Leakage-Resilient Identification Schemes from Zero-Knowledge Proofs of Storage[*]

Giuseppe Ateniese[†]
Sapienza, University of Rome
ateniese@di.uniroma1.it

Antonio Faonio[‡]
Aarhus University
antfa@cs.au.dk

Seny Kamara
Microsoft Research
senyk@microsoft.com

Abstract

We provide a framework for constructing leakage-resilient identification (ID) protocols in the bounded retrieval model (BRM) from proofs of storage (PoS) that hide partial information about the file. More precisely, we describe a generic transformation from any zero-knowledge PoS to a leakage-resilient ID protocol in the BRM. We then describe a ZK-PoS based on RSA which, under our transformation, yields the first ID protocol in the BRM based on RSA (in the ROM). The resulting protocol relies on a different computational assumption and is more efficient than previously-known constructions.

Keywords: Leakage Resilience, Bounded Retrieval Model, Proof of Storage, Identification Scheme, Generic Transformation, RSA security.

1 Introduction

Cryptographic schemes are traditionally designed under the assumption that the adversary cannot learn any information about the secret key. In practice, however, this assumption does not always hold as the adversary could recover information about the key through various means such as side-channel attacks [19, 20, 26, 5, 23], memory leakage attacks [16] or by compromising the system on which the keys are stored. These attacks, commonly referred to as *leakage attacks*, have motivated the design of *leakage-resilient* cryptosystems which remain secure even against adversaries that may obtain partial information about the secret state (clearly, under some limitations on the kind of leakage allowed). Several models of leakage-resilience have been proposed and many cryptographic primitives have been realized under gradually stronger models [22, 12, 24, 1, 18, 14, 10]. In what follows we discuss only the most relevant to our work, specifically, we focus on the *bounded retrieval model* (BRM). In this model, there is an absolute upper bound λ on the total amount of information the adversary can recover about the secret key. In the BRM this bound is independent of k, the security parameter, thus security can only be achieved if the key is larger than λ. Since the latter can be very large, we require that the efficiency of the scheme be related only to the security

[*] A preliminary version of this paper is published in the proceedings of the 15th IMA International Conference on Cryptography and Coding, IMACC 2015, St. Catherine's College, University of Oxford, UK. This is the full version.

[†] Partially supported by the European Unions Horizon 2020 research and innovation programme under grant agreement No 644666, project "SUNFISH".

[‡] Supported by European Research Council Starting Grant 279447.

parameter. The BRM model was introduced by Di Crescenzo *et al.* [8] and by Dziembowski [11]. The former showed how to construct password-based key agreement protocols while the latter proposed a symmetric-key authenticated key agreement (AKA) protocol. In this work, we consider the problem of identification in the BRM. More precisely, we are interested in practical identification schemes that support large secret keys and whose efficiency is independent of the key length. The problem was first considered by Alwen *et al.* [1], our contribution provides a new and different perspective, which results in a practical scheme based on RSA.

1.1 Our Contributions

We provide a framework for constructing leakage-resilient ID protocols in the BRM from publicly-verifiable proofs of storage (PoS) that are computationally zero-knowledge (ZK). PoS are interactive protocols allowing a client to verify that a server faithfully stores its file. A PoS is publicly verifiable if anyone with access to the client's public-key can verify the server's storage and it is computationally ZK if, roughly speaking, its verification phase leaks no useful information about the file to a bounded adversary. We show how to construct such a scheme based on the RSA assumption.

PoS were introduced independently by Ateniese *et al.* [2] and Juels and Kaliski [17]. Publicly verifiable PoS were first considered in [2] with extensions and improvements given in [27, 3]. We summarize the contributions of this work as follows:

1. **(generality)** We provide a transformation from any zero-knowledge (ZK) PoS to a BRM identification scheme.

2. **(efficiency)** Our ZK-PoS-to-BRM-ID transformation is very efficient, leading to BRM-ID schemes that are practical and more efficient than prior work.

3. **(security)** We show how to build ZK-PoS under standard cryptographic assumptions. In particular, we propose a novel BRM-ID scheme based on the standard RSA assumption in the random oracle model (ROM).

1.2 Related Work

Leakage-resilient identification schemes in the BRM were first considered in [1] which proposed a scheme based on the generalized Okamoto scheme (see Okamoto [25]) and the pairing-based public-key homomorphic linear authenticator of Shacham and Waters [27]. In [1], a transformations is also given from absolute leakage-resilient ID schemes to leakage-resilient signature schemes and AKA protocols. The transformation relies on parallel-repetition and consists in taking n independent copies of the basic relative-leakage scheme. Since n is large, this yields complex and relatively inefficient schemes, thus a more efficient transformation is described by the authors that employs subset selection and reduces both communication and time complexity.

For a detailed comparison between the constructions of [1] and our own, we refer the reader to Section 4.1. Here, we just mention that the framework of [1] works only for an extension of the Okamoto ID scheme [25] and is not generalizable. Also, the BRM-ID scheme based on the Okamoto ID scheme relies on BLS signatures [6] and thus on the Gap Diffie-Hellman assumption. For the same level of security, we provide schemes that rely on weaker computational assumptions and that are more efficient in terms of computation.

While zero-knowledge PoS can be designed from general-purpose zero-knowledge proofs by having the server prove knowledge of the file, such an approach would not be efficient. The first practical ZK-PoS scheme was proposed by Wang *et al.* [28] who extended the pairing-based PoS construction of Shacham and Waters [27] to be zero-knowledge. In comparison, our RSA-based ZK-PoS relies on a weaker computational assumption and, as far as we know, is the first construction to have a full proof of security.

1.3 Overview of Our Technique

At a high level, our framework works as follows. The secret key of the identification protocol is the encoding of a randomly-generated file and the public key is the state information generated by encoding the file together with the public key for the PoS. To identify itself, the prover executes the verification phase of the PoS with the verifier to prove that it indeed holds the file. Note that while (in the context of a BRM leakage attack) the verifier can learn λ bits about the key/file, the properties of the PoS allow us to increase the file size beyond λ without increasing the communication complexity of the verification phase.

One problem with the above approach is that standard PoS do not necessarily hide information about the file from the verifier and, therefore, the ID scheme verifier above could learn the remaining $n - \lambda$ bits of the key from the verification phase. To address this, we need a *zero-knowledge* PoS; that is, a PoS with a verification phase that hides all partial information about the file.

More formally, for the identification scheme we consider the security notion of pre-impersonation leakage-resistance, in which an attacker, in a test stage of the experiment, can interact with an honest prover and leak arbitrary functions of the secret key. We model the latter with a leakage oracle that on input an efficiently computable (and adaptively chosen) function f_i outputs the value $f_i(sk)$. The restriction is that the total length of the leaked information is bounded by some a-priori fixed value λ.

For the PoS, we phrase the soundness definition using the paradigm of "witness-extended emulation" (see Lindell [21]). Intuitively, this guarantees that there exists an expected polynomial time extractor that, for any adversary that convinces the verifier with some probability, outputs the original file with approximately the same probability.

The main intuition is that even after the test stage, an adversary cannot have *full knowledge* of the secret key/file. It follows then by the (knowledge) soundness of the PoS that the adversary cannot convince the verifier. In the intuition above we have not defined the meaning of knowledge of the adversary after the test stage. At first glance, one might consider the average conditional min-entropy of the secret key/file after the test stage. This measure, however, is insufficient for two reasons:

1. The PoS is only *computationally* zero knowledge so, in principle, all the min-entropy of the file could be lost after the test stage.

2. The conditional average min-entropy is not "smooth" with respect to statistically-close distributions. Specifically, given a random variable X and two statistically-close random variables Y and Y', there could be an arbitrary gap between $\widetilde{\mathbf{H}}_\infty(X \mid Y)$ and $\widetilde{\mathbf{H}}_\infty(X \mid Y')$. Therefore, even if we considered the stronger notion of statistical zero-knowledge PoS, we might run into the same problem.

We overcome the above problems by considering a slightly different experiment. In the new experiment the prover oracle is substituted by the simulator guaranteed to exist by the zero knowledge property of the PoS. The crux is that a polynomially-bounded adversary cannot distinguish the two experiments and, therefore, it can convince the verifier with approximately the same probability. Now we can give a meaningful lower bound on the average conditional min-entropy of the secret key/file after the test stage. The adversary cannot guess the original secret with probability roughly more than $2^{-|sk|+\lambda} \leq 2^{-\omega(\log k)}$ so, by soundness of the PoS, it cannot convince the verifier with noticeable probability.

Concretely, the proof proceeds in two steps. First, we establish a lower bound on the conditional average min-entropy of an *encoding* $\vec{f'}$ of a uniformly random file $\vec{f}$ when the adversary is given access to a leakage oracle parameterized with $\vec{f'}$, and the randomness necessary to encode $\vec{f}$. We then show that if there exists a probabilistic polynomial time (ppt) adversary $\mathcal{A}$ that succeeds in the pre-impersonation leakage experiment with a noticeable probability, then, by the soundness of the PoS, the lower bound on the average conditional min-entropy mentioned above is violated. This follows because we can simulate the pre-impersonation leakage experiment and then successfully extract from the adversary the file $\vec{f}$ during the impersonation stage. Furthermore, the experiment provides the information necessary to reconstruct $\vec{f'}$ from $\vec{f}$. This leads to a predictor that guesses the encoded file $\vec{f'}$ with noticeable probability.

A comparison. Consider the proof of security of the identification schemes presented in [1]. Briefly, their proof technique relies on a collision resistant hash (CRH) function and the identification scheme is a proof of knowledge of a preimage x (the secret key) for an element y (the public key) in the co-domain of the hash function. The reduction samples a secret key x in the domain of the CRH function h and given the secret key, the reduction can easily reply to all the leakage queries. If the adversary succeeds in the pre-impersonation experiment then the reduction can extract a pre-image x'. Their analysis shows that the uncertainty of x is high even after the test stage and therefore with high probability $x' \neq x$ and $y = h(x') = h(x)$. In comparison with our work, they present a direct reduction to the computational problem of breaking a CRH function.

Our proof has a similar interpretation. Given a successful adversary for the pre-impersonation leakage experiment we define a new adversary for the PoS security experiment. This new adversary "forgets" part of the file (namely it has only λ bits of information about it) and convinces the verifier of the PoS scheme, therefore breaking the knowledge soundness of the proof of storage. However, since we cannot directly argue that a forgetful adversary that convinces the verifier breaks the security of PoS, we formalize it providing the two bounds mentioned before. A similar technique, although based on a different measure of min-entropy, was recently used in the context of fully leakage-resilient signature (see Faonio *et al.* [13]).

2 Definitions

2.1 Preliminaries

If x is a string, we denote its length by $|x|$; if $\mathbf{X}$ is a set, $|\mathbf{X}|$ represents the number of elements in $\mathbf{X}$. When x is chosen randomly in $\mathbf{X}$, we write $x \leftarrow \mathbf{X}$. When $\mathcal{A}$ is an algorithm, we write $y \leftarrow \mathcal{A}(x)$ to denote a run of $\mathcal{A}$ on input x and output y; if $\mathcal{A}$ is randomized, then y is a random variable and $\mathcal{A}(x; r)$ denotes a run of $\mathcal{A}$ on input x and randomness r; sometimes, when $\mathcal{A}$ is deterministic we write $y := \mathcal{A}(x)$. An algorithm $\mathcal{A}$ is *probabilistic polynomial-time* (ppt) if it is randomized and for any input $x, r \in \{0,1\}^*$ the computation of $\mathcal{A}(x; r)$ terminates in at most $\mathsf{poly}(|x|)$ steps.

Throughout the paper we let k denote the security parameter. We say that a function $\nu : \mathbb{N} \to \mathbb{R}$

is negligible in the security parameter k if $\nu(k) = k^{-\omega(1)}$. A positive function f is noticeable if there exist a positive polynomial p and a number n_0 such that $f(n) \geq 1/p(n)$ for all $n \geq n_0$.

We start by recalling the notion of conditional min-entropy. We adopt the definition given in [1], where the authors generalize the notion of conditional min-entropy to *interactive* predictors that participate in some randomized experiment $\mathbf{E}$. The (average) conditional min-entropy of random variable X given any randomized experiment $\mathbf{E}$ is defined as follows:

$$\widetilde{\mathbf{H}}_\infty (X \mid \mathbf{E}) = \max_{\mathcal{B}} \left(- \log \Pr \left[\mathcal{B}()^{\mathbf{E}} = X \right] \right),$$

where the maximum is taken over all predictors without any requirement on efficiency. Note that w.l.o.g. the predictor $\mathcal{B}$ is deterministic, in fact, we can derandomize $\mathcal{B}$ by hardwiring the random coins that maximize his outcome. Sometimes we write $\widetilde{\mathbf{H}}_\infty(X|Y)$ for a random variable Y, in this case we mean the average conditional min-entropy of X given the random experiment that gives Y as input to the predictor.

We recall the definition of δ-indistinguishability for ensembles of distribution, both in the computational and statistical flavors.

Definition 1 (Indistinguishability). *Given a function $\delta : N \to \mathbb{R}$ and two distribution ensembles $\{X_k\}_{k \geq 0}$ and $\{Y_k\}_{k \geq 0}$ such that $|X_k| \leq p(k)$ and $|Y_k| \leq p(k)$ for a polynomial $p(k)$, we say that the ensemble $\{X_k\}_{k \geq 0}$ is δ-indistinguishable from $\{Y_k\}_{k \geq 0}$ if for any non-uniform polynomial time distinguisher $\mathcal{D}$ the following holds:*

$$| \Pr \left[\mathcal{D}(1^k, X_k) = 1 \right] - \Pr \left[\mathcal{D}(1^k, Y_k) = 1 \right] \leq \delta(k).$$

When we refer to statistical δ-indistinguishability, the equation above holds for all distinguishers without any bound on the running time.

2.2 Proofs of Storage

Publicly-verifiable PoS consist of two phases: a setup phase where the client encodes the file and sends it to the server; and a verification phase where a verifier (which may or may not be the original client) engages in an interactive protocol with the server to determine if it indeed possesses the file. The encoding algorithm also outputs a "state information" which represents a pointer to the encoded file and has size independent of the file size. Moreover, we require that knowledge of the state information doesn't help a malicious server to violate the soundness property. Later, we formalize this notion by giving to the adversary oracle access to the encoding algorithm.

We consider PoS in which the verification phase requires three moves (as opposed to two as in previous work [2, 27, 3]): the server generates the first message a using the public key pk and randomness r; the verifier sends a random challenge c; and the server returns a proof π using pk, the encoded file, the challenge and the randomness used to generate the first message a.

Definition 2 (Proof of storage). *A publicly-verifiable* proof of storage *(PoS) is a tuple of six* ppt *algorithms* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comm}, \mathsf{Chall}, \mathsf{Prove}, \mathsf{Vrfy})$ *such that:*

$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$ *is a probabilistic algorithm that is run by the client to set up the scheme. It takes as input a security parameter, and outputs a public and private key pair* (pk, sk)*.*

$(\vec{f}', st) \leftarrow \mathsf{Enc}_{sk}(\vec{f})$ *is a probabilistic algorithm that is run by the client in order to encode the file. It takes as input the secret key* sk*, and a file* $\vec{f}$ *viewed as an* n*-dimensional vector over a block space* $\mathbf{B} = \{0,1\}^{p(k)}$ *for some polynomial* $p(k)$ *(let* p *be the block size of* Π*). It outputs an encoded file* $\vec{f}'$ *and public state information* st *in* $\{0,1\}^{\ell_{st}(k)}$ *(let* ℓ_{st} *be the state information size of* Π*).*

$a \leftarrow \mathsf{Comm}(pk)$ *is a probabilistic algorithm run by the server to generate the first message. It takes as input the public key and outputs an initial message* a*.*

$c \leftarrow \mathsf{Chall}(pk)$ *is a probabilistic algorithm that takes as input the public key and outputs a challenge* c*.*

$\pi \leftarrow \mathsf{Prove}(pk, \vec{f}', r, c)$ *is a probabilistic algorithm that takes as input the public key* pk*, an encoded file* $\vec{f}'$*, a string* r*, and a challenge* c*. It outputs a proof* π*.*

$b := \mathsf{Vrfy}(pk, st, a, c, \pi)$ *is a deterministic algorithm that takes as input the public key* pk*, the state information* st*, the first message* a*, a challenge* c*, and a proof* π*. It outputs a bit, where '1' indicates acceptance and '0' indicates rejection.*

We say that Π *is correct if for all* $k \in N$*, all* (pk, sk) *output by* $\mathsf{Gen}(1^k)$*, all* $n \in N$ *and* $\vec{f} \in \mathbf{B}^n$*, all* $(\vec{f}', st)$ *output by* $\mathsf{Enc}_{sk}(\vec{f})$*, and all* c *output by* $\mathsf{Chall}(pk)$*, it holds that*

$$\Pr_{r_c, r_p}\left[\mathsf{Vrfy}\left(pk, st, \mathsf{Comm}(pk;\ r_c), c, \mathsf{Prove}(pk, \vec{f}', r_c, c;\ r_p)\right) = 1\right] = 1.$$

An important characteristic of a PoS is *locality* which requires that the running time of the Prove algorithm be polynomial in the security parameter (independent of the parameter n).

Locality effectively captures the server-side efficiency guarantee provided by a PoS and, as we will show in Section 3, is what allows us to meet the efficiency requirements of the BRM.

Informally, soundness of a PoS guarantees that if the verifier accepts the proof then the prover indeed has sufficient information to recover the entire original file $\vec{f}$. As noted in [2, 17, 27, 9], soundness can be formalized using the notion of a knowledge extractor [15, 4]. As in [3], we phrase our definition using the paradigm of "witness-extended emulation" [21].

Definition 3 (Soundness for a publicly-verifiable PoS)**.** *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comm}, \mathsf{Chall}, \mathsf{Prove}, \mathsf{Vrfy})$ *be a publicly-verifiable PoS. We say that* Π *is sound with knowledge error* $\varepsilon(k)$ *if there exists an expected polynomial-time knowledge extractor* $\mathcal{K}$ *such that for all adversaries* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ *where* $\mathcal{A}_0$ *is an oracle* ppt *algorithm and* $\mathcal{A}_1$ *is an interactive* ppt *algorithm involved in the following probabilistic experiment:*

1. **Key Stage:** *The challenger computes* $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$*. The adversary* $\mathcal{A}_0$ *takes as input* pk *and gets oracle access to* $\mathsf{Enc}_{sk}(\cdot)$*. Eventually,* $\mathcal{A}_0$ *outputs a tuple* $(\vec{f}, st_{\mathcal{A}})$ *and the challenger computes* $(\vec{f}', st) \leftarrow \mathsf{Enc}_{sk}(\vec{f})$*.*
2. **Extraction Stage:** *The extractor* $\mathcal{K}$ *takes as input* pk *and* st *and gets access to the oracle* $\mathcal{A}_1(st_{\mathcal{A}}, \vec{f}', st, \ \cdot\ ;\ \cdot\ )$ *modeled as an interactive oracle. Finally* $\mathcal{K}$ *outputs the tuple* $((a, c, \pi), \vec{f}^*)$*.*
3. *The output of the experiment is the tuple* $(pk, st, (a, c, \pi), \vec{f}^*, \vec{f})$*.*

The properties listed below hold:

i) The following probability is at most $\varepsilon(k)$:

$$\Pr\left[\mathsf{Vrfy}(pk, st, a, c, \pi) = 1 \bigwedge \vec{f}^* \neq \vec{f}\right], \tag{1}$$

where the probability is over the outputs of the experiment above.

ii) For any pk and st, the distribution (a', c', π') induced by an execution of $\mathcal{A}_1(st_{\mathcal{A}}, \vec{f'}, st)$ with an honest verifier and the distribution (a, c, π) as output by the extractor $\mathcal{K}$ in the experiment above are identically distributed.

We say that Π is sound if $\varepsilon(k)$ is negligible.

For simplicity, we consider only PoS Π where the function Enc is injective for any sk and for any assignment of the internal randomness. This assumption is made without loss of generality, in fact any PoS scheme can be converted into one with this property by "appending the missing data" in the encoded file. By the soundness property, the procedure is efficient and the average size of the encoded file increases only by a negligible factor [1].

We now turn to our definition of zero-knowledge. Namely, we consider the notion of black-box zero-knowledge which guarantees that there exists a simulator for any adversary and the simulator has only black-box oracle access to the adversary's algorithm.

Definition 4 (zero-knowledge)**.** *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comm}, \mathsf{Chall}, \mathsf{Prove}, \mathsf{Vrfy})$ be a publicly-verifiable PoS. Π is δ-zero-knowledge (δ-ZK) if there is an expected polynomial time* transcript *simulator $\mathcal{S}$ such that for all non-uniform polynomial time adversaries $\mathcal{A}$, for any $n \geq 0$, for any $\vec{f} \in \mathbf{B}^n$ and for any infinite sequence $\mathcal{L} = \{(pk, sk, \vec{f'}, st)\}_{k \geq 0}$ indexed by the security parameter k and where (pk, sk) is output by $\mathsf{Gen}(1^k)$ and $(\vec{f'}, st)$ is output by $\mathsf{Enc}_{sk}(\vec{f})$, the distribution ensemble*

$$\left\{(a', c', \pi') \leftarrow \mathcal{S}^{\mathcal{A}}(st, pk, sk)\right\}_{(pk, sk, \vec{f'}, st) \in \mathcal{L}}$$

is $\delta(k)$-indistinguishable from the following distribution ensemble:

$$\left\{ (a, c, \pi) : \begin{array}{l} r \leftarrow \{0,1\}^*; a := \mathsf{Comm}(pk; r); \\ c \leftarrow \mathcal{A}(pk, st, a); \\ \pi \leftarrow \mathsf{Prove}(pk, \vec{f'}, r, c) \end{array} \right\}_{(pk, sk, \vec{f'}, st) \in \mathcal{L}}.$$

In the definition above, the secret key for the PoS is given as input to the simulator. We could consider a stronger definition where the secret key is given to the distinguisher, but we dismissed this option since a weaker zero-knowledge requirement makes our final compiler more general.

2.3 Identification Protocols

An identification protocol allows a prover $\mathcal{P}$ in possession of a secret key sk to prove its identity to a verifier $\mathcal{V}$ that holds the corresponding public key pk.

[1] To see this, consider the procedure that first encodes using Enc, then runs internally the extractor with oracle access to the honest prover and, if the extractor fails, appends the original file to the encoding. Since the extractor fails only with negligible probability the average size of the encoded file increases only by a negligible factor.

We consider 3-move identification protocols where the prover generates the first message α using the public key pk and randomness r; the verifier sends a random challenge β; and the prover then computes a response γ using (pk, sk), the randomness r and the verifier's challenge β. Given the transcript of the protocol, the verifier decides whether to accept or not. The prover algorithm of any identification scheme in the BRM must have efficiency essentially independent of the size of the secret key. This is captured by the following definition.

Definition 5 (Identification protocol in BRM)**.** *A* 3*-move* identification protocol *is a protocol between a* ppt *prover* $\mathcal{P}$ *and a* ppt *verifier* $\mathcal{V}$ *that consists of five polynomial-time algorithms* $\Sigma = (\mathsf{Setup}, \mathsf{Comm}, \mathsf{Chall}, \mathsf{Resp}, \mathsf{Vrfy})$ *such that:*

$(pk, sk) \leftarrow \mathsf{Setup}(1^k, 1^s)$ *is a probabilistic algorithm that takes as input the security parameter and the key-size parameter and outputs a public and private key pair* (pk, sk) *such that* $|pk| = \mathsf{poly}(k)$ *and* $|sk| = \mathsf{poly}(k, s)$*.*

$\alpha \leftarrow \mathsf{Comm}(pk)$ *is a probabilistic algorithm run by the prover* $\mathcal{P}$ *to generate the first message. It takes as input the public key and outputs an initial message* α*.*

$\beta \leftarrow \mathsf{Chall}(pk)$ *is a probabilistic algorithm run by the verifier* $\mathcal{V}$ *that takes as input the public key and outputs a challenge* β*.*

$\gamma \leftarrow \mathsf{Resp}(pk, sk, r, \beta)$ *is a probabilistic algorithm that is run by the prover* $\mathcal{P}$ *to generate the second message. It takes as input the public key pk, the secret key sk, the randomness r, and a challenge* β *(from some associated challenge space), and outputs a response* γ*.*

$b := \mathsf{Vrfy}(pk, \alpha, \beta, \gamma)$ *is a deterministic algorithm run by the verifier* $\mathcal{V}$ *to decide whether to accept the interaction. It takes as input the first message* α*, the public key pk, a challenge* β*, and a response* γ*. It outputs a bit b, where '1' indicates acceptance and '0' indicates rejection.*

The following properties hold:

Correctness. *For all* $k \in N$*, all* $s \in N$*, all* (pk, sk) *output by* $\mathsf{Setup}(1^k, 1^s)$*, and* β *output by* $\mathsf{Chall}(pk)$*, it holds that*

$$\Pr_{r,r'}\left[\mathsf{Vrfy}\left(pk, \mathsf{Comm}(pk; r), \beta, \mathsf{Resp}(pk, sk, r, \beta; r')\right) = 1\right] = 1.$$

Efficiency. *The prover* $\mathcal{P}$ *has running time* $\mathsf{poly}(k, \log s)$*. We call the* locality *of the protocol the number of bits of the secret key read as a function of the security parameter k.*

By saying "run the protocol Σ*" we refer to the execution of the protocol between* $\mathcal{P}$ *and* $\mathcal{V}$*.*

As in previous work [1, 18], we model leakage attacks by providing the adversary with access to a leakage oracle that returns arbitrary bits of information related to the secret key. Since we are working in the BRM, we require that the oracle returns at most λ bits.

Definition 6 (Leakage oracle)**.** *A leakage oracle* $\mathsf{Leak}_{sk}^{\lambda,k}(\cdot)$ *is parameterized by a secret key sk, a security parameter k and a leakage parameter* λ*. It takes as input a function f (specified as a circuit) and returns* $f(sk)$ *subject to the restriction that the total output length of all its replies is at most* λ*, otherwise it outputs* $\perp$*.*

Roughly speaking, security for identification schemes requires that an adversary should not convince an honest verifier to accept an interaction unless it knows the secret key corresponding to a given public key. In the case of *security against impersonation under active attacks*, this should hold even if the adversary is previously allowed to interact with the honest prover a polynomial number of times. In [1], Alwen *et al.* extend this notion to capture leakage attacks by providing the adversary with a $\mathsf{Leak}^{\lambda,k}_{sk}(\cdot)$ oracle. This leads to two definitions: security against pre-impersonation leakage, where the adversary can only access the oracle *before* interacting with the verifier; and security against anytime leakage, where the adversary can access the oracle even *during* the interaction with the verifier.

Definition 7 (Security against pre-impersonation leakage [1])**.** *Let Σ be an identification protocol and $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ be an adversary. Consider the following experiment:*

1. **Key Stage:** *The challenger computes $(pk, sk) \leftarrow \mathsf{Setup}(1^k, 1^s)$.*
2. **Test Stage:** *The adversary $\mathcal{A}_0$ takes as input pk and gets oracle access to $\mathsf{Leak}^{\lambda,k}_{sk}(\cdot)$ and to an honest prover $\mathcal{P}(sk, pk)$, modeled as an oracle that runs (arbitrarily many) proofs upon request; access to proofs is sequential. Finally $\mathcal{A}_0$ outputs $st_{\mathcal{A}}$.*
3. **Impersonation Stage:** *$\mathcal{A}_1(st_{\mathcal{A}})$ executes Σ as a prover with an honest verifier (running with pk).*
4. *The adversary succeeds if the honest verifier accepts the interaction.*

Σ is $\varepsilon(k)$-secure against pre-impersonation leakage $\lambda(k, s)$ if the success probability of every ppt *adversary $\mathcal{A}$ and for infinitely many positive integer s in the above experiment is at most $\varepsilon(k)$. We say that Σ is secure against pre-impersonation leakage $\lambda(k, s)$ if $\varepsilon(k)$ is negligible.*

3 From Proofs of Storage to Leakage-Resilient ID Protocols

In this section we show how to transform any computationally ZK publicly-verifiable proof of storage into a leakage-resilient identification protocol in the BRM. The basic idea is to use the file as the secret key of the identification protocol and the state information as its public key. A basic version of this approach would work as follows. The honest prover generates a public and private key pair for the PoS. A file is chosen at random and encoded. The encoded file $\vec{f'}$ serves as the identification secret key, and the state information *st* together with the public key of the PoS serves as the public key. To identify itself, the prover executes the verification phase of the PoS with the verifier.

One problem with the above approach is that, in the context of a pre-impersonation leakage attack, the adversary receives access to a $\mathsf{Leak}^{\lambda,k}_{\vec{f'}}(\cdot)$ oracle and to an honest prover. The effect of the leakage oracle can be mitigated somewhat by increasing the size of the file to be larger than λ. Since the communication complexity of the PoS is effectively constant, this will not degrade the efficiency of the protocol. However, to prevent the adversary's interaction with the honest prover from revealing too much information about the file, we will require the verification phase of the PoS to be zero-knowledge.

The compiler is shown in Figure 1. If the Prove algorithm of Π is local then the resulting scheme is an identification scheme in the BRM. We recall here a lemma from [1] that we make use of.

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Comm}, \mathsf{Prove}, \mathsf{Vrfy})$ be a PoS with block size $p(k)$. Construct a leakage-resilient ID protocol $\Sigma = (\mathsf{Setup}, \mathsf{Comm}, \mathsf{Resp}, \mathsf{Vrfy})$ as follows:

- $\mathsf{Setup}(1^k, 1^s)$:
  Set $n = s/p(k)$;
  Compute $(pk', sk') \leftarrow \Pi.\mathsf{Gen}(1^k)$ and sample a file $\vec{f} \leftarrow \mathbf{B}^n$;
  Compute $(\vec{f'}, st) \leftarrow \Pi.\mathsf{Enc}_{sk}(\vec{f})$ and
  set $sk = \vec{f'}$ and $pk = (pk', st)$; Delete sk' and $\vec{f}$.
- $\mathsf{Comm}(pk; r)$: Output $\alpha := \Pi.\mathsf{Comm}(pk; r)$.
- $\mathsf{Chall}(pk)$: Output $\beta \leftarrow \Pi.\mathsf{Chall}(pk)$.
- $\mathsf{Resp}(pk, sk, r, \beta)$: Output $\gamma := \Pi.\mathsf{Prove}(pk', \vec{f'}, r, \beta)$.
- $\mathsf{Vrfy}(pk, \alpha, \beta, \gamma)$: Output $b := \Pi.\mathsf{Vrfy}(pk', st, \alpha, \beta, \gamma)$.

Figure 1: Transforming a ZK PoS with block size $p(k)$ into a leakage-resilient ID protocol.

Lemma 1. *For any random variable X and for any experiment $\mathbf{E}$ with oracle access to $\mathsf{Leak}_X^{\lambda}(\cdot)$, consider the experiment $\mathbf{E}'$ which is the same as $\mathbf{E}$ except that the predictor does not have oracle access to $\mathsf{Leak}_X^{\lambda}(\cdot)$, then $\widetilde{\mathbf{H}}_\infty (X \mid \mathbf{E}) \geq \widetilde{\mathbf{H}}_\infty (X \mid \mathbf{E}') - \lambda$.*

Let $\mathbf{E}$ be the following randomized experiment:

1. It generates a key pair (pk', sk') for Π, samples a file $\vec{f}$ uniformly at random, samples random coins ω_{enc} and computes $(\vec{f'}, st) := \mathsf{Enc}_{sk'}(\vec{f}; \omega_{enc})$.
2. The predictor takes as input $pk = (pk', st), sk'$ and ω_{enc} and gets oracle access to $\mathsf{Leak}_{\vec{f'}}^{\lambda,k}(\cdot)$.

Lemma 2. *Let ℓ_{st} be the size of the state of Π. Then, $\widetilde{\mathbf{H}}_\infty(\vec{f'} \mid \mathbf{E}) \geq |\vec{f}| - \lambda - \ell_{st}$.*

Proof. Consider the experiment $\mathbf{E}'$ which is the same as $\mathbf{E}$ except that $\mathcal{B}$'s oracle access to $\mathsf{Leak}_{\vec{f'}}^{\lambda,k}$ is removed. We apply Lemma 1:

$$\widetilde{\mathbf{H}}_\infty \left(\vec{f'} \mid \mathbf{E}\right) \geq \widetilde{\mathbf{H}}_\infty \left(\vec{f'} \mid \mathbf{E}'\right) - \lambda,$$

Consider the experiment $\mathbf{E}''$ which is the same as $\mathbf{E}'$ but where the predictor does not get the state information st as input. We apply Lemma 1:

$$\widetilde{\mathbf{H}}_\infty \left(\vec{f'} \mid \mathbf{E}'\right) \geq \widetilde{\mathbf{H}}_\infty \left(\vec{f'} \mid \mathbf{E}''\right) - \ell_{st}.$$

Notice that in the experiment $\mathbf{E}''$ the information about $\vec{f'}$ is limited to sk and ω_{enc} and recall that $\mathsf{Enc}_{sk}(\cdot; \omega_{enc})$ is injective, thus any predictor guesses $\vec{f'}$ with probability $2^{-|\vec{f}|}$. □

In the next lemma we give an upper bound on the average conditional min entropy of $\vec{f'}$ given the experiment $\mathbf{E}$ that depends on the winning probability of a ppt adversary in the pre-impersonation leakage experiment.

Lemma 3. *Let Π be a δ-ZK PoS with knowledge error ε_Π and let $\varepsilon_\mathcal{A}$ be the probability with which an adversary $\mathcal{A}$ succeeds in the pre-impersonation leakage experiment. If δ is negligible then*

$$\widetilde{\mathbf{H}}_\infty(\vec{f'} \mid \mathbf{E}) \leq \log(1/\varepsilon_\mathcal{A}) + 2\frac{\varepsilon_\Pi}{\varepsilon_\mathcal{A}} + 1.$$

Proof. Consider the predictor $\mathcal{B}$ that, given the public key $pk = (pk', st)$ and sk', ω_{enc} works as follows during the experiment $\mathbf{E}$:

1. **Setup Stage:** It chooses a string ω for $\mathcal{A}_0$ that maximizes the winning probability of $\mathcal{A}$ in the pre-identification leakage experiment. Let $\mathcal{A}^\omega$ be the algorithm $\mathcal{A}_0$ with the randomness fixed to ω.

2. **Test Stage:** It executes $\mathcal{A}^\omega(pk)$ and answers its leakage queries using its own leakage oracle. At the i-th oracle call of $\mathcal{A}^\omega$ to the prover oracle, it executes the simulator $(a'_i, c'_i, \pi'_i) \leftarrow \mathcal{S}^{\mathcal{A}^\omega_i}(sk)$, where $\mathcal{A}^\omega_i$ is a copy of the adversary $\mathcal{A}^\omega$ where the machine state is set as the machine state of $\mathcal{A}^\omega$ just before the i-th call. The messages a'_i and π'_i are sequentially fed to the adversary $\mathcal{A}^\omega$. Eventually, $\mathcal{A}^\omega$ outputs $st_\mathcal{A}$.

3. **Extraction Stage:** It uses the extractor $\mathcal{K}(pk', st)$, guaranteed to exist by the soundness of Π, with $\mathcal{A}_1(st_\mathcal{A})$ to recover a file $\vec{f}^*$. It returns as its output $\mathsf{Enc}_{sk}(\vec{f}^*; \omega_{enc})$.

$\mathcal{A}^\omega$ is deterministic thus, for all i at the i-th interaction with the prover, $\mathcal{A}^\omega$ will reply with the challenge message c_i equal to the one in the simulated transcript. To bound the probability that the extractor $\mathcal{K}$ outputs the correct file, we first argue that the probability with which $\mathcal{A}_1$ succeeds in the impersonation stage is roughly the same whether it receives its state from a $\mathcal{A}^\omega$ that was executed with oracle access to an honest prover or to a simulator.

Proposition 1. *Let $q(k)$ (resp. $q'(k)$) be an upper bound on the number of queries made by $\mathcal{A}^\omega$ to the prover oracle (resp. leakage oracle). The view of $\mathcal{A}^\omega$ in the Test Stage of the predictor $\mathcal{B}$, as described below,*

$$\left\{pk, (a'_i, c'_i, \pi'_i)_{i\in[q(k)]}, (f_i(\vec{f'}))_{i\in[q'(k)]}\right\},$$

and the view of $\mathcal{A}^\omega$ in the Test Stage of the pre-impersonation leakage experiment

$$\left\{pk, (a_i, c_i, \pi_i)_{i\in[q(k)]}, (f_i(\vec{f'}))_{i\in[q'(k)]}\right\},$$

where, for all $i \in [q(k)]$, the tuple (a_i, c_i, π_i) is a transcript of the interaction between $\mathcal{A}^\omega_i$ and the honest prover, are $(q(k)\delta(k))$-indistinguishable.

The proposition can be proved with a hybrid argument based on the zero-knowledge property of the PoS. Indeed, the zero-knowledge property holds for any non-uniform polynomial-time adversary $\mathcal{A}$.

Recall that $\mathcal{A}^\omega$ at the end of the test stage outputs the state information $st_\mathcal{A}$. The probability that $\mathcal{A}_1(st_\mathcal{A})$ succeeds in the impersonation stage is at least $\varepsilon_\mathcal{A} - q\delta \geq \frac{\varepsilon_\mathcal{A}}{2}$. This holds because δ is negligible in k and by Proposition 1. In fact, if this were not the case, the concatenation of $\mathcal{A}^\omega$ and $\mathcal{A}_1(st_\mathcal{A})$ executing Π as prover with an honest verifier would distinguish the two distributions with noticeable probability.

Now, we can bound the probability that the extractor $\mathcal{K}$ outputs the correct file. From the soundness of Π, the extractor $\mathcal{K}$ outputs a tuple $((a, c, \pi), \vec{f}^*)$ such that $\mathsf{Vrfy}(pk, a, c, \pi) = 1$ and $\vec{f}^* \neq \vec{f}$ with probability at most $\varepsilon_\Pi(k)$. But note that

$$\begin{aligned} &\Pr[\mathsf{Vrfy}(pk, a, c, \pi) = 1 \wedge \vec{f}^* \neq \vec{f}] \\ &\quad \geq \Pr[\mathsf{Vrfy}(pk, a, c, \pi) = 1] - \Pr[\vec{f}^* = \vec{f}] \geq \ \tfrac{\varepsilon_\mathcal{A}}{2} - \Pr[\vec{f}^* = \vec{f}]. \end{aligned}$$

Hence, it follows that

$$\Pr[\vec{f}^* = \vec{f}] \ \geq \ \tfrac{\varepsilon_\mathcal{A}}{2} - \varepsilon_\Pi \ = \ \tfrac{\varepsilon_\mathcal{A}}{2}\left(1 - \tfrac{2\varepsilon_\Pi}{\varepsilon_\mathcal{A}}\right) > \tfrac{\varepsilon_\mathcal{A}}{2} \cdot 2^{-2\frac{\varepsilon_\Pi}{\varepsilon_\mathcal{A}}} = \varepsilon_\mathcal{A} \cdot 2^{-2\frac{\varepsilon_\Pi}{\varepsilon_\mathcal{A}} - 1},$$

where we used $(1 - x) \geq e^{-x} > 2^{-2x}$. The lemma follows because of Eq.(2) below and by taking the log:

$$2^{-\widetilde{\mathbf{H}}_\infty(\vec{f'}|\mathbf{E})} \geq \Pr\left[\mathcal{B}^\mathbf{E} = \vec{f'}\right] \geq \Pr\left[\mathsf{Enc}_{sk}(\vec{f}^*; \omega_{enc}) = \vec{f'}\right] = \Pr\left[\vec{f}^* = \vec{f}\right]. \qquad (2)$$

□

We are now ready to prove our main theorem which establishes the security of our transformation.

Theorem 1. *Let Π be a proof of storage that is sound with knowledge error $\varepsilon_\Pi(k)$, computational $\delta(k)$-zero-knowledge and with state information size $\ell_{st}(k)$. If $\delta(k)$ and $\varepsilon_\Pi(k)$ are negligible in k and if $|f| > \lambda + \ell_{st} + \omega(\log k)$, then Σ as in Figure 1 is secure against pre-impersonation leakage λ.*

Proof. Let $\varepsilon_\mathcal{A}$ be the pre-impersonation leakage winning probability of an adversary $\mathcal{A}$, since ε_Π and δ are negligible in k, by Lemma 3:

$$\widetilde{\mathbf{H}}_\infty(\vec{f'}|\mathbf{E}) \leq -\log(1/\varepsilon_\mathcal{A}) + \mathsf{negl}(k) + 1.$$

It follows then that if $\varepsilon_\mathcal{A}$ is noticeable in k, there exists a constant c such that

$$\widetilde{\mathbf{H}}_\infty(\vec{f'}|\mathbf{E}) \leq c \cdot \log(k) \qquad (3)$$

for infinitely many k. Thus, if $|\vec{f}| > \lambda + \ell_{st} + \omega(\log k)$, Equation 3 contradicts Lemma 2. □

4 A ZK-PoS based on RSA

We now describe a (statistical) zero-knowledge proof of storage. The scheme, described in Figure 2, is an extension of the RSA-based construction of Ateniese *et al.* [2]. It relies on a modulus generator Gen_Q that takes as input a security parameter 1^k and outputs a tuple (N, p', q') such that $N = (2p' + 1) \cdot (2q' + 1) = p \cdot q$, where p' and q' are random primes such that $p'q' \in [2^{k-1}, 2^k - 1]$ and p and q are primes.

Abstractly, the scheme can be seen as a witness-indistinguishable Sigma protocol (see Cramer [7]) for the relation:

$$\mathcal{R} = \left\{ \left((pk, st, \vec{c}),\ (\tilde{t}, \tilde{f})\right) \,\middle|\, \frac{\tilde{t}^e}{\prod_i H(st, i)^{c_i}} \equiv g_1^{\tilde{f}} \bmod N \right\},$$

```
$\mathsf{Gen}(1^k)$: Set $\bar{k} = \omega(\log k)$ and generate $(N, p', q') \leftarrow \mathsf{Gen}_Q(1^{k+5\bar{k}})$.
    Choose a prime $e$ such that $e > 2^{k+5\bar{k}}$ and $d$ such that $ed = 1 \pmod{p'q'}$. Let $g_1$ and $g_2$
    be generators of the unique cyclic subgroup $\mathcal{Q}_N$ of order $p'q'$ (i.e., the set of quadratic
    residues modulo $N$). Let $H : \{0,1\}^* \rightarrow \mathcal{Q}_N$ be a RO. Set $pk = (N, g_1, g_2, e, H)$ and
    $sk = (N, d, H)$.
    The block space $\mathbf{B}$ is $\mathbb{Z}_{2^k}$ and the challenge space $\mathbf{C}$ for $n$-block long file is $\mathbb{Z}^n_{2^{\bar{k}}} \times \mathbb{Z}_{2^{\bar{k}}}$.

$\mathsf{Enc}_{sk}(\vec{f})$:
    1. sample $st \leftarrow \{0,1\}^k$.
    2. for $1 \leq i \leq n$:
        (a) set $r_i := H(st, i)$.
        (b) compute $t_i := \left(r_i \cdot g_1^{f_i}\right)^d \bmod N$.
    3. let $\vec{t} := (t_1, \ldots, t_n)$
    4. output the encoded file $\vec{f'} := (\vec{f}, \vec{t})$ and state information $st$.

$\mathsf{Comm}(pk)$:
    sample $z_1 \leftarrow \mathbb{Z}_{2^{k+4\bar{k}}}$ and $z_2 \leftarrow \mathbb{Z}_{2^{k+8\bar{k}}}$ and
    output $a := g_1^{z_1} \cdot g_2^{e \cdot z_2} \bmod N$

$\mathsf{Chall}(pk)$:
    sample $\vec{c} \leftarrow \mathsf{Sparse}(\mathbb{Z}_{2^{\bar{k}}}, n, m)$ and $v \leftarrow \mathbb{Z}_{2^{\bar{k}}}$ and
    output $c := (\vec{c}, v)$.

$\mathsf{Prove}(pk, \vec{f'}, a, c)$:
    1. parse $c$ as $\vec{c} \in \mathbb{Z}^n_{2^{\bar{k}}}$ and $v \in \mathbb{Z}_{2^{\bar{k}}}$
    2. sample $\rho \leftarrow \mathbb{Z}_{2^{k+6\bar{k}}}$
    3. compute $\tau := g_2^{\rho} \cdot \Pi_i t_i^{c_i} \bmod N$
    4. compute $\mu := z_1 + v \cdot \sum_i c_i \cdot f_i$
    5. compute $\sigma := z_2 + v \cdot \rho$
    6. output $\pi := (\tau, \mu, \sigma)$

$\mathsf{Vrfy}(pk, st, \vec{c}, \pi)$:
    1. for $1 \leq i \leq n$, set $r_i := H(st, i)$
    2. output 1 iff $\mu < 2^{k+5\bar{k}}$ and $a \cdot (\tau^e / \Pi_i r_i^{c_i})^v \overset{?}{\equiv} g_1^{\mu} \cdot g_2^{e \cdot \sigma} \pmod N$
```

Figure 2: A statistical ZK PoS based on RSA with locality parameter m.

where $pk = (N, g_1, g_2, e, H)$ as defined in Figure 2, and where the equation that defines the relation $\mathcal{R}$ is essentially the verification procedure of the PoS presented in [2]. We note that for any file $\vec{f} \in \mathbf{B}^n$ and any challenge $\vec{c} \in \mathbb{Z}^n_{2^{\bar{k}}}$, let $\vec{f'}, st \leftarrow \mathsf{Enc}_{sk}(\vec{f})$ where $\vec{f'} = (\vec{f}, \vec{t})$, a witness for the instance $(pk, st, \vec{c})$ can be derived as

$$\tilde{t} = \prod_i t_i^{c_i} \quad \text{and} \quad \tilde{f} = \sum_i c_i \cdot f_i. \tag{4}$$

The witness indistinguishability property of the Sigma protocol is enough to derive the zero-knowledge property of the PoS. Witness indistinguishability means that the distributions of the transcript for two distinct witnesses are indistinguishable, even when the verifier is malicious. Recall that the simulator of ZK-PoS takes as input the secret key $sk = (N, d, H)$, and thus it can efficiently derive a valid witness (t', f') for the instance $(pk, st, \vec{c})$ for any challenge $\vec{c}$ chosen by the adversary. Specifically, it can encode an uniformly random file (or even a fixed one) using the same state information and compute an honest proof of storage for the challenge $\vec{c}$ and the encoded file[2]. Notice that we are assuming that the first message of the Sigma protocol is independent from the witness, which is usually true for Sigma protocols.

The locality of the scheme depends on how the challenges are generated. In fact, to make the scheme local it is enough to use "probabilistic checking" and make the server generate a proof for a random subset of the blocks. More concretely, we define a distribution $\mathsf{Sparse}(\mathbb{Z}_{2^{\bar{k}}}, n, m)$ by sampling a vector $\vec{c}$ such that for all $i \in [n]$: (1) with probability m/n the element $\vec{c_i}$ is chosen uniformly at random from $\mathbb{Z}_{2^{\bar{k}}}$; otherwise (2) $\vec{c_i}$ is set to 0. For locality m the challenge is sampled from the distribution $\mathsf{Sparse}(\mathbb{Z}_{2^{\bar{k}}}, n, m)$. This ensures that Prove and Vrfy have locality m *on average.* If the scheme needs to be *always* local, the honest-prover can just discard the challenge if the number of non-zero locations in $\vec{c}$ is not in the range $\{(1 \pm \varepsilon)m\}$, for a constant ε. The behavior will be indistinguishable from the original scheme with all but negligible probability in k.

Theorem 2. *The scheme described in Figure 2 is statistical zero-knowledge.*

Proof. For any adversary $\mathcal{A}$, consider the simulator $\mathcal{S}^{\mathcal{A}}$ that on input the key pair (N, g_1, g_2, e, d, H) samples $a \leftarrow \mathsf{Comm}(pk)$, then executes $(\vec{c}, v) \leftarrow \mathcal{A}(pk, st, a)$. If $\mathcal{A}$ aborts then the simulator returns the special symbol $\perp$. Otherwise, with the knowledge of the secret key, the simulator computes $v' := v^{-1} \bmod p'q'$ and samples an element μ in $\mathbb{Z}_{2^{k+4\bar{k}}}$, an element σ in $\mathbb{Z}_{2^{k+8\bar{k}}}$ and sets

$$\tau := \left(\left(g_1^{\mu} \cdot g_2^{\sigma} \cdot a^{-1} \right)^{v'} \cdot \Pi_i r_i^{c_i} \right)^{d} \bmod N$$

where $r_i := H(st, i)$, and outputs the tuple $(st, a, (\vec{c}, v), (\tau, \mu, \sigma))$.

The output of $\mathcal{S}^{\mathcal{A}}$ is statistically close to a real transcript since a, v and $\vec{c}$ are distributed exactly as they would be in a real transcript, and since τ, μ, and σ are statistically close to elements from a real transcript. Moreover by definition $v < p'$ and $v < q'$, thus the element $v^{-1} \bmod p'q'$ is well defined. □

Theorem 3. *For locality parameter $m = \omega(\log k)$, the scheme described in Figure 2 is sound if the RSA assumption holds with respect to* Gen_Q.

Proof. We describe a knowledge extractor $\mathcal{K}$ that runs in expected polynomial-time and satisfies Definition 3. Recall that $\mathcal{K}$ is given (pk, st) as input and has oracle access to $\mathcal{A}_1(st_{\mathcal{A}}, \vec{f'}, st, \cdot\,; \cdot)$ which we abbreviate as $\mathcal{A}(\cdot)$. $\mathcal{K}$ works as follows:

1. It chooses a random challenge $c := (\vec{c}, v)$ and runs $\mathcal{A}$ on c, obtaining a first message a and a proof π. If $\mathsf{Vrfy}(pk, st, a, c, \pi) = 0$, $\mathcal{K}$ outputs $(\tau, \perp)$ and halts. Otherwise, its first output will still be τ but it attempts to recover the original file as described next. From now on, we assume that $\mathcal{A}$ will be rewound to right after it outputs its first message a so that it can be challenged on distinct challenge pairs. We sometimes denote the adversary that outputs a as

[2] The actual simulator does it implicitly, without sampling the entire file.

its first message as $\mathcal{A}_a$ and write $\pi \leftarrow \mathcal{A}_a(\vec{c}, v)$ to refer to the proof it outputs when given challenge $(\vec{c}, v)$.

2. It initializes a set $\mathsf{Basis} = \emptyset$, keeps track of the total number of calls to $\mathcal{A}$ and halts with output fail if 2^k calls are made.

3. A challenge pair $(\vec{c}, v)$ is *valid* if $\mathcal{A}_a(\vec{c}, v)$ outputs π such that $\mathsf{Vrfy}(pk, st, a, (\vec{c}, v), \pi) = 1$. $\mathcal{K}$ estimates the probability $\tilde{\varepsilon}$ with which a pair $(\vec{c}, v)$ is valid by running $\mathcal{A}_a$ with a random challenge until some fixed polynomial number $t = t(k)$ of successful verifications occur. By appropriate choice of t it is possible to ensure that $\tilde{\varepsilon}$ is within a factor of 2 of the true probability with all but negligible probability 2^{-k^2}.

4. For $j = 1$ to n do:
   - Repeatedly sample the pair $(\vec{c}_j, v_j) \leftarrow \mathsf{Chall}(pk)$ until:
     (a) $\vec{c}_j$ does not lie in $\mathsf{span}(\mathsf{Basis})$;
     (b) The pair $(\vec{c}_j, v_j)$ is valid;
     (c) Sample $4k/\tilde{\varepsilon}$ random values $v_j^{(1)}, \ldots, v_j^{(4k/\tilde{\varepsilon})}$, and pick a value $v_j^* \in \left\{ v_j^{(1)}, \ldots, v_j^{(4k/\tilde{\varepsilon})} \right\}$ such that $v_j^* \neq v_j$ and that $(\vec{c}_j, v_j^*)$ is valid.
   - If no such tuple $(\vec{c}_j, v_j, v_j^*)$ is found within $16k/\tilde{\varepsilon}$ tries then output fail and halt. If found, add $\vec{c}$ to Basis.

5. Let $\mathsf{Basis} = \{\vec{c}_1, \ldots, \vec{c}_n\}$. Let $\pi_j = (\tau_j, \mu_j)$ and $\pi_j^* = (\tau_j^*, \mu_j^*)$ be the outputs of $\mathcal{A}_a(\vec{c}_j, v_j)$ and $\mathcal{A}_a(\vec{c}_j, v_j^*)$, respectively. Setup the system of linear equations
$$\left\{ \sum_i c_{j,i} \cdot f_i = (\mu_j - \mu_j^*)/(v_j - v_j^*) \right\}_{1 \leq j \leq n}$$
in the unknowns $\vec{f} = (f_1, \ldots, f_n)$. Solve for $\vec{f}$ (over the integers) and output it.

Fixing $st_{\mathcal{A}}$, $\vec{f'}$ and st, we let ε denote the probability that a random challenge $(\vec{c}, v)$ is valid. We assume $st_{\mathcal{A}}$ includes $\mathcal{A}$'s coins thus this corresponds to the probability with which $\mathcal{A}(st_{\mathcal{A}}, \vec{f'}, st, \cdot)$ responds correctly to the verifier's challenge.

We note that the first point of Definition 3 is satisfied. Indeed, distribution of transcripts generated by an honest verifier interacting with $\mathcal{A}$ is identical to the distribution of the first output of $\mathcal{K}$. In fact, $\mathcal{K}$ produces its first output by emulating an interaction between $\mathcal{A}$ and the honest verifier.

Claim 1. *$\mathcal{K}$ runs in expected polynomial time for any adversary $\mathcal{A}$.*

If $\varepsilon = 0$ then $\mathcal{K}$ halts in Step 1, thus assume $\varepsilon > 0$. Steps 1 and 5 run in strict polynomial time. The expected running time of Step 3 is exactly some polynomial times $t(k)/\varepsilon$. As for Step 4, there are two cases. If $\tilde{\varepsilon} \leq \varepsilon/2$ then the running time is bounded by some polynomial times 2^k due to the counter being maintained in Step 2. But the probability that $\tilde{\varepsilon} \leq \varepsilon/2$ is at most 2^{-k^2}.

On the other hand, if $\tilde{\varepsilon} > \varepsilon/2$, then the expected running time of Step 4 is at most some polynomial times $n \cdot 16k \cdot 4k/\tilde{\varepsilon} < n \cdot 128k^2/\varepsilon$. Since $\mathcal{K}$ only reaches Step 4 with probability ε, the overall expected running time of $\mathcal{K}$ is upper bounded by
$$\varepsilon \cdot \Big(\mathsf{poly}(k) + \mathsf{poly}(k) \cdot t(k)/p \ + \mathsf{poly}(k) \cdot 2^k \cdot 2^{-k^2} + \mathsf{poly}(k) \cdot n \cdot 128k^2/\varepsilon \Big)$$

which is polynomial.

□

Claim 2. *If $\varepsilon > 4 \cdot (2^{-\bar{k}} + e^{-m})$ then the probability (conditioned on $\mathcal{K}$ reaching Step 4) that $\mathcal{K}$ outputs* fail *is negligible. Observe that this implies that*

$$\Pr_{pk,st,a,\pi,\vec{f}^*}\left[\mathsf{Vrfy}(pk, st, a, c, \pi) = 1 \bigwedge \vec{f}^* = \mathsf{fail}\right],$$

where the probability is over the output of $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{PoS}}$, is negligible in the security parameter.

First, observe that the probability that $\mathcal{K}$ times out by virtue of running for 2^k steps is negligible (this follows from the fact that the expected running time of $\mathcal{K}$ is polynomial). Next, fix any j and consider Step 4.

We say that a vector $\vec{c}$ is good if there are at least a $\varepsilon/2$ fraction of v's for which $(\vec{c}, v)$ is valid. Let $\mathcal{E}_1$ be the event that a pair $(\vec{c}, v)$ is such that

$$\vec{c} \notin \mathsf{span}(\mathsf{Basis}) \bigwedge (\vec{c}, v) \text{ is valid} \bigwedge \vec{c} \text{ is good}$$

We claim that the probability that $\vec{c}$ lies in $\mathsf{span}(\mathsf{Basis})$ is at most $2^{-\bar{k}} + e^{-m}$.
The probability that $\vec{c}$ is bad and does not lie in $\mathsf{span}(\mathsf{Basis})$ is at most $(1 - 2^{-k})\varepsilon/2$. We therefore have that

$$\Pr[\mathcal{E}_1] \geq \varepsilon - (2^{-\bar{k}} + e^{-m}) - (1 - 2^{-k})\varepsilon/2 \geq \varepsilon/4 \tag{5}$$

where the last inequality holds since $\varepsilon > 4(2^{-\bar{k}} + e^{-m})$.

Now let $\mathcal{E}_2$ be the event that $v^* \neq v$ and that $(\vec{c}, v^*)$ is valid. Note that if $\vec{c}$ is good, there is at least a $2^k \cdot \varepsilon/2 - 1$ total number of v^*'s that are different from v and such that $(\vec{c}, v^*)$ is valid. Therefore, it follows that

$$\Pr[\mathcal{E}_2 \mid \vec{c} \text{ is good}] \geq \frac{2^k \cdot \varepsilon/2 - 1}{2^k} \geq \varepsilon/4$$

where the last inequality follows from the assumption that $\varepsilon > 2^{-k+2}$. The probability that (conditioned on $\vec{c}$ being good) $\mathcal{K}$ finds a $v^* \neq v$ such that $(\vec{c}, v^*)$ is valid within $4k/\tilde{\varepsilon}$ samples is at least $(1 - e^{-k/2})$ since $\tilde{\varepsilon} \leq 2\varepsilon$ with all but negligible probability in k.

Combined with Equation 5 we have that the probability that $\mathcal{K}$ succeeds in finding a tuple $(\vec{c}, v, v^*)$ such that $\vec{c} \notin \mathsf{span}(\mathsf{Basis})$ and that both $(\vec{c}, v)$ and $(\vec{c}, v^*)$ are valid is at least

$$(1 - 2^{-k^2}) \cdot (1 - e^{-k}) \cdot \varepsilon/4 \geq \varepsilon/16$$

since $k \geq 1$. It follows then that, in Step 4, $\mathcal{K}$ will not find such a tuple within $16k/\tilde{\varepsilon}$ iterations (and therefore outputs fail) with probability at most e^{-k}, which is negligible. This end the proof of the claim.

For completeness, we show that $\vec{c}$ lies in $\mathsf{span}(\mathsf{Basis})$ with probability $2^{-\bar{k}} + e^{-m}$.
Note that the larger $|\mathsf{Basis}|$ is, the most likely $\vec{c} \in \mathsf{span}(\mathsf{Basis})$, therefore the worst case is $|\mathsf{Basis}| = n - 1$. If $\vec{c} \in \mathsf{span}(\mathsf{Basis})$ then there exist $\alpha_1, \ldots, \alpha_n \in \mathbb{Z}$ with at least one coordinate i with $\alpha_i \neq 0$

and such that $\vec{c} \in V$ if and only if $E_n := (\sum_{i=0}^{n} \alpha_i \cdot \vec{c}_i = 0)$. W.l.o.g. let n be such a coordinate:

$$\begin{aligned}
\Pr\left[E_{n'}\right] &\leq \sum_{x \in \mathbb{F}} \cdot \Pr\left[\vec{c}_{n'} = x\right] \Pr\left[\Sigma_i^{n'-1} \alpha_i \vec{c}_i = -\alpha'_n x\right] \\
&\leq \Pr\left[\vec{c}'_n = 0\right] \Pr\left[E_{n'-1}\right] + \sum_{x \in \mathbb{F} \setminus \{0\}} \Pr\left[\vec{c}'_n = x\right] \cdot \Pr\left[\Sigma_i^{n'-1} \alpha_i \vec{c}_i = -\alpha'_n x\right] \\
&\leq \left(\frac{m \cdot 2^{-\bar{k}}}{n} + 1 - \frac{m}{n}\right) \Pr\left[E_{n'-1}\right] + \frac{m \cdot 2^{-\bar{k}}}{n} \cdot \sum_{x \in \mathbb{F} \setminus \{0\}} \Pr\left[\Sigma_i^{n'-1} \alpha_i \vec{c}_i = -\alpha'_n x\right] \\
&\leq \left(\frac{m \cdot 2^{-\bar{k}}}{n} + 1 - \frac{m}{n}\right) \Pr\left[E_{n'-1}\right] + \frac{m \cdot 2^{-\bar{k}}}{n} \cdot \left(1 - \Pr\left[E_{n'-1}\right]\right) \\
&\leq \quad \left(1 - \frac{m}{n}\right) \Pr\left[E_{n'-1}\right] + \frac{m \cdot 2^{-\bar{k}}}{n}
\end{aligned}$$

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

$$\begin{aligned}
\Pr\left[E_{n'}\right] &\leq \left(1 - \frac{m}{n}\right)^{n'} \Pr\left[E_0\right] + \frac{m \cdot 2^{-\bar{k}}}{n} \left(\sum_{i=0}^{n'-1} \left(1 - \frac{m}{n}\right)^{i}\right) \\
&\leq e^{-m} \Pr\left[E_0\right] + \frac{m \cdot 2^{-\bar{k}}}{n} \left(\frac{1 - (1 - \frac{m}{n})^{n'}}{m/n}\right) \ \leq \ e^{-m} + \frac{1}{r}.
\end{aligned}$$

□

Claim 3. *If RSA assumption holds then, for any* ppt *adversary $\mathcal{A}$,*

$$\Pr\left[\mathsf{Vrfy}(pk, st, a, c, \pi) = 1 \bigwedge \vec{f}^* \notin \{\mathsf{fail}, \vec{f}\}\right],$$

where the probability is over the output of $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{PoS}}$, is negligible in the security parameter.

As a sanity check, we show that if we run the extractor $\mathcal{K}$ on an honest prover then the procedure correctly outputs the original file $\vec{f}$. Given two honestly generated (therefore valid) proofs (τ, μ, σ) and $(\tau^*, \mu^*, \sigma^*)$, for $(\vec{c}, v)$ and $(\vec{c}, v^*)$ respectively, we obtain:

$$\tau^{ev} = (\Pi_i r_i^{c_i})^{v} \cdot a^{-1} \cdot g_1^{\mu} \cdot g_2^{e \cdot \sigma} \pmod{N}$$

$$(\tau^*)^{ev^*} = (\Pi_i r_i^{c_i})^{v^*} \cdot a^{-1} \cdot g_1^{\mu^*} \cdot g_2^{e \cdot \sigma^*} \pmod{N}$$

By dividing the two equations and using the definition of τ and τ^* we get

$$g^{\mu - \mu^*} = g^{(v - v^*) \cdot \sum_i c_i f_i} \pmod{N}$$

from which it follows that

$$(\mu - \mu^*) = (v - v^*) \cdot \sum_i c_i f_i \pmod{p'q'}.$$

The equation above, however, also holds over the integers since $|\mu - \mu^*| \leq 2^{k+5\bar{k}} < p'q'$ and $|(v - v^*) \cdot \sum_i c_i f_i| \leq 2^{k+\log(n)+2\bar{k}} < p'q'$. And since $v \neq v^*$ we have

$$(\mu - \mu^*)/(v - v^*) = \sum_i c_i f_i.$$

This ends the sanity check.

Note that if $\vec{f}^* \neq \mathsf{fail}$ then $\mathcal{K}$ reached Step 5 and therefore $(\vec{c}, v)$ and $(\vec{c}, v^*)$ are valid. Therefore $\vec{f}^* \neq \vec{f}$ occurs only if, at Step 5, there exists some $\vec{c} \in \mathsf{Basis}$ for which the challenges $(\vec{c}, v)$ and $(\vec{c}, v^*)$ and corresponding proofs $\pi = (\tau, \mu, \sigma)$ and $\pi^* = (\tau^*, \mu^*, \sigma^*)$ are such that

$$\mathsf{Vrfy}(pk, st, a, (\vec{c}, v), \pi) = \mathsf{Vrfy}(pk, st, a, (\vec{c}, v^*), \pi^*) = 1$$

yet

$$\sum_i c_i \cdot f_i^* = (\mu - \mu^*)/(v - v^*) \neq \sum_i c_i \cdot f_i. \tag{6}$$

We now argue that if this occurs with noticeable probability, then there exists a ppt adversary $\mathcal{B}$ that violates the RSA assumption with respect to Gen_Q. Let N and e be a modulus and exponent output by Gen_Q and let y be a random element of $\mathcal{Q}_N$. The adversary $\mathcal{B}$ works as follows:

1. It chooses a generator u of $\mathcal{Q}_N$ uniformly at random and set $g_1 := u^e \cdot y$ and $g_2 := y$ and $pk := (N, g_1, g_2, e, H)$.

2. It simulates the experiment in the claim by answering Enc and random oracle queries as follows:

   (Enc queries): given a file $\vec{f}$ compute a set of tags $\vec{t}$ such that $t_i = u^{w_i}$ where $w_i \leftarrow \mathbb{Z}_{N^2}$, choose a random $st \leftarrow \{0,1\}^k$ and keep track of $(\vec{f}, st)$. If st was already chosen in a previous query or $\mathcal{B}$ has already queried the Random Oracle on a value (st, i), abort the simulation. Otherwise, return $(\vec{f}, \vec{t})$.

   (RO queries): if query x has the form (st, i) for some st such that there already exists a record $(\vec{f}, st)$ and $i \in [n]$, return $u^{ew_i} \cdot g_1^{-f_i}$. Otherwise, return a random value in $\mathcal{Q}_N$ and save the query/answer pairs to answer queries consistently.

3. It runs the extractor $\mathcal{K}$ and finds a vector $\vec{c} \in \mathsf{Basis}$ such that the Equation 6 holds. Finally, it computes and returns

   $$\left(\tau^v/(\tau^*)^{v^*} \cdot u^{\Delta v\left(\sum_i c_i(f_i - w_i)\right) - \Delta\mu}\right)^\alpha y^\beta \pmod{N}$$

   Where $\Delta v := (v - v^*)$, $\Delta\mu := (\mu - \mu^*)$, $\Delta\sigma := (\sigma - \sigma^*)$ and set

   $$\Phi := -\Delta v \left(\sum_i c_i f_i\right) + \Delta\mu + \Delta\sigma \cdot e$$

   and α,β are such that:

   $$\alpha \cdot \Phi + \beta \cdot e = 1 \tag{7}$$

First note that the r_i values are statistical close to the uniform distribution over $\mathcal{Q}_N$ because w_i values are picked from $\mathbb{Z}_{N^2}$, thus the simulation is statistical close to the real game. Moreover, there are a polynomial number of queries therefore the aborting probability is negligible in k.

From the verification equations we get that

$$\left(\tau^v/(\tau^*)^{v^*}\right)^e \equiv (\Pi_i r_i^{c_i})^{\Delta v} \cdot g_1^{\Delta\mu} \cdot g_2^{e\cdot\Delta\sigma} \pmod{N}$$

Applying the definitions from the simulation we get:

$$\left(\tau^v/(\tau^*)^{v^*} \cdot u^{\Delta v(\sum_i c_i(f_i-w_i))-\Delta\mu}\right)^e \equiv y^{\Phi} \pmod{N}$$

Equation (7) holds here, in fact e is prime and divides Φ if and only if e divides $\Phi - \Delta\sigma \cdot e < e$ (recall that the verification procedure ensures that $|\Delta\mu| < 2^{k+5\bar{k}} < e$ and $|\Delta v \sum_i c_i f_i| < 2^{k+\log n+2\bar{k}} < e$), thus we can apply Shamir's trick to find $y^d \pmod{N}$.

Remark 1. *In order to simulate an RO from $\{0,1\}^*$ to $\mathcal{J}_N^+$ (the subspace of elements with Jacobi Symbol +1), we follow the same simulation and set $h := u^e \cdot y$, $g_1 := h^2$, $r_i := (-1)^b \cdot u^{ew_i} \cdot h^{-f_i}$ for a random bit b and $t_i := u^{2w_i}$,*

□

4.1 Efficiency Comparison with Previous Work

We compare the identification scheme derived by applying our transformation to the RSA-based ZK PoS from Section 4 with the third (and most efficient) construction of Alwen *et al.* [1]. In the following, we denote our construction by RSA-ID and that of Alwen *et al.* by GDH-ID.

We consider multiplications and additions as constant-time operations and denote by t_e the time for an exponentiation, by t_s the time for an exponentiation with a small (i.e., $o(k)$) exponent, and by t_p the time for a pairing operation. For the same security level, modular exponentiations in RSA groups are more expensive than modular exponentiations in groups for which GDH seems to hold, therefore we distinguish them by using the upper scripts RSA and GDH to indicate in which group the operations are carried out. We can assume that $t_e^{\mathsf{GDH}} < t_e^{\mathsf{RSA}} \ll t_p$.

In GDH-ID, the prover needs $\Omega(\ell \cdot m \cdot t_e^{\mathsf{GDH}})$ work to generate each of its two messages (the first and third) while the verifier needs $\Omega(m \cdot t_e^{\mathsf{GDH}} + t_p)$ time to verify the interaction[3] . For our construction, on the other hand, the prover needs only $O(t_e^{\mathsf{RSA}})$ (i.e., two exponentiations and one multiplication) and $O(t_e^{\mathsf{RSA}} + m \cdot t_s^{\mathsf{RSA}})$ work for the first and third messages, respectively, and the verifier requires only $O(t_e^{\mathsf{RSA}} + m \cdot t_s^{\mathsf{RSA}})$ time to verify the interaction. We also note that while the locality m in RSA-ID can be any function that is $\omega(\log k)$, in GDH-ID m must be at least $\Omega(k)$. In particular, to get approximately 1/2 tolerance of relative leakage, m must be 12 times larger than k.

With respect to communication complexity, the third message of GDH-ID requires roughly ℓ times the number of group element as the third message of RSA-ID—though GDH-ID works in smaller groups than RSA-ID for the same security parameter.

There are two negative aspects of RSA-ID compared with GDH-ID: The first is that, for the same security level, RSA groups are bigger than groups for which GDH seems to hold; The second is the ratio between the secret-key size and the leakage tolerated. However, the difference is relevant

[3] The integer parameter $\ell \geq 2$ in their construction can be arbitrarily set.

only when ℓ is $\omega(1)$ and m is $\omega(k)$ in which case the time complexity of GDH-ID becomes much worse than that of RSA-ID.

5 Conclusions

We showed that zero-knowledge proof-of-storage schemes can be used to build leakage-resilient identification protocols in the bounded retrieval model (BRM). Our framework provides new insights into the BRM and unfolds new ways to build leakage-resilient identification protocols in this model. For instance, we described a ZK-PoS based on RSA which yields the first ID protocol in the BRM based on RSA. When combined with the compiler in [3], our framework establishes a compelling connection between homomorphic ID and leakage-resilient ID schemes. However, the missing step toward an efficient compiler between homomorphic ID and leakage-resilient ID schemes is to find an efficient compiler between PoS and ZK-PoS. We do not explore any approach in this paper and leave it as an open problem.

Acknowledgments.

We are grateful to Jonathan Katz for his insightful comments, suggestions, and contributions to this work.

References

[1] J. Alwen, Y. Dodis, and D. Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *CRYPTO*, pages 36–54, 2009.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In *CCS*, 2007.

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

[4] M. Bellare and O. Goldreich. On defining proofs of knowledge. In *CRYPTO*, pages 390–420, 1992.

[5] D. Boneh and D. Brumley. Remote timing attacks are practical. In *12th Usenix Security Symposium*, 2003.

[6] D. Boneh, B. Lynn, and H. Shacham. In *ASIACRYPT*, pages 514–32, 2001.

[7] Ronald Cramer. PhD thesis.

[8] G. Di Crescenzo, R. Lipton, and S. Walfish. Perfectly secure password protocols in the bounded retrieval model. In *TCC*, pages 225–244, 2006.

[9] Y. Dodis, S. Vadhan, and D. Wichs. Proofs of retrievability via hardness amplification. In *TCC*, pages 109–127, 2009.

[10] A. Duc, S. Dziembowski, and S. Faust. Unifying leakage models: From probing attacks to noisy leakage. In *EUROCRYPT*, pages 423–440, 2014.

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

[12] S. Dziembowski and K. Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302, 2008.

[13] A. Faonio, J.B. Nielsen, and D. Venturi. Mind your coins: Fully leakage-resilient signatures with graceful degradation. In *ICALP, Part I*, pages 456–468, 2015.

[14] S. Faust, T. Rabin, L. Reyzin, E. Tromer, and V. Vaikuntanathan. Protecting circuits from leakage: the computationally-bounded and noisy cases. In *EUROCRYPT*, pages 135–156, 2010.

[15] U. Feige, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.

[16] J.A. Halderman, S.D. Schoen, N. Heninger, W. Clarkson, W. Paul, J.A. Calandrino, A.J. Feldman, J. Appelbaum, and E.W. Felten. Lest we remember: cold boot attacks on encryption keys. In *USENIX*, pages 45–60, 2008.

[17] A. Juels and B. Kaliski. PORs: Proofs of retrievability for large files. In *CCS*, 2007.

[18] J. Katz and V. Vaikuntanathan. Signature schemes with bounded leakage resilience. In *ASIACRYPT*, pages 703–720, 2009.

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

[20] P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *CRYPTO*, pages 388–397, 1999.

[21] Y. Lindell. Parallel coin-tossing and constant-round secure two-party computation. In *CRYPTO*, pages 171–189, 2001.

[22] S. Micali and L. Reyzin. Physically observable cryptography (extended abstract). In *TCC*, pages 278–296, 2004.

[23] E. Miles and E. Viola. Shielding circuits with groups. In *STOC*, pages 251–260, 2013.

[24] M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages 18–35, 2009.

[25] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *CRYPTO*, pages 31–53, 1992.

[26] J.J. Quisquater and D. Samyde. Electromagnetic analysis (ema): Measures and countermeasures for smart cards. In *E-Smart*, pages 200–210, 2001.

[27] H. Shacham and B. Waters. Compact proofs of retrievability. In *ASIACRYPT*, 2008.

[28] C. Wang, Q. Wang, K. Ren, and W. Lou. Privacy-preserving public auditing for data storage security in cloud computing. In *INFOCOM*, pages 525–533, 2010.