyond the algebraic barrier. In *ASIACRYPT*, pages 140–160, 2013.

[20] Ivan Damgaard, Sebastian Faust, Pratyay Mukherjee, and Daniele Venturi. Bounded tamper resilience: How to go beyond the algebraic barrier. *J. Cryptology*, ??(??):1–39, 2015.

[21] Ivan Damgaard and Jesper Buus Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In *CRYPTO*, pages 581–596, 2002.

[22] Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In *SCN*, pages 121–137, 2010.

[23] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *TCC*, pages 361–381, 2010.

[24] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Cryptography against continuous memory attacks. In *IEEE FOCS*, pages 511–520, 2010.

[25] 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.

[26] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *ACM STOC*, pages 621–630, 2009.

[27] Yevgeniy Dodis, Allison B. Lewko, Brent Waters, and Daniel Wichs. Storing secrets on continually leaky devices. In *IEEE FOCS*, pages 688–697, 2011.

[28] 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.

[29] Stefan Dziembowski. Intrusion-resilience via the bounded-storage model. In *TCC*, pages 207–224, 2006.

[30] Stefan Dziembowski. On forward-secure storage. In *CRYPTO*, pages 251–270, 2006.

[31] Antonio Faonio, Jesper Buus Nielsen, and Daniele Venturi. Mind your coins: Fully leakage-resilient signatures with graceful degradation. In *ICALP*, pages 456–468, 2015.

[32] Sebastian Faust, Carmit Hazay, Jesper Buus Nielsen, Peter Sebastian Nordholt, and Angela Zottarel. Signature schemes secure against hard-to-invert leakage. In *ASIACRYPT*, pages 98–115, 2012.

[33] Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy N. Rothblum. Leakage-resilient signatures. In *TCC*, pages 343–360, 2010.

[34] 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.

[35] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

[36] Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In *CRYPTO*, pages 152–168, 2005.

[37] Sanjam Garg, Abhishek Jain, and Amit Sahai. Leakage-resilient zero knowledge. In *CRYPTO*, pages 297–315, 2011.

[38] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *ASIACRYPT*, pages 444–459, 2006.

[39] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In *EUROCRYPT*, pages 339–358, 2006.

[40] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, pages 415–432, 2008.

[41] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest we remember: Cold boot attacks on encryption keys. In *USENIX Security Symposium*, pages 45–60, 2008.

[42] Shai Halevi and Huijia Lin. After-the-fact leakage in public-key encryption. In *TCC*, pages 107–124, 2011.

[43] Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. In *CRYPTO*, pages 21–38, 2008.

[44] Yael Tauman Kalai, Bhavana Kanukurthi, and Amit Sahai. Cryptography with tamperable and leaky memory. In *CRYPTO*, pages 373–390, 2011.

[45] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *ASIACRYPT*, pages 703–720, 2009.

[46] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *CRYPTO*, pages 104–113, 1996.

[47] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *CRYPTO*, pages 388–397, 1999.

[48] Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and cca2-secure encryption from homomorphic signatures. In *EUROCRYPT*, pages 514–532, 2014.

[49] Jake Longo, Daniel P. Martin, Elisabeth Oswald, Daniel Page, Martijn Stam, and Michael Tunstall. Simulatable leakage: Analysis, pitfalls, and new constructions. In *ASIACRYPT*, pages 223–242, 2014.

[50] Vadim Lyubashevsky, Adriana Palacio, and Gil Segev. Public-key cryptographic primitives provably as secure as subset sum. In *TCC*, pages 382–400, 2010.

[51] Tal Malkin, Isamu Teranishi, Yevgeniy Vahlis, and Moti Yung. Signatures resilient to continual leakage on memory and computation. In *TCC*, pages 89–106, 2011.

[52] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *TCC*, pages 278–296, 2004.

[53] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages 18–35, 2009.

[54] Jesper Buus Nielsen, Daniele Venturi, and Angela Zottarel. On the connection between leakage tolerance and adaptive security. In *PKC*, pages 497–515, 2013.

[55] Jesper Buus Nielsen, Daniele Venturi, and Angela Zottarel. Leakage-resilient signatures with graceful degradation. In *PKC*, pages 362–379, 2014.

[56] Rafail Ostrovsky, Giuseppe Persiano, and Ivan Visconti. Impossibility of black-box simulation against leakage attacks. In *CRYPTO*, pages 130–149, 2015.

[57] Omkant Pandey. Achieving constant round leakage-resilient zero-knowledge. In *TCC*, pages 146–166, 2014.

[58] Rafael Pass. On deniability in the common reference string and random oracle model. In *CRYPTO*, pages 316–337, 2003.

[59] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO*, pages 129–140, 1991.

[60] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.

[61] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In *E-smart*, pages 200–210, 2001.

[62] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *CRYPTO*, pages 566–598, 2001.

[63] Hovav Shacham and Brent Waters. Compact proofs of retrievability. In *ASIACRYPT*, pages 90–107, 2008.

[64] Fran**c**cois-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *EUROCRYPT*, pages 443–461, 2009.

[65] Fran**c**cois-Xavier Standaert, Olivier Pereira, and Yu Yu. Leakage-resilient symmetric cryptography under empirically verifiable assumptions. In *CRYPTO*, pages 335–352, 2013.

[66] Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.