Survey:
Leakage Resilience and the Bounded Retrieval Model

Joël Alwen, Yevgeniy Dodis, and Daniel Wichs

Department of Computer Science, New York University.
{jalwen,dodis,wichs}@cs.nyu.edu

Abstract. This survey paper studies recent advances in the field of *Leakage-Resilient Cryptography*. This booming area is concerned with the design of cryptographic primitives resistant to arbitrary side-channel attacks, where an attacker can repeatedly and adaptively learn information about the secret key, subject *only* to the constraint that the *overall amount* of such information is bounded by some parameter ℓ. We start by surveying recent resultw in the so called *Relative Leakage Model*, where all the parameters of the system are allowed to depend on ℓ, and the goal is to make ℓ large relative to the length of the secret key. We conclude by showing how to extend the relative leakage results to the *Bounded Retrieval Model* (aka "Absolute Leakage Model"), where only the secret key length is allowed to be slightly larger than ℓ, but all other system parameters (e.g., public-key, communication, etc.) are independent of the absolute value of ℓ. Throughout the presentation we will emphasize the information-theoretic techniques used in leakage-resilient cryptography.

1 Introduction

Traditionally, cryptographic systems rely on complete privacy of cryptographic keys. Unfortunately, in real systems, this idealized assumption is hard to meet perfectly. In many situations, the attacker might get some partial information about the secret keys through means which were not anticipated by the designer of the system and, correspondingly, not taken into account when arguing its security. Such attacks, typically referred to as *side-channel attacks*, come in a large variety (radiation, power, temperature, running time, fault detection, etc.), and often lead to a complete break of an otherwise "secure" system (e.g. [Koc96,BDL97,BS97,KJJ99,QS01,GMO01]). The situation becomes even worse if one also takes into account various computer viruses, internet worms and other malware, which might persist in a system inconspicuously for some time and leak private information to a remote attacker, until it is eventually detected.

Given that one cannot hope to eliminate the problem of side-channel and malware attacks altogether, it is natural to design cryptographic schemes which remain (provably) secure, even in the face of such attacks. To do so, we must first decide on an appropriate model of what information the adversary can learn during a side-channel attack. In this work, we assume that the attacker can repeatedly and adaptively learn *arbitrary functions* of the secret key sk, as long as the total number of bits leaked is bounded by some parameter ℓ. Due to its generality, this model seems to include essentially all known side-channel attacks, and has recently attracted a lot of attention from the research community. In particular, this model simultaneously covers the following two typical scenarios, which seem to be treated differently in the existing literature.

RELATIVE LEAKAGE. Here, for a secret key of some particular length s, we assume that the leakage ℓ is bounded by some shrinking function of s; e.g., the attacker's leakage is less than half of the key-size. This assumption seems to be natural for modeling attacks where, no matter what the key-size is, the attacker gets some imperfect reading of the key. For example, this naturally models "memory" attacks [HSH+08] (where the attacker might get part of the key stored in RAM), "microwave" attacks (where the attacker manages to extract a corrupted copy of the key from a smart-card), or various power attacks (which repeatedly leak almost the same information about the secret, such as its hamming weight), among others.

ABSOLUTE LEAKAGE. Here we assume that there is a natural bound ℓ on the overall amount of information the attacker can learn throughout the lifetime of the system, particularly concentrating on the setting when ℓ can be extremely large. A prime example of this comes from most malware attacks, where a persistent virus may transmit a large amount of private data to a remote attacker. Nevertheless, in many situations it is either impossible, too time-consuming, or simply not cost-effective for the virus to download "too much data" (e.g. many gigabytes). In such situation one might resist side-channel attacks, but only by making the secret key *intentionally large*, to dominate the retrieval bound ℓ. This *by itself* might not be a big problem for usability, given the extremely cheap price of storage nowadays. Therefore, the main goal of this setting, usually refereed to as the *Bounded Retrieval Model* (BRM) [CLW06,Dzi06], is to ensure that the *necessary* inefficiency in storage is essentially the *only* inefficiency that the users of the system incur. In particular, honest users should only have to read a small portion of the secret (this is called *locality*), and their computation and communication should not be much larger than in conventional cryptosystems.

To summarize, both leakage models – relative and absolute – study essentially the same technical question. However, the BRM setting additionally demands that: *users can increase their secret key size flexibly, so as to allow for an arbitrary large absolute leakage ℓ, but without degrading other efficiency parameters, such as computation, communication and locality.* This is the perspective we will take in this paper, treating both settings together, while striving to allow for the above flexibility. Indeed, we will see that a natural paradigm for designing efficient BRM scheme often starts with designing a relative leakage scheme first, and then extending the basic scheme to the BRM model.

Another interesting feature of leakage-resilient cryptography is that information-theoretic techniques are often used even in the design of computationally secure schemes, such as password authentication, public-key encryption or digital signature schemes. We will try to emphasize these techniques throughout the presentation.

1.1 Related Work

WEAK SECRETS, SIDE-CHANNEL ATTACKS AND BRM. The model of side-channel attacks, as studied in this work, is very related to the study of cryptography with *weak* *secrets*. A weak secret is one which comes from some arbitrary distribution that has a sufficient level of (min-)entropy, and one can think of a secret key that has been partially compromised by side-channel attacks as coming from such a distribution. Most of the prior work concerning weak secrets is specific to the *symmetric key setting* and much of this work is *information-theoretic in nature*. For example, the study of privacy-amplification [BBR88,Mau92b,BBCM95] shows how two users who *share* a weak secret can agree on a uniformly random key in the presence of a passive attacker. The works of [MW97,RW03,DKRS06,KR09,DW09] extend this to active attacks, and the works of [Mau92a,AR99,ADR02,Lu02,Vad04] extended this to the case of *huge* secrets (motivated by the Bounded Storage Model, but also applicable to the BRM). Such information-theoretically secure schemes can only be used *once* to convert a shared secret, which may have been partially compromised by side-channel attacks, into a *single* uniform session-key.

In the computational setting, users can agree on *arbitrarily many* session-keys using Password Authenticated Key Agreement (PAKE) [BM93,BPR00,BMP00,KOY01,GL06], where they use their shared weak (or partially compromised) secret key as the password. However, these solutions do not scale to the BRM, as they do not preserve low locality when the secret is large. The Bounded Retrieval Model (BRM), where users have a huge secret key which is subject to large amounts of adversarial leakage, was introduced by [CLW06,Dzi06]. In particular, Dziembowski [Dzi06] constructed a *symmetric key* authenticated key agreement protocol for this setting in the Random Oracle model. This was later extended to the standard model by [CDD$^+$07]. Other symmetric-key applications, such as password authentication and secret sharing, were studied in the BRM setting by [CLW06] and [DP07], respectively. We also note that *non-interactive* symmetric key encryption schemes using partially compromised keys were constructed implicitly in [Pie09] (based on weak pseudorandom functions) and explicitly in [DKL09] (based on "learning parity with noise").

The study of side-channel attacks in the *public-key* setting was initiated by Akavia et al. [AGV09], who showed that Regev's public-key encryption scheme [Reg05] (based on lattices) is secure against the side-channel attacks in the relative leakage model. Subsequently, Naor and Segev [NS09] presented several new constructions of public-key encryption schemes for this setting, based on other (non-lattice) assumptions, tolerating more leakage and achieving CCA2 security. Very recently, Alwen et al. [ADN$^+$09] showed how to build the first public-key encryption in the BRM based on a variety of assumptions (lattices, quadratic residuosity, bilinear maps). Along the way, they also build identity-based encryption (IBE) schemes in the relative leakage model. The main drawback of these works is that (non-interactive) encryption schemes *inherently* only allow the adversary to perform side-channel attacks *prior to* seeing a ciphertext. This concern was addressed by Alwen et al. [ADW09] who showed how to construct public-key (interactive) key-exchange protocols both in the relative leakage-model and in the BRM, where the leakage was allowed to occur both before and after running the protocol. Along the way, the work of [ADW09] built leakage-resilient identification schemes (again, both in the relative leakage model and the BRM), used them to construct leakage-resilient signature schemes (in the random oracle model), and also developed general tools for converting schemes in the relative-leakage models into the more general BRM setting. Finally, Katz and Vaikuntanathan [KV09] recently developed leakage-resilient signature scheme in the standard model.

This survey article could be viewed as the digest of the main ideas and constructions from [ADW09,NS09,ADN+09,KV09], with the emphasis of trying to unify the different-looking techniques used in these works.

OTHER MODELS OF ADVERSARIAL KEY COMPROMISE. It is worth describing several related models for key compromise. One possibility is to restrict the *type* of information that the adversary can learn about the secret key. For example a line of work called *exposure resilient cryptography* [CDH+00,DSS01] studies a restricted class of adversarial leakage functions, where the adversary gets a *subset of the bits* of the secret key. In this setting, one can secure keys against leakage generically, by encoding them using an *all-or-nothing transform (AONT)*. We note that some natural side-channel attacks (e.g. learning the hamming weight of the key) and malware attacks are not captured by this model.

Another line of work, initiated by Micali and Reyzin [MR04] and studied further by [DP08,Pie09,FKPR09], designs various symmetric-key primitives and digital signatures under the axiom that "only computation leaks information". These models are incomparable to our setting, as they restrict the *type* of information the attacker can obtain, but can allow a greater overall *amount* of such information to be leaked. While quite reasonable in some application scenarios, such as power/radiation attacks, the above axiom does not seem to apply to many other natural attacks, such as the memory/microwave attacks or virtually all malware/virus attacks. A related model, where the adversary can learn/influence the values on some subset of wires during the evaluation of a circuit, was studied by Ishai et al. [ISW03,IPSW06], and recently generalized by [FRT09].

Lastly, the recent works [DKL09,DGK+09] study *auxiliary input*, where the adversary can learn functions $f(\mathsf{sk})$ of the secret key sk subject only to the constraint that such a function is *hard to invert*. Technically, this is a strictly stronger model than the one considered in this work as such functions f can have output length larger than the size of the secret key.

2 Preliminaries

ENTROPY. The *min-entropy* of a random variable W is $\mathbf{H}_\infty(W) \stackrel{\text{def}}{=} -\log(\max_w \Pr[W = w])$. This is a standard notion of entropy used in cryptography, since it measures the worst-case predictability of W. We also review a generalization from [DORS08], called *average conditional min-entropy* defined by

$$\widetilde{\mathbf{H}}_\infty(W|Z) \stackrel{\text{def}}{=} -\log\left(\mathbb{E}_{z\leftarrow Z}\left[\ \max_w \Pr[W = w|Z = z]\ \right]\right) = -\log\left(\mathbb{E}_{z\leftarrow Z}\left[2^{-\mathbf{H}_\infty(W|Z=z)}\right]\right).$$

This measures the worst-case predictability of W by an adversary that may observe a correlated variable Z. We will use the following lemmas to reason about entropy.

Lemma 1 ([DORS08]). *Let W, X, Z be random variables where Z takes on values in a set of size at most 2^ℓ. Then $\widetilde{\mathbf{H}}_\infty(W|(X,Z)) \geq \widetilde{\mathbf{H}}_\infty((W,X)|Z) - \ell \geq \widetilde{\mathbf{H}}_\infty(W|X) - \ell$ and, in particular, $\widetilde{\mathbf{H}}_\infty(W|Z) \geq \mathbf{H}_\infty(W) - \ell$.*

In [ADW09], the authors define a more general notion of conditional min-entropy $\widetilde{\mathbf{H}}_\infty(W \mid \mathcal{E})$, where $\mathcal{E}$ can denote any arbitrary experiment (and not just some "one-time" random variable Z). Intuitively, this measures the (log of the) best prediction probability for W after running the experiment $\mathcal{E}$. We refer to [ADW09] for the details.

REVIEW OF Σ-PROTOCOLS. Let $\mathcal{R}$ be a relation consisting of *instance, witness* pairs $(x, w) \in \mathcal{R}$ and let $L_R = \{x \mid \exists w, (x, w) \in \mathcal{R}\}$ be the *language* of $\mathcal{R}$. A Σ-protocol for $\mathcal{R}$ is a protocol between a PPT ITM prover $\mathcal{P}(x, w)$ and a PPT ITM verifier $\mathcal{V}(x)$, which proceeds in three rounds where: (1) the prover $\mathcal{P}(x, w)$ sends an initial message a, (2) the verifier $\mathcal{V}(x)$ sends a uniformly random challenge c, (3) the prover $\mathcal{P}(x, w)$ sends a response z. The verifier $\mathcal{V}(x)$ either *accepts* or *rejects* the conversation by computing some predicate of the instance x and the conversation (a, c, z). We require that Σ-protocols satisfy the following three properties:

1. *Perfect Completeness:* For any $(x, w) \in \mathcal{R}$, the execution $\{\mathcal{P}(x, w) \rightleftharpoons \mathcal{V}(x)\}$ is always accepting.
2. *Special Soundness:* There is an efficient algorithm such that, given an instance x and two accepting conversations for x: (a, c, z), (a, c', z') where $c \neq c'$, the algorithm outputs w such that $(x, w) \in \mathcal{R}$.
3. *Perfect Honest Verifier Zero Knowledge (*HVZK*):* There is a PPT simulator $\mathcal{S}$ such that, for any $(x, w) \in \mathcal{R}$, the simulator $\mathcal{S}(x)$ produces conversations (a, c, z) which are *identically distributed* to the conversations produced by an honest execution $\{\mathcal{P}(x, w) \rightleftharpoons \mathcal{V}(x)\}$.

As was shown in [CDS94], the HVZK property implies *witness indistinguishability*. Here, we rephrase essentially the same property in a slightly different manner. We show that, oracle access to a prover $\mathcal{P}(x, w)$ does not decrease the entropy of w *in any experiment* in which x is given to the predictor.

Lemma 2. *Let $(\mathcal{P}, \mathcal{V})$ be an* HVZK *protocol for the relation $\mathcal{R}$, and let (X, W) be random variables over $\mathcal{R}$. Let $\mathcal{E}_1$ be an arbitrary experiment in which $\mathcal{A}$ is given X at the start of the experiment, and let $\mathcal{E}_2$ be the same as $\mathcal{E}_1$, except that $\mathcal{A}$ is also given oracle access to $\mathcal{P}(X, W)$ throughout the experiment. Then $\widetilde{\mathbf{H}}_\infty(W|\mathcal{E}_2) = \widetilde{\mathbf{H}}_\infty(W|\mathcal{E}_1)$.*

ONE-WAY FUNCTIONS (OWF) AND SECOND-PREIMAGE RESISTANCE (SPR). We review these two standard notions. In the full generality, the index i for the OW/SPR function f_i is sampled by a special index generation procedure $\mathsf{Gen}(1^\lambda)$ (where λ is the security parameter), which also defines the domain D_i and the range R_i for the function.

Definition 1 (One Way Functions (OWF)). *A family of functions $\mathcal{F} = \{f_i : D_i \to R_i\}$ is* one-way *if:*

- *Easy to generate, sample and compute: There exist efficient algorithms for key generation $i \leftarrow \mathsf{Gen}(1^\lambda)$, sampling $w \leftarrow D_i$ and for computing $f_i(w)$ in time* $\mathsf{poly}(\lambda)$.
- *Hard to invert: For any PPT algorithm $\mathcal{A}$, we have* $\Pr[f_i(\mathcal{A}(i, f_i(w))) = f_i(w)] \leq \mathrm{negl}(\lambda)$*, where the probability is over random $i \leftarrow \mathsf{Gen}(1^\lambda)$, $w \leftarrow D_i$ and the random coins of $\mathcal{A}$.*

Definition 2 (Second Pre-Image Resistant Functions (SPR)). *A family of functions* $\mathcal{F} = \{f_i : D_i \to R_i\}$ *is* second-preimage resistant *(SPR) if* $\mathcal{F}$ *is easy to generate, sample and compute (defined the same way as for OWF) and, for any PPT algorithm* $\mathcal{A}$, $\Pr[w' \neq w \wedge f_i(w') = f_i(w) \mid w' = \mathcal{A}(i, f_i(w), w)] \leq \mathrm{negl}(\lambda)$*, where the probability is over random* $i \leftarrow \mathsf{Gen}(1^\lambda)$, $w \leftarrow D_i$ *and the random coins of* $\mathcal{A}$*. We define the* loss *of* f_i *to be* $\mathcal{L}(f_i) \stackrel{def}{=} (\log(|D_i|) - \log(|R_i|))$.

In theory, it is known [Rom90] that for any polynomial $p(\lambda)$, the existence of OWFs implies the existence of SPR functions with $D_i = \{0,1\}^{p(\lambda)}$, $R_i = \{0,1\}^\lambda$. In practice, it is easy to construct SPR functions from most natural number-theoretic assumptions. For example, if the discrete log problem is hard in some group G of prime order q, the following is a simple SPR function from $\mathbb{Z}_q^n \to G$: $(w_1 \ldots w_n) \mapsto \prod_{j=1}^n g_j^{w_j}$, where $g_1 \ldots g_n$ are random generators of G (forming part of the function index i).

As we shall see, SPR functions will play a critical role in the design of leakage-resilient schemes, but first we need to model leakage-resilience.

LEAKAGE ORACLE. We model adversarial side-channel attacks on a secret key sk, by giving the adversary access to a *side-channel oracle*, which the adversary can (periodically) query to gain information about sk. Intuitively, we would like to capture the fact that the adversary can compute arbitrary efficient functions of the secret key as long as the *total* number of bits learned is *bounded* by some parameter ℓ. In general, these *leakage functions* can be chosen adaptively, based on the results of prior leakage attacks and any other events that may take place during the attack game. The following definition formalizes the above concept.

Definition 3. *A* leakage oracle $\mathcal{O}_{\mathsf{sk}}^{\lambda,\ell}(\cdot)$ *is parameterized by a secret key* sk*, a* leakage parameter ℓ *and a security parameter* λ*. A query to the oracle consists of (a description of) a* leakage function $h : \{0,1\}^* \to \{0,1\}$*. The oracle computes the function* $h(\mathsf{sk})$ *for at most* $\mathsf{poly}(\lambda)$ *steps and, if the computation completes, responds with the output, and otherwise, outputs* 0*. A leakage oracle* $\mathcal{O}_{\mathsf{sk}}^{\lambda,\ell}(\cdot)$ *responds to at most* ℓ *queries, and ignores all queries afterwards.*

3 Relative Leakage Model

We start with the relative leakage model, where the goal is to design a cryptographic scheme allowing one to tolerate relative leakage ℓ as close to the length of the secret key of the system as possible.

3.1 Password Authentication and OWF

Pasword authentication is, perhaps, the most basic cryptographic problem. A client Alice has a secret key sk and wishes to authenticate herself to a server Bob, who stores some function pk of Alice's key. It is assumed the the communication channel between Alice and Bob is secure, but server Bob's storage pk is not. Thus, it must be the case that no valid sk can be computed from pk. Therefore, it is clear that a necessary and sufficient primitive for the problem of password-authentication is a OWF. Namely, the key generation algorithm KeyGen sets $\mathsf{sk} = w$ and $\mathsf{pk} = (i, f_i(w))$, where i is the index of a OWF from D_i to R_i. In the setting of leakage, the adversary $\mathcal{A}$ is also given oracle access to $\mathcal{O}_{\mathsf{sk}}^{\lambda,\ell}(\cdot)$. Notice, in this setting adaptive access to the leakage oracle is equivalent to choosing a single leakage function $h(\mathsf{sk})$ whose output is ℓ bits. We call the resulting OWF family $\mathcal{F}$ *ℓ-leakage-resilient* (ℓ-LR).

The first hope of building LR-OWFs is to hope that all OWF's are LR. The good news is that it is true for $\ell(\lambda) = O(\log \lambda)$, since one can always guess the proper leakage with probability $\frac{1}{2^\ell} \geq \frac{1}{poly(\lambda)}$. The bad news is that it is unlikely we can say more about it. As an example, consider $f(x_1, x_2) = f'(x_1)$ where $|x_1| = \lambda^{0.01}$, $|x_2| = (\lambda - \lambda^{0.01})$ and f' is some auxiliary OWF. Clearly, f is not even $(\lambda^{0.01})$-LR. The next hope is to try some natural OWF's and hope that they happen to be leakage-resilient. Unfortunately, this is also problematic. For example, consider the modular exponentiation function $f(w) = g^w$ over some group G of order q. It turns out that we do not have any attacks on this f, and, yet, we cannot prove the leakage-resilience of this function based on the discrete log assumption either. The difficulty is in simulating the leakage oracle: given only $f(w) = g^w$, there does not appear to be any way to compute (with any decent probability) $h(w)$ for an adversarially chosen function $h : \mathbb{Z}_q \to \{0,1\}^\ell$, when $\ell = \omega(\log \lambda)$.

This is where the SPR functions come to the rescue. In the SPR attack on a function f, the SPR attacker $\mathcal{A}$ is given a valid pre-image w of $x = f(w)$. Thus, it is easy to simulate the correct value $z = h(w)$ for the leakage attacker $\mathcal{B}$. However, if both z and x are much shorter than w, the leakage attacker $\mathcal{B}$ still has a lot of uncertainty about the original value w used by $\mathcal{A}$. Hence, there is a good chance that $\mathcal{B}$ will compute a different pre-image $w' \neq w$ of x, therefore violating the SPR security of f. This easy observation is formalized below, but will form the basis for building more complicated leakage-resilient primitives.

Theorem 1. *If $\mathcal{F}$ is an SPR family with loss $\ell = \ell(\lambda)$ (see Definition 2), then $\mathcal{F}$ is $(\ell - \omega(\log \lambda))$-LR-OWF.*

Proof. Assume that f_i is not a ℓ'-LR-OWF, where $\ell' = (\ell - \omega(\log \lambda))$. So there exists an inverter $\mathcal{B}$ which inverts $f_i(w)$ (given $f_i(w)$ and leakage $h(w)$) with probability ε which is non-negligible. We construct an algorithm $\mathcal{A}$ which breaks the SPR security with non-negligible advantage (analyzed below).

On input $(i, w, x = f_i(w))$, $\mathcal{A}$ invokes $\mathcal{B}(i, x)$. When $\mathcal{B}$ makes a leakage query h, $\mathcal{A}$ responds with $h(w) \in \{0,1\}^{\ell'}$. If $\mathcal{B}$ then returns a valid pre-image w' such that $f_i(w') = x$, $\mathcal{A}$ returns w' iff $w' \neq w$. It is clear that $\mathcal{A}$ simulated $\mathcal{B}$ perfectly. Hence,

$$\Pr(\mathcal{A} \text{ succeeds}) \geq \Pr(\mathcal{B} \text{ succeeds } \wedge\ w \neq w') \geq \varepsilon - \Pr(w = w')$$

Let W be the random variable corresponding to sampling w from D_i, and denote by $X = f_i(W)$, $Z = h(W)$. It is clear that even if $\mathcal{B}$ is infinitely powerful, its best chance to predict W from X and Z is $2^{-\widetilde{\mathbf{H}}_\infty(W|X,Z)}$. However, using Lemma 1, we know that $\widetilde{\mathbf{H}}_\infty(W \mid X, Z) \geq \widetilde{\mathbf{H}}_\infty(W) - (\log |R_i| + \ell') = \log(|D_i|/|R_i|) - \ell' = \ell - \ell'$, which gives $\Pr(w = w') \leq 2^{\ell' - \ell}$. Setting $\ell' = (\ell - \omega(\log \lambda))$, we get that $\mathcal{A}$ succeeds with non-negligible probability $(\varepsilon - \mathrm{negl}(\lambda))$. $\square$

As an example, recall the SPR function $f(w_1, \ldots, w_n) = \prod_{j=1}^{n} g_j^{w_j}$ defined over some group G of prime order q. We conclude that if the discrete logarithms in G are hard, then f is ℓ-LR-OWF for $\ell = (n \log q - \log |G| - \omega(\log \lambda))$. For large n, this value of ℓ approaches the length $(n \log q)$ of the secret key $w = (w_1 \ldots w_n)$.

3.2 Identification Schemes

Recall, (public-key) identification (ID) schemes are similar to password authentication schemes, except the communication between the client Alice and the server Bob is no longer assumed secure. As a result, ID schemes must be interactive. We informally recall two main notions of security for ID schemes: *passive* security and *active* security. Both notions proceed in two stages. In the *learning stage*, the attacker $\mathcal{A}(\mathsf{pk})$ gets access to the communication channel between Alice and the verifier. In the passive attack, this is modeled by giving $\mathcal{A}$ oracle access to the transcript oracle $\mathcal{T}$, which returns an honestly generated communication transcript between Alice and Bob. In the active attack, $\mathcal{A}$ is actually allowed to play the role of the verifier with Alice (and possibly deviate from the honest verifier behavior). Formally, $\mathcal{A}$ is given oracle access to polynomially many "copies of Alice". After the end of the learning stage, $\mathcal{A}$ enters the *impersonation stage* and loses its "learning oracle" (either $\mathcal{T}$ or Alice herself). In this stage $\mathcal{A}$ tries to impersonate Alice to the honest verifier Bob, and wins the game if it succeeds.

LEAKAGE-RESILIENT ID SCHEMES. In the setting of leakage, the adversary $\mathcal{A}$ is also given oracle access to the leakage oracle $\mathcal{O}_{\mathsf{sk}}^{\lambda,\ell}(\cdot)$. Not very surprisingly, it is easier to handle leakage calls made during the learning stage than the leakage calls made during the impersonation stage (which might depend on the actual challenges received). For this reason, we will call the ID scheme (ℓ_1, ℓ_2)-*leakage-resilient* (LR) if the attacker can learn up to ℓ_1 bits in the learning stage, and up to ℓ_2 bits in the impersonation stage. For simplicity of exposition, from now now we assume that the attacker calls the leakage oracle precisely once in each stage, learning ℓ_1 and ℓ_2 bits respectively.

CONSTRUCTIONS. Recall, in the leak-free setting, a Σ-protocol for proving the knowledge of a pre-image of any OWF immediately gives a passively secure ID scheme. Namely, setting $\mathsf{sk} = w$, $\mathsf{pk} = (i, x = f_i(w))$, let $\mathcal{R} = \{(x = f(w), w)\}$ and Π be a Σ-protocol for $\mathcal{R}$ with challenge size $|c| = k = \omega(\log \lambda)$. Then Π is a passively secure ID scheme. Intuitively, the HVZK property of Π enables us to perfectly simulate the transcript queries in the learning stage. On the other hand, if an attacker $\mathcal{A}$ can respond to a random challenge c with probability ε in the impersonation stage, then by rewinding the attacker with a new (random) challenge c', one can obtain two accepting conversations $(a, c, z), (a, c', z')$ with $c \neq c'$ with probability $\varepsilon(\varepsilon - \frac{1}{2^k})$,[1] which is non-negligible if ε is non-negligible and $k = \omega(\log \lambda)$. Then, the special soundness of Π implies that we can extract a valid witness w' from the attacker, contradicting the one-wayness of f_i.

It is easy to see that this analysis easily extends to the leakage-resilient setting, provided that: (a) one uses a *leakage-resilient* OWF instead of any OWF; and (b) the leakage threshold ℓ of this OWF is greater than $\ell_1 + 2\ell_2$, since we need to rewind the attacker in the impersonation stage, and hence double the leakage to $2\ell_2$ bits.

[1] We omit this standard derivation.

Theorem 2. *Assume Π is a Σ-protocol for $(\ell_1 + 2\ell_2)$-LR-OWF with challenge size $\omega(\log \lambda)$. Then Π is (ℓ_1, ℓ_2)-LR passively secure ID scheme.*

Using Theorem 1, this means we can use an SPR function with loss $\ell = (\ell_1 + 2\ell_2 + \omega(\log(\lambda))$. It turns out, however, that this will immediately give an actively secure ID scheme! The reason is that, in the SPR reduction, the SPR adversary actually knows the pre-image w, so it can easily simulate the leakage oracle, as well as play the role of the prover in the active learning stage. Moreover, since Σ-protocols are witness indistinguishable, Lemma 2 implies that, information-theoretically, the oracle access to the prover does not reduce the min-entropy of w conditioned on the leakage. Namely, all the information the ID attacker learns about w comes from the leakage queries. Overall, we get the following result:

Theorem 3. *Assume Π is a Σ-protocol with challenge size $\omega(\log \lambda)$ for an SPR function with loss $\ell(\lambda) = (\ell_1 + 2\ell_2 + \omega(\log \lambda))$. Then Π is (ℓ_1, ℓ_2)-LR actively secure ID scheme.*

We notice that, in principle, any SPR function has a Σ-protocol with challenge size $\omega(\log \lambda)$ if OWFs exist [FS89,GMW91]. However, concrete SPR functions often have very efficient protocols. For example, such an efficient Σ-protocol for the SPR function $f(w_1, \ldots, w_n) = \prod_{j=1}^{n} g_j^{w_j}$ is given by Okamoto [Oka92]. This gives a very efficient (ℓ_1, ℓ_2)-LR active ID scheme where $\ell_1 + 2\ell_2$ approaches the length of the secret key w as n grows.

3.3 Signatures

Recall, a signature scheme consists of a key-generation procedure $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, a signing procedure $\sigma \leftarrow \mathsf{Sign}(m, \mathsf{pk})$ which produces a signature σ for the message m, and a verification procedure $\mathsf{Ver}(m, \sigma, \mathsf{sk})$, which uses the secret key sk to assess the (in)validity of the signature σ of m. The standard existential unforgeability (UF) against the chosen message attack (CMA) of the signature scheme states that no efficient attacker $\mathcal{A}(\mathsf{pk})$, given oracle access to the signing procedure $\mathsf{Sign}(\cdot, \mathsf{sk})$, should be unable to forge a valid signature σ of some message m not queried to the signing oracle. In the setting of leakage, the usual UF-CMA security is augmented and the attacker $\mathcal{A}$ is also given oracle access to $\mathcal{O}_{\mathsf{sk}}^{\lambda,\ell}(\cdot)$. The resulting signature scheme is called *ℓ-leakage-resilient (LR).*

t-TIME LEAKAGE-RESILIENT SIGNATURES. In general, the forger $\mathcal{A}$ is allowed to make an arbitrary polynomial number of oracle calls to the signing oracle. For the special case where this number is a-priori bounded by a constant $t \geq 1$, we call the resulting signature scheme a *t-time* signature scheme. In the leak-free setting, such t-time schemes are easier to construct [Lam79] and can be more efficient then general schemes. Further, Naor and Yung [NY89] show how to construct general UF-CMA secure signatures from any such 1-time scheme. Although this transformation does not work in the setting of leakage, [FKPR09] show a similar transformation turns any 3-time ℓ-LR signature into and ℓ-LR signature in the "only computation leaks information" model of [MR04]. Thus, it is still interesting to build *leakage-resilient t*-time signatures for a small constant t. Two such constructions are given by Katz and

Vaikuntanathan [KV09]. One general construction is a variant of Lamport's t-time signatures [Lam79] with $\ell \approx |\mathsf{sk}|/4$, and the other is a much more efficient construction from any sufficiently shrinking "homomorphic collision-resistant hash function" (which can be built from a variety of specific assumptions) with $\ell \approx |sk|/2$. We refer to [KV09] for the details.

LEAKAGE-RESILIENT SIGNATURES VIA FIAT-SHAMIR. Recall, the standard Fiat-Shamir transformation [FS86,AABN02] builds a secure signature scheme from any passively-secure, public-coin, 3-round ID scheme, such as the ID schemes originating from Σ-protocols. To sign the message m, the signer generates the first flow a, sets the challenge $c = H(a, m)$, where H is modeled as a random oracle, and finally computes the third flow z. The signature consists of the tuple (a, z). Not surprisingly, the construction generalizes to the setting of leakage [ADW09,KV09], modulo the following two caveats: (a) the ID scheme must be $(0, \ell)$-LR (i.e., leakage should be allowed in the impersonation stage); and (b) the leakage oracle cannot depend on the random oracle. Luckily, using the construction of passively (in fact, even actively) secure LR ID schemes from SPR functions given in Theorem 3, we satisfy the requirement (a) and can easily eliminate the restriction (b) by direct analysis, obtaining the following result:

Theorem 4. *Assume Π is a Σ-protocol with challenge size $\omega(\log \lambda)$ for an SPR function with loss $\ell(\lambda) = (2\ell + \omega(\log \lambda))$. Then, applying the Fiat-Shamir heuristics to Π, we obtain an ℓ-LR signature scheme in the random oracle model.*

STANDARD MODEL LEAKAGE-RESILIENT SIGNATURE. On an abstract level, the construction in Theorem 4 can be viewed as choosing a secret key $\mathsf{sk} = w$, $\mathsf{pk} = (i, x = f_i(w))$, and letting the signature of m be a "m-dependent, non-interactive, zero-knowledge proof of knowledge (NIZK-POK) of w, in the Random Oracle Model". Katz and Vaikuntanathan [KV09] observed that one can instead use NIZK-POKs in the common-reference string (CRS) model, as opposed to the Random Oracle model. Formalizing this idea, they showed how to obtain a leakage-resilient signature scheme in the standard model. Unfortunately, this is mainly a feasibility result, since existing (so called simulation-sound) NIZK-POKs are extremely inefficient in the CRS model. Constructing practical LR signatures in the standard model remains an important open question.

3.4 Encryption and KEM

We will concentrate on leakage-resilient *public-key* encryption (PKE) schemes, noticing only that leakage-resilient symmetric-key schemes were constructed implicitly in [Pie09] (based on weak pseudorandom functions) and explicitly in [DKL09] (based on "learning parity with noise"). In fact, for our use it will be more convenient to use the notion of a *key-encapsulation mechanism* (KEM) [CS04], which implies PKE (see below). Recall, a KEM consists of a key-generation procedure $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, an encapsulation procedure $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$ which produces ciphertext/randomness pairs (c, k), and a decapsulation procedure $k = \mathsf{Decap}(c, \mathsf{sk})$, which uses the secret key sk to recover the randomness k from a ciphertext c. A KEM allows a sender that knows pk, to securely agree on randomness k with a receiver that possesses sk, by sending an encapsulation-ciphertext c. Once this is done, one can use the randomness k to symmetrically encrypt the message m, giving a trivial way to get PKE from KEM.

The standard *chosen plaintext attack* (CPA) security of a KEM requires that the distribution (pk, c, k), where $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$, is computationally indistinguishable from (pk, k^*, c), where k^* is truly random and independent of c. One can naturally define *ℓ-leakage-resilient (LR) KEMs*, where the attacker $\mathcal{A}(\mathsf{pk})$ gets access to the leakage oracle $\mathcal{O}_{\mathsf{sk}}^{\lambda,\ell}(\cdot)(\mathsf{sk})$ *before* the challenge encapsulation c is produced. Notice, in this setting adaptive access to the leakage oracle is equivalent to choosing a single leakage function $h(\mathsf{sk})$ whose output is ℓ bits.

HASH PROOF SYSTEMS AND LEAKAGE-RESILIENT KEMS. As with the other primitives we studied, not every KEM is leakage-resilient. However, Naor and Segev [NS09] showed that a special class of KEMs, called *hash proof systems* (HPS) [CS02,KPSY09], can be used to easily construct leakage-resilient KEMs.[2] Informally, am HPS is a KEM with the following two properties:

- There exists an *invalid-encapsulation procedure* $c \leftarrow \mathsf{Encap}^*(\mathsf{pk})$, so that ciphertexts generated by $\mathsf{Encap}^*(\mathsf{pk})$ are computationally indistinguishable from those generated by $\mathsf{Encap}(\mathsf{pk})$, *even given the secret key* sk.
- For a fixed pk and *invalid ciphertext* c generated by $\mathsf{Encap}^*(\mathsf{pk})$, the output of $\mathsf{Decap}(c, \mathsf{sk})$ is *statistically* uniform, over the randomness of sk. This property can only hold if a fixed pk leaves statistical entropy in sk.

Notice the difference between valid and invalid ciphertexts. For a fixed pk, a *valid* c, produced by $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$, always decapsulated to the same value k, no matter which secret key sk is used to decapsulate it. On other hand, an invalid c produced by $c \leftarrow \mathsf{Encap}^*(\mathsf{pk})$, decapsulated to a statistically random value based on the randomness of sk.

The above two properties are sufficient to prove leak-free KEM security, showing that for $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$, an attacker given c cannot distinguish k from uniform. The proof by contradiction proceeds as follows. As the first step, we replace the honestly generated $(c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})$ with $c' \leftarrow \mathsf{Encap}^*(\mathsf{pk})$ and $k' \leftarrow \mathsf{Decap}(c', \mathsf{sk})$. Since valid ciphertexts are indistinguishable from invalid ciphertexts even given the secret key sk, the attacker must still distinguish (pk, c', k') from (pk, c', k^*). As the second step, this is argued impossible, since $k' = \mathsf{Decap}(c', \mathsf{sk})$ is *statistically uniform* over the choice of sk, which is unknown to the adversary.

As Naor and Segev noticed in [NS09], this proof also works in the presence of leakage, since the first argument of replacing (c, k) by (c', k') holds even if the adversary saw *all of* sk, and the second argument is *information-theoretic*, so we can argue that ℓ bits of leakage about sk will only reduce the statistical entropy of k' by at most ℓ bits. Thus, as long as decapsulation k' of the invalid ciphertext has $m > \ell$ bits of entropy without leakage, it will still have at least $(m - \ell)$ bits of entropy after the leakage (see Lemma 1). To agree on a uniform value k in the presence of leakage, we just compose

[2] Our informal description and definition of HPS here is a simplified version of the standard one. Although the two are *not* technically equivalent, the standard definition implies ours, which is in-turn sufficient for leakage-resilience and captures the main essence of HPS.

the HPS KEM with a randomness extractor [NZ96], such as a universal hash function. The main benefit of this proof strategy is that, after switching valid/invalid ciphertexts in the first step, we can argue about leakage using a purely information-theoretic analysis.

Since HPS KEMs can be constructed from a variety of assumptions (see [NS09]), we can construct leakage-resilient KEMs and PKEs from many assumptions as well. We also mention that Alwen et al. [ADN$^+$09] recently generalized the notion of HPS to the identity-based setting, which allowed them to construct leakage-resilient identity-based encryption (IBE) schemes in a similar manner (extending and improving the prior LR-IBE construction from [AGV09]).

4 Bounded Retrieval Model

Now that we saw how to build many leakage-resilient primitives in the *relative-leakage model*, we would like to extend the constructions to the bounded retrieval model as well. In the BRM, we want to have the flexibility to allow for arbitrarily large leakage-bounds ℓ, just by increasing the size of the secret, but without any other unnecessary affect on efficiency. The main question that we address in the BRM is one of *leakage-resilience amplification*: assuming we start with some ℓ-leakage-resilient primitive in the relative-leakage model, how can we construct an L-leakage-resilient primitive for arbitrary values of $L \gg \ell$. Ideally, we would like to achieve leakage-resilience amplification with minimal efficiency degradation: even though the "secrets" of the scheme will need to be made potentially huge so that L bits of leakage does not reveal the entire value, we want to make sure that the computational effort and public-key sizes *do not need to grow proportionally*. Following similar discussion in [ADN$^+$09], we consider several approaches, and hone in on the right one. We put most of our discussion into the "toy example" of password authentication. However, this will be the simplest way to showcase the methodology, and the ideas used to construct identification schemes, signatures and public-key encryption in the BRM will be analogous.

4.1 Password Authentication in the BRM

Let us start with the question of building a leakage-resilient "password authentication scheme" (as described in Section 3.1) in the BRM. We now want to build such a scheme where, for any leakage bound L, we have a $\mathsf{KeyGen}()$ procedure that outputs a $(\mathsf{pk}, \mathsf{sk})$ pair where the client's password sk is made potentially *huge* depending on the leakage bound L. As a security guarantee, we would like to ensure that, given pk and L bits of leakage about sk, it is infeasible to come up with any value sk' for which $\mathsf{Verify}(\mathsf{pk}, \mathsf{sk}') = 1$. In addition, the efficiency requirements of the BRM dictate that the size of pk and the computation time of $\mathsf{Verify}(\mathsf{pk}, \mathsf{sk})$ are *independent of L*. We start with the question of leakage-amplification and then address efficiency.

BAD APPROACH: ARTIFICIALLY INFLATING THE SECURITY PARAMETER. As we saw, many of the leakage-resilient primitives in the *relative-leakage model* have leakage-bounds $\ell(\lambda)$ being a large portion of the key-size $s(\lambda)$ which, in turn, depends on a security parameter λ. Therefore, one solution to leakage-amplification is to simply artificially inflate the security parameter λ sufficiently, until $s(\lambda)$ and, correspondingly,

$\ell(\lambda)$ reach the desired level of leakage L we would like to tolerate. Unfortunately, it is clear that this approach gets extremely inefficient very fast – e.g. to allow for Gigabytes worth of leakage, we may need to perform exponentiations on group elements with Gigabyte-long description sizes.

NEW APPROACH: PARALLEL REPETITION. As an improvement over the previous suggestion, we propose an alternative which we call *parallel-repetition*. Assume we have a leakage-resilient scheme in the relative-leakage model, tolerating ℓ-bits of leakage, for some small ℓ. We can create a new "parallel-repetition scheme", by taking n independent copies of the original scheme so that the new secret key $\overline{\mathsf{sk}} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ and the public key $\overline{\mathsf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$ consists of n independently sampled key-pairs of the original scheme. To run verify in the new scheme, the server simply runs $\mathsf{Verify}(\mathsf{pk}_i, \mathsf{sk}_i)$ for each of the component keys individually and accepts if all runs are accepting. One may hope to show that, if the original scheme is ℓ-leakage-resilient than the new construction is L-leakage resilient for $L = n\ell$. Intuitively, if an adversary gets $\leq L = n\ell$ bits of leakage in the new scheme, than there should be many values sk_i for which the adversary learned less than ℓ bits and hence will be unable to come up with any "good value" sk'_i that verifies for the ith position.

Unfortunately, it is far from clear how to prove the above intuition, if we only assume that the underlying scheme is ℓ-leakage resilient. In particular, we would need a reduction showing how to use an adversary that expects L bits of leakage on $\overline{\mathsf{sk}}$ to break the underlying scheme given ℓ bits of leakage on some sk_i. Unfortunately, this seems impossible in general: if the adversary expects to learn the output of some complicated leakage function (for example a hash function) $H(\overline{\mathsf{sk}})$ with L bit output, it is unlikely that we can evaluate this function correctly by learning only some $h(\mathsf{sk}_i)$ with ℓ bit output (even if we know all of sk_j for $j \neq i$).

PARALLEL REPETITION OF SPR FUNCTIONS. To make leakage amplification via parallel repetition work, let us look more specifically at some concrete examples of leakage-resilient password authentication schemes. One such example (Theorem 2) consisted of using ℓ-leakage-resilient OWF where each $\mathsf{pk}_i = f(\mathsf{sk}_i)$ for a uniformly random sk_i. In addition, we showed (Theorem 1) that SPR functions f with loss $\mathcal{L}(f) \geq \ell + \omega(\log(\lambda))$ are ℓ-leakage-resilient OWFs. It is fairly easy to see that n-wise parallel repetition of such a scheme based on an SPR function $f : D \to R$ yields a new SPR function $f' : D^n \to R^n$ with loss $\mathcal{L}(f') = n(\mathcal{L}(f))$. Therefore, we can show directly that parallel-repetition amplifies leakage in this special case, producing an $L = n\ell$-leakage-resilient "passwords authentication scheme".

EFFICIENCY IMPROVEMENT: RANDOM SUBSET SELECTION. To decrease the computational effort of the verification procedure, we have $\mathsf{Verify}^*(\overline{\mathsf{pk}}, \overline{\mathsf{sk}})$ selects some random subset $\{r_1, \ldots, r_t\} \subseteq \{1 \ldots n\}$ of t indices, and only run the original verification procedure $\mathsf{Verify}(\mathsf{pk}_{r_i}, \mathsf{sk}_{r_i})$ for the t selected key-pairs at indices $\{r_1, \ldots, r_t\}$. Here t will be only proportional to the security parameter λ, and can be much smaller than the keys size (which depends on n).

EFFICIENCY IMPROVEMENT: PUBLIC-KEY SIZE REDUCTION. Using parallel-repetition and random-subset selection, we get a "password authentication scheme" which can be made L-leakage-resilient for arbitrarily large L, with the computational effort of verification only proportional to the security parameter λ and not proportional to L. Unfortunately, the public-key size $\overline{\mathsf{pk}}$ is still large and proportional to the leakage-bound L. We can reduce the public-key in the following way:

- The new KeyGen^* procedure of the BRM scheme generates n pairs $(\mathsf{pk}_1, \mathsf{sk}_1), \ldots,$ $(\mathsf{pk}_n, \mathsf{sk}_n)$ of the underlying scheme in the relative-leakage model. It also generates a signing/verification key $(\mathsf{sigk}, \mathsf{verk})$ for a (standard, non-leakage-resilient) signature scheme and computes signatures $\sigma_i = \mathsf{Sign}_{\mathsf{sigk}}(\mathsf{pk}_i)$ for each $i = 1, \ldots, n$. It outputs $\mathsf{pk} = \mathsf{verk}$ and $\mathsf{sk} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_n, \sigma_1, \ldots, \sigma_n)$.
- The new verification procedure $\mathsf{Verify}^*(\mathsf{pk}, \mathsf{sk})$ of the BRM scheme selects t random indices r_i and, for each one verifies that $\mathsf{Verify}(\mathsf{pk}_{r_i}, \mathsf{sk}_{r_i}) = 1$ and also $\mathsf{Ver}_{\mathsf{verk}}(\mathsf{pk}_{r_i}, \sigma_i) = 1$.

The security of this scheme follows from that of the previous paragraph, given the unforgeability of the signature scheme (note that the signing key sigk is never stored by the client or server).

4.2 Identification Schemes and Signatures in the BRM

Recall that our main construction of leakage-resilient ID schemes was based on Σ-protocols for SPR functions. We can essentially use both techniques from the previous section to build leakage-resilient ID schemes in the BRM. This leads to the main construction given in [ADW09]. Essentially, the only difference between the identification scheme and the "password authentication" scheme from the previous section is that, instead of having the client simply "hand over" the secret keys sk_{r_i}, the client runs Σ-protocols for the relation $\{(\mathsf{pk}, \mathsf{sk}) \ : \ \mathsf{pk} = f(\mathsf{sk})\}$. We leverage the fact that the Σ-protocol is Witness Indistinguishable, to argue that observing executions of the Σ-protocol does not reduce the entropy of sk from the point of view of the attacker.

Once we have ID schemes in the BRM, we can just use the Fiat-Shamir transform to get signature schemes in the BRM, as we showed in Section 3.3. We notice that Fiat-Shamir preserves the efficiency properties (public-key size, computational effort, communication complexity) of the ID scheme. However, to maintain short signatures and allow for large leakage, one must relax the standard notion of existential unforgeability to a slightly weaker notion of *entropic unforgeability*. As illustrated by [ADW09], this (necessarily) weaker notion is still sufficient for many applications, such as bulding a signature-based key exchange protocol in the BRM.

In [ADW09], it was shown that for some specific schemes, one can get additional efficiency improvements in the communication complexity (res. signature size) of BRM ID schemes (resp. signatures) by "compacting" the t parallel runs of the Σ-protocol.

4.3 Public-Key Encryption in the BRM

The recent work of [ADN$^+$09] constructs public-key encryption and IBE schemes in the BRM. Again, one of the main components is to show that (a variant) of parallel-repetition can be used to amplify leakage-resilience for PKE schemes constructed out of Hash Proof Systems. Also, a variant of "random-subset selection" can be used to reduce encryption/decryption times and ciphertext sizes to be independent of the leakage bound L. It turns out that the main difficulty, however, is in reducing the public-key size. It is clear that our previous idea of signing the public-keys with a signature scheme and storing the signed values as part of the secret-key, will not work with PKE, where the encryptor needs to encrypt non-interactively, without talking to the decryptor. The difficulty is resolved using the idea of Identity Based Encryption (IBE), where there is a single master-public-key and many secret-keys for various identities. However, we still need the IBE to have the structure of an HPS scheme to prove leakage-resilience of the scheme and leakage-amplification via parallel repetition. Interestingly (variants of) several IBE schemes in the literature have an HPS-like structure. Such schemes can therefore be used to construct Public-Key Encryption schemes in the BRM. We refer to [ADN$^+$09] for the details.

References

[AABN02] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the fiat-shamir transform: Minimizing assumptions for security and forward-security. In *EUROCRYPT '02: Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pages 418–433, London, UK, 2002. Springer-Verlag.

[ADN$^+$09] Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrival model, 2009. Avaialble at `http://eprint.iacr.org/2009/512`.

[ADR02] Yonatan Aumann, Yan Zong Ding, and Michael O. Rabin. Everlasting security in the bounded storage model. *IEEE Transactions on Information Theory*, 48(6):1668–1680, 2002.

[ADW09] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 36–54. Springer, 2009.

[AGV09] A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *Theory of Cryptography — TCC 2009*, volume 5444 of *Lecture Notes in Computer Science*. Springer-Verlag, 2009.

[AR99] Yonatan Aumann and Michael O. Rabin. Information theoretically secure communication in the limited storage space model. In Wiener [Wie99], pages 65–79.

[BBCM95] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Ueli M. Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, 41(6):1915–1923, 1995.

[BBR88] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM J. Comput.*, 17(2):210–229, 1988.

[BDL97] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In *EUROCRYPT*, pages 37–51, 1997.

[BM93] Steven M. Bellovin and Michael Merritt. Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In *ACM Conference on Computer and Communications Security*, pages 244–250, 1993.

[BMP00] Victor Boyko, Philip D. MacKenzie, and Sarvar Patel. Provably secure password-authenticated key exchange using diffie-hellman. In *EUROCRYPT*, pages 156–171, 2000.

[Bon03] Dan Boneh, editor. *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*. Springer, 2003.

[BPR00] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In *EUROCRYPT*, pages 139–155, 2000.

[Bri93] Ernest F. Brickell, editor. *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, volume 740 of *Lecture Notes in Computer Science*. Springer, 1993.

[BS97] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In Jr. [Jr.97], pages 513–525.

[CDD+07] David Cash, Yan Zong Ding, Yevgeniy Dodis, Wenke Lee, Richard J. Lipton, and Shabsi Walfish. Intrusion-resilient key exchange in the bounded retrieval model. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 479–498. Springer, 2007.

[CDH+00] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In *EUROCRYPT*, pages 453–469, 2000.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1994.

[CLW06] Giovanni Di Crescenzo, Richard J. Lipton, and Shabsi Walfish. Perfectly secure password protocols in the bounded retrieval model. In Halevi and Rabin [HR06], pages 225–244.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2002.

[CS04] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.*, 33(1):167–226, 2004.

[DGK+09] Yevgeniy Dodis, Shafi Goldwasser, Yael Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs, 2009.

[DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *STOC*, pages 621–630, 2009.

[DKRS06] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 232–250. Springer, 2006.

[DORS08] 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.

[DP07] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *FOCS*, pages 227–237. IEEE Computer Society, 2007.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302. IEEE Computer Society, 2008.

[DSS01] Yevgeniy Dodis, Amit Sahai, and Adam Smith. On perfect and adaptive security in exposure-resilient cryptography. In *EUROCRYPT*, pages 301–324, 2001.

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *STOC*, 2009. Full version at `http://eprint.iacr.org/2008/503`.

[Dzi06] Stefan Dziembowski. Intrusion-resilience via the bounded-storage model. In Halevi and Rabin [HR06], pages 207–224.

[FKPR09] Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy Rothblum. Leakage-resilient signatures, 2009. Avaialble at `http://eprint.iacr.org/2009/282`.

[FRT09] Sebastian Faust, Leonid Reyzin, and Eran Tromer. Protecting circuits from computationally-bounded leakage. Cryptology ePrint Archive, Report 2009/379, 2009. `http://eprint.iacr.org/`.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[FS89] Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. In *CRYPTO '89: Proceedings on Advances in cryptology*, pages 526–544, New York, NY, USA, 1989. Springer-Verlag New York, Inc.

[GL06] Oded Goldreich and Yehuda Lindell. Session-key generation using human passwords only. *J. Cryptology*, 19(3):241–340, 2006.

[GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

[HR06] Shai Halevi and Tal Rabin, editors. *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*. Springer, 2006.

[HSH$^+$08] 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 Paul C. van Oorschot, editor, *USENIX Security Symposium*, pages 45–60. USENIX Association, 2008.

[IPSW06] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private circuits ii: Keeping secrets in tamperable circuits. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 308–327. Springer, 2006.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Boneh [Bon03], pages 463–481.

[Jr.97] Burton S. Kaliski Jr., editor. *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*. Springer, 1997.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Wiener [Wie99], pages 388–397.

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In Neal Koblitz, editor, *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[KOY01] Jonathan Katz, Rafail Ostrovsky, and Moti Yung. Efficient password-authenticated key exchange using human-memorable passwords. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 475–494. Springer, 2001.

[KPSY09] Eike Kiltz, Krzysztof Pietrzak, Martijn Stam, and Moti Yung. A new randomness extraction paradigm for hybrid encryption. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 590–609. Springer, 2009.

[KR09] Bhavana Kanukurthi and Leonid Reyzin. Key agreement from close secrets over unsecured channels. In *EUROCRYPT*, 2009. Full version at http://eprint.iacr.org/2008/494.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *ASIACRYPT 2009*. Springer, 2009. `http://www.mit.edu/~vinodv/papers/asiacrypt09/KV-Sigs.pdf`.

[Lam79] L. Lamport. Constructing digital signatures from a one-way function. Technical report, SRI International, October 1979.

[Lu02] Chi-Jen Lu. Hyper-encryption against space-bounded adversaries from on-line strong extractors. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 257–271. Springer, 2002.

[Mau92a] Ueli M. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *J. Cryptology*, 5(1):53–66, 1992.

[Mau92b] Ueli M. Maurer. Protocols for secret key agreement by public discussion based on common information. In Brickell [Bri93], pages 461–470.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In Moni Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 278–296. Springer, 2004.

[MW97] Ueli M. Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In Jr. [Jr.97], pages 307–321.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages 18–35, 2009. Also available at `http://eprint.iacr.org/2009/105`.

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pages 33–43. ACM, 1989.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996.

[Oka92] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Brickell [Bri93], pages 31–53.

[Pie09] Krzysztof Pietrzak. A leakage-resilient mode of operation. In *Eurocrypt 2009, Cologne, Germany*, pages 462–482, 2009.

[QS01] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In Isabelle Attali and Thomas P. Jensen, editors, *E-smart*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210. Springer, 2001.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *STOC*, pages 84–93. ACM, 2005.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394. ACM, 1990.

[RW03] Renato Renner and Stefan Wolf. Unconditional authenticity and privacy from an arbitrarily weak secret. In Boneh [Bon03], pages 78–95.

[Vad04] Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *J. Cryptology*, 17(1):43–77, 2004.

[Wie99] Michael J. Wiener, editor. *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.