# Efficient Public-Key Cryptography in the Presence of Key Leakage

Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs

New York University
dodis@cs.nyu.edu, kkh@cs.nyu.edu, lopez@cs.nyu.edu, wichs@cs.nyu.edu

**Abstract.** We study the design of cryptographic primitives resistant to a large class of side-channel attacks, called "memory 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 $\ell$. Although the study of such primitives was initiated only recently by Akavia et al. [2], subsequent work already produced many such "leakage-resilient" primitives [48,4,42], including signature, encryption, identification (ID) and authenticated key agreement (AKA) schemes. Unfortunately, every existing scheme, — for any of the four fundamental primitives above, — fails to satisfy at least one of the following desirable properties:

- **Efficiency.** While the construction may be generic, it should have some *efficient* instantiations, based on standard cryptographic assumptions, and without relying on random oracles.
- **Strong Security**. The construction should satisfy the strongest possible definition of security (even in the presence of leakage). For example, encryption schemes should be secure against chosen *ciphertext* attack (CCA), while signatures should be *existentially* unforgeable.
- **Leakage Flexibility.** It should be possible to set the scheme parameters so that the leakage bound $\ell$ can come arbitrarily close to the secret-key size.

In this work we design the first signature, encryption, ID and AKA schemes which overcome these limitations, and satisfy all the properties above. Moreover, all our constructions are generic, in several cases elegantly simplifying and generalizing the prior constructions (which did not have any efficient instantiations). We also introduce several tools of independent interest, such as the abstraction (and constructions) of *true-simulation extractable* NIZK arguments, and a new *deniable* DH-based AKA protocol based on any CCA-secure encryption.

## 1 Introduction

Traditionally, the security of cryptographic schemes has been analyzed in an idealized setting, where an adversary only sees the specified "input/output behavior" of a scheme, but has no other access to its internal secret state. Unfortunately, in the real world, an adversary may often learn some partial information about secret state via various *key leakage* attacks. Such attacks come in a large variety and include *side-channel attacks* [43,10,7,44,54,27], where the physical realization of a cryptographic primitive

M. Abe (Ed.): ASIACRYPT 2010, LNCS 6477, pp. 613–631, 2010.

can leak additional information, such as the computation-time, power-consumption, radiation/noise/heat emission etc. The cold-boot attack of Halderman et al. [34] is another example of a key-leakage attack, where an adversary can learn (imperfect) information about memory contents of a machine, even after the machine is powered down. Schemes that are proven secure in an idealized setting, without key leakage, may become completely insecure if the adversary learns even a small amount of information about the secret key. Indeed, even very limited leakage attacks have been shown to have devastating consequences for the security of many natural schemes.

Unfortunately, it is unrealistic to assume that we can foresee, let alone block, all of the possible means through which key leakage can occur in real-world implementations of cryptographic schemes. Therefore, the cryptographic community has recently initiated the investigation of increasingly general (formally modeled) classes of leakage attacks, with the aim of constructing *leakage-resilient* cryptographic schemes that remain provably secure even in the presence of such attacks. Of course, if an adversary can get unrestricted information about the secret key, then she can learn the key in its entirety and the security of the system is necessarily compromised. Therefore, we must first place some "upper bound" on the type or amount of information that the adversary can learn. The nature of such bounds varies in the literature, as we survey later. For this work, we only restrict the *amount*, but not the *type*, of information that an adversary can learn through a key-leakage attack. In particular, we will assume that the attacker can learn *any efficiently computable function of the secret key* $sk$, subject only to the constraint that the total amount of information learned (i.e. the output size of the leakage function) is bounded by $\ell$ bits, where $\ell$ is called the "leakage parameter" of the system.[1] Clearly, at this level of generality, the secret-key size $s$ must be strictly greater than the leakage-parameter $\ell$.[2] Therefore, the quantity $\ell/s$ can be thought as the *relative leakage* of the system, with the obvious goal to make it as close to 1 as possible.

Our model of leakage-resilience was recently introduced by Akavia et al. [2], but already attracted a lot of attention from the cryptographic community [48,4,42,3]. In particular, as we survey later, we already know many "leakage-resilient" primitives, including such fundamental primitives as signature schemes, encryption schemes, identification (ID) schemes and authenticated key agreement (AKA) protocols. Unfortunately, we observe that every existing scheme, — for any of the four fundamental primitives above, — fails to satisfy at least one of the following desirable properties:

– **Efficiency.** While the proposed construction may be based on some generic cryptographic primitives, — which is in fact preferable for modular design, — it should have some *efficient* instantiations, based on standard cryptographic assumptions, and without relying on random oracles. We view this property as the main property we will strive to achieve.

[1] More formally, we allow adaptive measurements, as long as the sum of leaked outputs is bounded by $\ell$.

[2] In fact, our actual constructions easily extend to the more general "noisy leakage" model of Naor and Segev [48], where the outputs can be longer than $s$, as long as the "average min-entropy" of $sk$ drops by at most $\ell$ bits. However, we do not pursue this generalization, in order to keep our notation simple.

- **Strong Security**. The construction should satisfy the strongest possible definition of security (even in the presence of leakage). For example, encryption schemes should be secure against chosen *ciphertext* attack (CCA), while signatures should be *existentially* unforgeable, etc.
- **Leakage Flexibility.** It should be possible to set the parameters of the schemes so that the relative leakage $\ell/s$ is arbitrarily close to 1. We call such schemes *leakage-flexible*.

### 1.1 Our Results

In this work we design the first signature, encryption, ID and AKA schemes which simultaneously satisfy the efficiency, strong security and leakage flexibility properties mentioned above. Moreover, all our constructions are generic. This means that the actual construction is modularly defined and explained using natural simpler blocks, and its security against key leakage is also proven no matter how these simpler blocks are (securely) implemented. However, unlike the prior generic constructions, which did not have any known efficient instantiations (at least, with the desired security and flexibility we seek), ours are yet more general, which will allow us to obtain several efficient instantiations. Given this fact, it is not surprising that our contributions can be roughly split into two categories: "conceptual" contributions, allowing us to obtain more general (and, yet, conceptually simpler) leakage-resilient constructions, and "concrete" contributions, allowing us to instantiate our general schemes efficiently.

CONCEPTUAL CONTRIBUTIONS. As we will see, existing schemes (e.g., signature and CCA-encryption) could be largely divided into two categories: potentially efficient schemes, with some *inherent* limitation not allowing them to achieve relative leakage approaching 1 (which also prevents us from using these ideas for our purposes), and more theoretical schemes [48,42], achieving good relative leakage, but relying on the notion of *simulation-sound* non-interactive zero-knowledge (ss-NIZK) [56]. Informally, ss-NIZK proofs remain sound even if the attacker can see simulated proofs of arbitrary (even false) statements. Unfortunately, it appears that the existing cryptographic machinery does not allow us to instantiate non-trivial ss-NIZK proofs efficiently.[3] On the other hand, a recent breakthrough result of Groth-Sahai [33] showed that one can obtain efficient *non-simulation-sound* NIZK proofs for a non-trivial class of languages. While the techniques of [31] could be applied to Groth-Sahai proofs to achiehve ss-NIZKs, it is a non-trivial "exercise" and the resulting proofs are *significantly* less efficient, as the construction involves OR-proofs for Groth-Sahai languages. Therefore, our first idea was to try to generalize the existing constructions sufficiently, making them rely only on regular NIZKs, in the hope that such NIZKs can then be instantiated using the powerful Groth-Sahai techniques.

In the end, this is indeed what we realized. However, in the process we also abstracted away an elegant notion of independent interest: *true-simulation extractable* (tSE) NIZKs. While similar to the notion of simulation-sound extractable NIZKs [31],

[3] The work of [31] constructs ss-NIZK proofs for practical languages and uses them to construct group signatures, but the resulting scheme has signature size of "thousands or perhaps even millions of group elements" [32] despite being constant.

it involves a subtle but rather important difference: whether the adversary has oracle access to simulated proofs for arbitrary (even false) statements or only true ones. Intuitively, both the Naor-Segev's leakage-resilient CCA encryption [48] and Katz-Vaikuntanathan's leakage-resilient signature scheme [42] used the technique of encrypting a witness $x$ for some relation $R$, and then providing a ss-NIZK proof $\varphi$ that the ciphertext $c$ indeed contains the encryption of a valid witness $x$. The main reason for using this technique is to allow the reduction to extract a valid witness from any "new" valid pair $(c^*, \varphi^*)$ produced by the attacker $\mathcal{A}$ (who saw many such valid pairs earlier). In this paper, we will abstract this property into the tSE notion mentioned above (of which the above mentioned technique is a specific example, where the pair $(c, \varphi)$ together makes up a single tSE-NIZK proof). Moreover, we show that true-simulation extractability, as we abstract it, is *precisely* the right notion for generalizing and proving the security of the previous constructions. This has two positive effects. First, it makes the generic constructions of CCA-encryption and signatures somewhat more intuitive, both for proving and understanding. For example, the traditional "double-encryption" paradigm of Naor-Yung [49] for designing CCA-secure schemes from chosen-plaintext secure (CPA-secure) schemes, also used by [48] in the context of key leakage, can be stated as "CPA-encrypting message $m$ under two keys and proving plaintext equality". Using our more general "simulation-extractability view", it is now stated as "CPA-encrypting $m$ and proving that one knows the plaintext". We believe that the latter view is not only more general, but also more intuitive as a way of explaining "CPA-to-CCA" transformation. It also follows the original intuition of Rackoff and Simon [55], who combine CPA-encryption with NIZK-POK to achieve CCA-encryption, but in the model where the sender also has a secret key. A similar discussion is true for our signature constructions.

Second, we show a generic way to build tSE-NIZKs which *avoids using (expensive) ss-NIZKs*. Instead, our method uses *regular* NIZKs and *any* CCA-secure encryption scheme.[4] Perhaps surprisingly, given the current state-of-the-art NIZK and CCA schemes, the combination "CCA + NIZK" appears to be much more efficient in practice than the combination "CPA + ss-NIZK".[5] As a result, we were able to provide a general framework for building leakage-flexible signature and CCA-encryption schemes, eventually allowing us to efficiently instantiate our schemes (by avoiding using ss-NIZKs). We summarize our results for signature and CCA-encryption schemes in Tables 1 and 2, also comparing them to the best prior constructions. In all the tables, the "sub-optimal" entries (for efficiency, security, model or relative leakage of prior constructions) are written in italics, and most prior rows are also explained in the related work Section 1.2. For signatures, we stress that no efficient construction in the standard model was known prior to our work, for any non-trivial relative leakage fraction (let alone 1).

Once we have efficient leakage-flexible signature schemes, we can obtain ID and AKA schemes with the same properties. The signature-based AKA protocol is not deniable. However, we also construct a *deniable* AKA protocol based on our construction

[4] This is OK for the signature application, but might appear strange for our CCA-encryption application, as we need "CCA to get CCA". However, as a building block for tSE-NIZKs, we only need *standard* CCA schemes and as a result obtain *leakage-resilient* CCA schemes.

[5] Indirectly, the same realization was made by Groth [32] and Camenisch et al. [11].

**Table 1.** Previous work on leakage-resilient signatures and results of this work

| Reference | Unforgeability | Model | Leakage | Efficient? |
|---|---|---|---|---|
| [4] | Existential | *Random Oracle* | *1/2* | Yes |
| [4] | *Entropic* | *Random Oracle* | 1 | Yes |
| [42] | Existential | Standard | 1 | *No* |
| This Work | Existential | Standard | 1 | Yes |

**Table 2.** Previous work on leakage-resilient encryption and results of this work

| Reference | Attack | Model | Leakage | Efficient? |
|---|---|---|---|---|
| [2,48] | *CPA* | Standard | 1 | Yes |
| [48] | CCA | Standard | *1/6* | Yes |
| [48] | CCA | Standard | 1 | *No* |
| This Work | CCA | Standard | 1 | Yes |

**Table 3.** Previous work on leakage-resilient identification schemes and results of this work

| Reference | Security | Model | Leakage | Efficient? |
|---|---|---|---|---|
| [4] | *Pre-Impersonation* | Standard | 1 | Yes |
| [4] | Anytime | Standard | *1/2* | Yes |
| [42] (implicit) | Anytime | Standard | 1 | *No* |
| This Work | Anytime | Standard | 1 | Yes |

**Table 4.** Previous work on leakage-resilient AKA and results of this work

| Reference | Model | Leakage | Deniable? | Efficient? |
|---|---|---|---|---|
| [4] | *Random Oracle* | 1 | *No* | Yes |
| [4,42] | Standard | 1 | *No* | *No* |
| This Work | Standard | 1 | *No*/Yes* | Yes |

* Our first AKA protocol is not deniable; our second — is.

of leakage-flexible CCA-secure encryption. We summarize our results for ID schemes in Table 3 and for AKA protocols in Table 4. See Section 6 for details.

CONCRETE CONTRIBUTIONS. As we explained above, we generically reduce the question of building efficient leakage-flexible ID schemes and AKA protocol to the question of efficiently instantiating our leakage-flexible signature and/or encryption schemes. Such instantiations are given in Section 5. We also explained how the latter instantiations became possible in our work, since we gave generic constructions of both primitives based on the new notion of tSE-NIZK, and then showed that satisfying this notion may be possible using *ordinary* NIZKs for appropriate languages, without relying on the expensive simulation-sound NIZKs. Unfortunately, efficient construction of (even ordinary) NIZKs, due to Groth and Sahai [33], are only known for a pretty restrictive class or languages in bilinear groups. Thus, obtaining a *concrete* efficient instantiation still requires quite a substantial effort.

Specifically, all the building blocks have to be instantiated efficiently, and expressed in a form such that the resulting NP relation satisfies the severe limitations imposed by the Groth-Sahai NIZKs. For example, to build leakage-resilient CCA-encryption, we need to have an efficient leakage-flexible CPA scheme, a CCA scheme supporting labels and a one-time signature scheme, all connected together by an efficient NIZK for a complicated "plaintext equality" relation. Similarly, for leakage-resilient signature schemes, we need an efficient second-preimage resistant (SPR; see Definition 1) relation and a CCA scheme supporting labels, once again connected by an efficient NIZK for a complex relation. Not surprisingly, such tasks cannot typically be done by simply combining "off-the-shelf" schemes from the literature. At best, it requires very careful selection of parameters to make everything "match", followed by a round of further efficiency optimizations. Usually, though, it requires the design of new primitives, which work well with other known primitives, to enable efficient NIZK. For example, in this work, we designed two new SPR relations (see Section 5), since prior SPR relations did not appear to mesh well with our CCA encryption scheme. To emphasize the importance of the new SPR relations, we point out that combining previous constructions with Groth-Sahai proofs would require committing to the witness bit-by-bit in order to achieve full extractability.

Overall, we get two different efficient instantiations of both leakage-resilient signature and CCA encryption schemes in the standard model, based on standard (static and "fixed-length") assumptions in bilinear groups, called external Diffie-Hellman (SXDH) and Decision Linear (DLIN). The high-level idea of these schemes, as well as their efficiency, is described in Section 5. The actual low-level details of how to put "everything together" in the most efficient manner, is described in the full version [18].

### 1.2 Related Work

LEAKAGE-RESILIENCE AND MEMORY ATTACKS. Our model of leakage, sometimes called memory-attacks, was first proposed by Akavia et al. [2], who also constructed CPA secure PKE and IBE schemes in this model under the *learning with errors (LWE)* assumption. Later Naor and Segev [48] generalized the main ideas behind these constructions to show that all schemes based on *hash proof systems* (see [15]) are leakage-resilient. In particular, this resulted in efficient constructions based on the DDH and $K$-Linear assumptions, where the relative leakage on the secret key could be made to approach 1. Moreover, [48] showed how to also achieve CCA security in this model by either: (1) relying on the generic (and inefficient) Naor-Yung paradigm where the leakage-rate can be made to approach 1 or (2) using efficient hash proof systems with leakage-rate only approaching $1/6$. Unfortunately, it seems that the hash proof system approach to building CCA encryption is inherently limited to leakage-rates below $1/2$: this is because the secret-key consists of two components (one for verifying that the ciphertext is well-formed and one for decrypting it) and the proofs break down if either of the components is individually leaked in its entirety. The work of [3] generalizes [48] still further by showing how to construct leakage-resilient IBE schemes generically based on *identity-based hash proof systems*, with several instantiations.

Leakage-resilient signature schemes in the model of memory attacks were constructed in the random-oracle model by [4,42], and in the standard model by [42]. The

random-oracle schemes are highly-efficient but suffer from two limitations. Firstly they rely on the Fiat-Shamir [25] transform which is only known to be secure in the Random Oracle model and is not sound in general [30]. Secondly, the schemes can only tolerate leakage which approaches $1/2$ of the secret key. On the other hand, the standard-model schemes allow for relative-leakage approaching 1, but are based on generic simulation-sound NIZKs and do not come with an efficient instantiation.

The work of [4] also constructs ID schemes and AKA protocols. For ID schemes, two notions of security were considered: a weaker notion called pre-impersonation leakage-resilience and a stronger notion called anytime leakage-resilience. Although efficient schemes in the standard model were given for both notions, the leakage resilience could be made to approach 1 only for pre-impersonation leakage while, for anytime leakage, the given schemes can only tolerate a leakage-rate below $1/2$. For AKA schemes, a construction was given based on leakage-resilient signatures (only requiring a weakened notion of security called entropic-unforgeability). Using the appropriate signature schemes, this yielded two types of constructions: efficient constructions in the random-oracle model and generic but inefficient constructions in the standard model (both of which have leakage-rates approaching 1).

OTHER MODELS OF LEAKAGE-RESILIENCE. Several other models of leakage-resilience have appeared in the literature. They differ from the model we described in that they restrict the *type*, as well as *amount*, of information that the adversary can learn. For example, *exposure resilient cryptography* [12,20,41] studies the case where an adversary can only learn some small *subset of the physical bits of the secret key*. Similarly, [38] studies how to implement arbitrary computation in the setting where an adversary can observe a small *subset o the physical wires of a circuit*. Most recently, [24] study a similar problem, where the adversary can observe a low-complexity (e.g. $AC^0$) function of the wires. Unfortunately, these models fail to capture many meaningful side-channel attacks, such as learning the hamming-weight of the bits or their parity.

In their seminal work, Micali and Reyzin [46] initiated the formal modeling of side-channel attacks under the axiom that *"only computation leaks information"* (OCLI), where each invocation of a cryptographic primitive leaks a function of *only* the bits accessed during that invocation. Several primitives have been constructed in this setting including stream ciphers [22,53] and signatures [23]. More recently, [40] construct a general compiler that can secure *all primitives* in this setting assuming the use of some limited leak-free components and the existence of fully homomorphic encryption. On the positive side, the OCLI model only imposes a bound on the amount of information learned during each invocation of a primitive, but not on the overall amount of information that the attacker can get throughout the lifetime of the system. On the negative side, this model fails to capture many leakage-attacks, such as the cold-boot attack of [34], where *all* memory contents leak information, even if they were never accessed.

Lastly, we mention models of leakage-resilience which are strictly stronger than the memory-attacks model. Firstly, the Bounded-Retrieval Model [16,21,4,3] imposes an additional requirement on leakage-resilient schemes, by insisting that they provide a way to "grow" the secret-key (possibly to many Gigabytes) so as to proportionally increase the amount of tolerated leakage, but without increasing the size of the public-key, the computational or communication efficiency of the scheme, or the lengths of the

ciphertexts or signatures. The work of [4] constructs "entropic" signatures, ID schemes and AKA protocols in this setting, while the work of [3] constructs PKE and IBE schemes in this model. A different strengthening is the auxiliary input model [19,17] where the leakage is not necessarily bounded in length, but it is (only) assumed to be computationally hard to recover the secret-key from the leakage. The work of [19] constructs symmetric-key encryption in this model, under a strengthening of the learning parity with noise (LPN) assumption, while [17] constructs public-key encryption under the DDH and LWE assumptions. Yet another strengthening of the memory-attacks model, proposed by [29], is to require that there is a single scheme (parameterized only by the security parameter) which can tolerate essentially any amount of relative-leakage where the exact-security of the scheme degrades smoothly as the relative-leakage increases. In this model, [29] construct a symmetric-key encryption scheme.

## 2 Definitions of Leakage-Resilient Primitives

We model leakage attacks by giving the adversary access to a *leakage oracle*, which he can adaptively access to learn leakage on the secret key. A leakage oracle $\mathcal{O}_{sk}^{\lambda,\ell}(\cdot)$ is parametrized by a secret key $sk$, a leakage parameter $\ell$, and a security parameter $\lambda$. A query to the leakage oracle consists of a function $h_i : \{0,1\}^* \to \{0,1\}^{\alpha_i}$, to which the oracle answers with $y_i = h_i(sk)$. We only require that the functions $h_i$ be efficiently computable, and the total number of bits leaked is $\sum_i \alpha_i \leq \ell$.

**Definition 1 (Leakage Resilient Hard Relation).** *A relation $R$ with a randomized PPT sampling algorithm* $\texttt{KeyGen}$ *is an* $\ell$-leakage resilient hard relation *if:*

- *For any $(sk, pk) \leftarrow \texttt{KeyGen}(1^\lambda)$, we have $(sk, pk) \in R$.*
- *There is a poly-time algorithm that decides if $(sk, pk) \in R$.*
- *For all PPT adversaries $\mathcal{A}^{\mathcal{O}_{sk}^{\lambda,\ell}(\cdot)}$ with access to the leakage oracle $\mathcal{O}_{sk}^{\lambda,\ell}(\cdot)$:*

$$\Pr\left[R(sk^*, pk) = 1 \;\mid\; (pk, sk) \leftarrow \texttt{KeyGen}(1^\lambda)\,,\; sk^* \leftarrow \mathcal{A}^{\mathcal{O}_{sk}^{\lambda,\ell}(\cdot)}(pk)\right] \leq negl(\lambda)$$

*Notice that without loss of generality, we can assume that $\mathcal{A}$ queries $\mathcal{O}_{sk}^{\lambda,\ell}(\cdot)$ only once with a function $h$ whose output is $\ell$ bits.*

**Definition 2 (Leakage Resilient Signatures).** *A signature scheme* $\mathcal{S} = (\texttt{KeyGen}, \texttt{Sign}, \texttt{SigVer})$ *is* $\ell$-leakage resilient *if* $\forall$ *PPT $\mathcal{A}$ we have* $\Pr[\mathcal{A}\ wins] \leq negl(\lambda)$ *in the following game:*

1. **Key Generation:** *The challenger runs $(vk, sk) \leftarrow \texttt{KeyGen}(1^\lambda)$ and gives $vk$ to $\mathcal{A}$.*
2. **Signing and leakage queries:** *$\mathcal{A}^{\mathcal{O}_{sk}^{\lambda,\ell}(\cdot),\mathcal{S}_{sk}(\cdot)}$ is given access to the leakage oracle $\mathcal{O}_{sk}^{\lambda,\ell}(\cdot)$ and the signing oracle $\mathcal{S}_{sk}(\cdot)$. A query to the signing oracle $\mathcal{S}_{sk}(\cdot)$ consists of a message $m$, to which the oracle responds with $\sigma = \texttt{Sign}_{sk}(m)$.*
3. *$\mathcal{A}$ outputs $(m^*, \sigma^*)$ and wins if $\texttt{SigVer}_{vk}(m^*, \sigma^*) = 1$ and $m^*$ was not given to $\mathcal{S}_{sk}(\cdot)$ as a signing query.*

**Definition 3 (Leakage Resilient CCA-Secure Encryption).** *We say that an encryption scheme* $\mathcal{E} = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Dec})$ *is* $\ell$-leakage resilient CCA-secure *if* $\forall$ *PPT $\mathcal{A}$ we have* $\Pr[\mathcal{A}\ wins] \leq \frac{1}{2} + negl(\lambda)$ *in the following game:*

1. **Key Generation:** *The challenger runs* $(pk, sk) \leftarrow \mathtt{KeyGen}(1^\lambda)$ *and gives* $pk$ *to* $\mathcal{A}$.
2. **Decryption and leakage queries:** $\mathcal{A}^{\mathcal{O}_{sk}^{\lambda,\ell}(\cdot),\mathcal{D}_{sk}(\cdot)}$ *is given access to the leakage oracle* $\mathcal{O}_{sk}^{\lambda,\ell}(\cdot)$ *and the decryption oracle* $\mathcal{D}_{sk}(\cdot)$. *A query to the decryption oracle* $\mathcal{D}_{sk}(\cdot)$ *consists of a ciphertext* $c$, *to which the oracle responds with* $m = \mathtt{Dec}_{sk}(c)$.
3. **Challenge generation:** $\mathcal{A}$ *sends plaintexts* $m_0, m_1$ *to the challenger. The challenger chooses* $b \xleftarrow{\$} \{0,1\}$, *and sends* $c^* \leftarrow \mathtt{Enc}_{pk}(m_b)$ *to* $\mathcal{A}$.
4. **Decryption queries:** $\mathcal{A}^{\mathcal{D}_{sk}(\cdot)}$ *is given access to the decryption oracle* $\mathcal{D}_{sk}(\cdot)$ *with the restriction that* $\mathcal{A}$ *cannot send* $c^*$ *as a decryption query. Notice also that* $\mathcal{A}^{\mathcal{D}_{sk}(\cdot)}$ *is* not *given access to the leakage oracle* $\mathcal{O}_{sk}^{\lambda,\ell}(\cdot)$.
5. $\mathcal{A}$ *outputs* $b'$, *and wins if* $b = b'$.

*We refer to a* 0-leakage-resilient CCA-secure *as simply* CCA-secure.

Recall that we can define labeled CCA encryption in which a message is encrypted and decrypted according to a public label $L$. If an encryption scheme $\mathcal{E} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ supports labels, we use the syntax $\mathtt{Enc}^L(m)$ to denote the encryption of message $m$ under label $L$. Similarly, we use $\mathtt{Dec}^L(c)$ to denote the decryption of ciphertext $c$ under the label $L$. In this case, we extend the correctness of encryption/decryption to requiring that $\mathtt{Dec}^L(\mathtt{Enc}^L(m)) = m$. The security definition described in Definition 3 can also be easily modified as follows. A query to the decryption oracle now consists of a ciphertext $c$ and a label $L$, to which the oracle responds with $m = \mathtt{Dec}_{sk}^L(c)$. In the challenge generation stage, $\mathcal{A}$ submits a label $L^*$ as well as messages $m_0, m_1$ and the challenger computes $c^* \leftarrow \mathtt{Enc}_{pk}^{L^*}(m_b)$ for $b \xleftarrow{\$} \{0,1\}$. Finally, in the second stage of decryption queries we require that the adversary is allowed to ask for decryptions of any ciphertext $c$ under label $L$ only subject to $(L, c) \neq (L^*, c^*)$.

**Definition 4 (Leakage Resilient CPA-Secure Encryption).** *We say that an encryption scheme* $\mathcal{E} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ *is* $\ell$-leakage resilient CPA-secure *if* $\forall$ *PPT* $\mathcal{A}$ *we have* $\Pr[\mathcal{A}\ \textit{wins}] \leq \frac{1}{2} + negl(\lambda)$ *in the game described above with the modification that* $\mathcal{A}$ *does not have access to the decryption oracle* $\mathcal{D}_{sk}(\cdot)$. *If an encryption scheme is* 0-leakage-resilient CPA-secure *we simply refer to it as being* CPA secure.

## 3 Simulation Extractability

We start by briefly recalling the notion of *non-interactive zero-knowledge (NIZK)* [8]. For our purposes, it will be slightly more convenient to use the notion of *(same-string) NIZK argument* from [57]. Note, however, that the definitions and constructions given in this section can be extended to the case of NIZK proofs.

Let $R$ be an NP relation on pairs $(x, y)$ with corresponding language $L_R = \{y \mid \exists x \text{ s.t. } (x, y) \in R\}$. A *non-interactive zero-knowledge (NIZK) argument* for a relation $R$ consists of three algorithms $(\mathtt{Setup}, \mathtt{Prove}, \mathtt{Verify})$ with syntax:

- $(\mathrm{CRS}, \mathrm{TK}) \leftarrow \mathtt{Setup}(1^\lambda)$: Creates a common reference string (CRS) and a trapdoor key to the CRS.
- $\pi \leftarrow \mathtt{Prove}_{\mathrm{CRS}}(x, y)$: Creates an argument that $R(x, y) = 1$.
- $0/1 \leftarrow \mathtt{Verify}_{\mathrm{CRS}}(y, \pi)$: Verifies whether or not the argument $\pi$ is correct.

For the sake of clarity, we write $\mathtt{Prove}$, $\mathtt{Verify}$ without the CRS in the subscript when the CRS can be inferred from the context. We require that the following properties hold:

**Completeness:** For any $(x, y) \in R$, if $(\text{CRS}, \text{TK}) \leftarrow \texttt{Setup}(1^\lambda)$ , $\pi \leftarrow \texttt{Prove}(x, y)$, then $\texttt{Verify}(y, \pi) = 1$.

**Soundness:** For any PPT adversary $\mathcal{A}$, $\Pr[\texttt{Verify}(y, \pi^*) = 1 \wedge y \notin L_R] \leq negl(\lambda)$, where the probability is taken over $(\text{CRS}, \text{TK}) \leftarrow \texttt{Setup}(1^\lambda), (y, \pi^*) \leftarrow \mathcal{A}(\text{CRS})$.

**Composable Zero-Knowledge:** There exists PPT simulator $\texttt{Sim}$ such that, for any PPT adversary $\mathcal{A}$ we have $\left|\Pr[\mathcal{A} \text{ wins }] - \frac{1}{2}\right| \leq negl(\lambda)$ in the following game:

- The challenger samples $(\text{CRS}, \text{TK}) \leftarrow \texttt{Setup}(1^\lambda)$ and gives $(\text{CRS}, \text{TK})$ to $\mathcal{A}$.
- $\mathcal{A}$ chooses $(x, y) \in R$ and gives these to the challenger.
- The challenger samples $\pi_0 \leftarrow \texttt{Prove}(x, y), \pi_1 \leftarrow \texttt{Sim}(y, \text{TK}), b \leftarrow \{0, 1\}$ and gives $\pi_b$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs a bit $\tilde{b}$ and wins if $\tilde{b} = b$.

We revisit the notion of simulation extractable NIZK arguments [58,13,50,51,31], and define a new primitive called *true-simulation extractable* NIZK arguments. Apart from satisfying the properties described above, an NIZK argument is simulation extractable if there exists a PPT *extractor* $\texttt{Ext}$ which given an additional trapdoor to the CRS, extracts a witness $x'$ from any proof $\pi$ produced by a malicious prover $P^*$, *even* if $P^*$ has previously seen some *simulated proofs* for other statements. We make an important distinction between our new definition of *true*-simulation extractability, where all simulated proofs seen by $P^*$ are only of *true* statements, and the stronger notion of *any*-simulation extractability, where $P^*$ can also see proofs of *false* statements. As we will see, the former notion is often simpler to construct and sufficient in our applications.

We extend our definition to $f$*-extractability*, where $\texttt{Ext}$ only needs to output some function $f(x')$ of a valid witness $x'$. We further extend this definition to support *labels*, so that the $\texttt{Prove}$, $\texttt{Verify}$, $\texttt{Sim}$, and $\texttt{Ext}$ algorithms also take a public label $L$ as input, and the correctness, soundness, and zero-knowlegde properties are updated accordingly. If $\Pi = (\texttt{Setup}, \texttt{Prove}, \texttt{Verify})$ is an NIZK argument with simulator $\texttt{Sim}$ and extractor $\texttt{Ext}$, we write $\texttt{Prove}^L, \texttt{Verify}^L, \texttt{Sim}^L, \texttt{Ext}^L$ to denote proof, verification, simulation, and extraction under label $L$, respectively.

We start by defining a simulation oracle $\mathcal{SIM}_{\text{TK}}(\cdot)$. A query to the simulation oracle consists of a pair $(x, y)$ and a label $L$. The oracle checks if $(x, y) \in R$. If true, it ignores $x$ and outputs a simulated argument $\texttt{Sim}^L(\text{TK}, y)$, and otherwise outputs $\perp$. We now give a formal definition of true-simulation extractability.

**Definition 5 (True-Simulation $f$-Extractability).** *Let $f$ be a fixed efficiently computable function and let $\Pi = (\texttt{Setup}, \texttt{Prove}, \texttt{Verify})$ be an NIZK argument for a relation $R$, satisfying the completeness, soundness and zero-knowledge properties above. We say that $\Pi$ is* true-simulation $f$-extractable *($f$-tSE) with labels if:*

- *Apart from outputting a CRS and a trapdoor key,* $\texttt{Setup}$ *also outputs an extraction key:* $(\text{CRS}, \text{TK}, \text{EK}) \leftarrow \texttt{Setup}(1^\lambda)$.
- *There exists a PPT algorithm* $\texttt{Ext}(y, \varphi, \text{EK})$ *such that for all* $P^*$, $\Pr[P^* \; wins] \leq negl(\lambda)$ *in the following game:*
  1. **Key Generation:** *The challenger runs* $(\text{CRS}, \text{TK}, \text{EK}) \leftarrow \texttt{Setup}(1^\lambda)$ *and gives* CRS *to* $P^*$.
  2. **Simulation queries:** $P^{*\mathcal{SIM}_{\text{TK}}(\cdot)}$ *is given access to the simulation oracle* $\mathcal{SIM}_{\text{TK}}(\cdot)$*, which it can adaptively access.*

3. **Adversary Output:** *$P^*$ outputs a tuple $(y^*, L^*, \varphi^*)$.*
4. **Extraction:** *The challenger runs $z^* \leftarrow \mathtt{Ext}^{L^*}(y^*, \varphi^*, \mathtt{EK})$.*
5. *$P^*$ wins if (a) the pair $(y^*, L^*)$ was not part of a simulator query, (b) $\mathtt{Verify}^{L^*}(y^*, \varphi^*) = 1$, and (c) for all $x'$ such that $f(x') = z^*$ we have $R(x', y^*) = 0$.*[6]

*In the case when $f$ is the identity function, we simply say that $\Pi$ is* true-simulation extractable (tSE).

We give several variations of this new primitive. First, we define *one-time* simulation extractability, in which the adversary $P^*$ is only given *a single* query to the simulation oracle $\mathcal{SIM}_{\mathrm{TK}}(\cdot)$. Second, we define the notion of *strong* simulation extractability by changing the winning condition so that $P^*$ is now required to output a new statement/argument pair instead of a new statement. More formally, condition 5a becomes: the tuple $(y^*, L^*, \varphi^*)$ is new, that is, either $(y^*, L^*)$ was not part of a simulator query, or if it was, the argument $\varphi^*$ is different from the one(s) given to $P^*$ by $\mathcal{SIM}_{\mathrm{TK}}(\cdot)$. We observe that we can generically construct strong $f$-tSE NIZK arguments from (standard) $f$-tSE NIZK arguments if we additionally use a strongly-secure one-time signature. In particular, the prover now computes the standard $f$-tSE argument, signs it, and attaches the verification key $vk$ to the public label. To verify, we first check that the signature is valid and then verify the $f$-tSE argument.

Finally, we say that an NIZK argument $\Pi$ is *any-simultation $f$-extractable ($f$-aSE)* (similar to the notion of simulation-sound extractability of [31]) if the adversary $P^*$ instead has access to a modified simulation oracle $\widetilde{\mathcal{SIM}}_{\mathrm{TK}}(\cdot)$ that responds to all simulation queries without checking that $R(x, y) = 1$ (and hence might also give simulated arguments of false statements). In this work we do not make use of this variation, but state it here because as we will see, this notion has been implicitly used in prior works. However, $f$-aSE is a stronger notion than $f$-tSE and is *not needed*, as we will show that $f$-tSE is sufficient in constructing leakage-resilient signatures and CCA-encryption.

## 4 Generic Constructions

In this section we give generic constructions of leakage-resilient hard relations, signatures, and CCA-secure encryption. In the latter two we use the $f$-tSE NIZK primitive that we defined in Section 3. Finally, we give a construction of $f$-tSE NIZK arguments.

LEAKAGE-RESILIENT HARD RELATIONS. We begin by showing how to generically construct leakage-resilient hard relations from SPR relations. Informally, we say that a relation $R$ is *second-preimage resistant (SPR)* if given a random $(x, y) \in R$ it is difficult to find $x' \neq x$ such that $(x', y) \in R$. We formalize this in the following definition.

**Definition 6 (Second-Preimage Resistant (SPR) Relation).** *A relation $R$ with a randomized PPT sampling algorithm* KeyGen *is* second-preimage resistant *if:*

- *For any $(x, y) \leftarrow \mathtt{KeyGen}(1^\lambda)$, we have $(x, y) \in R$.*
- *There is a poly-time algorithm that decides if $(x, y) \in R$.*

[6] In other words, the adversary wins if the extractor fails to extract a good value $z^*$ which corresponds to at least one valid witness $x'$; i.e. $f(x') = z^*$. For the identity function, $f(x) = x$, this corresponds to the statement: $R(z^*, y) = 0$.

– *For any PPT algorithm* $\mathcal{A}$, $\Pr[(x', y) \in R \wedge x' \neq x] \leq negl(\lambda)$, *where the probability is taken over* $(x, y) \leftarrow \mathtt{KeyGen}(1^\lambda), x' \leftarrow \mathcal{A}(x, y)$.

*We define the* average-case pre-image entropy *of the SPR relation to be* $\mathbf{H}_{avg}(R) = \widetilde{\mathbf{H}}_\infty(X \mid Y)$ *, where random variables* $(X, Y)$ *are distributed according to* $\mathtt{KeyGen}(1^\lambda)$. *(We refer the reader to the full version [18] for the definition of* $\widetilde{\mathbf{H}}_\infty(X \mid Y)$*.)*

**Theorem 1.** *If* $R(x, y)$ *is an SPR relation, then it is also an* $\ell$*-leakage resilient hard relation for* $\ell = \mathbf{H}_{avg}(R) - \omega(\log \lambda)$*, where* $\lambda$ *is the security parameter.*

LEAKAGE-RESILIENT SIGNATURES. We give a generic construction of leakage-resilient signatures based on leakage-resilient hard relations and tSE-NIZK arguments. Let $R(x, y)$ be an $\ell$-leakage resilient hard relation with sampling algorithm $\mathtt{KeyGen}_R(1^\lambda)$. Let $\Pi = (\mathtt{Setup}, \mathtt{Prove}, \mathtt{Verify})$ be a tSE-NIZK argument for relation $R$ supporting labels. Consider the following signature scheme:

– $\mathtt{KeyGen}(1^\lambda)$ : Output $sk = x$ and $vk = (\mathrm{CRS}, y)$ where $(x, y) \leftarrow \mathtt{KeyGen}_R(1^\lambda)$ , $(\mathrm{CRS}, \mathrm{TK}, \mathrm{EK}) \leftarrow \mathtt{Setup}(1^\lambda)$.
– $\mathtt{Sign}_{sk}(m)$ : Output $\sigma = \varphi$ where $\varphi \leftarrow \mathtt{Prove}^m(x, y)$. (Note that $m$ is the *label*.)
– $\mathtt{SigVer}_{vk}(m, \sigma)$: Output $\mathtt{Verify}^m(y, \sigma)$.

**Theorem 2.** *If* $R(x, y)$ *is an* $\ell$*-leakage resilient hard relation and* $\Pi$ *is a labeled tSE-NIZK argument for* $R$*, then the above scheme is an* $\ell$*-leakage resilient signature scheme.*

LEAKAGE-RESILIENT CCA-SECURE ENCRYPTION. We give a generic construction of leakage-resilient CCA-secure encryption from leakage-resilient CPA-secure encryption and strong $f$-tSE NIZK arguments. Let $\mathcal{E} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ be an $\ell$-LR-CPA secure encryption scheme and let $\Pi = (\mathtt{Setup}, \mathtt{Prove}, \mathtt{Verify})$ be a one-time strong $f$-tSE NIZK argument for the relation $R_{enc} = \{(\,(m, r), (pk, c)\,) \mid c = \mathtt{Enc}_{pk}(m; r)\}$. where $f(m, r) = m$ (i.e. the extractor only needs to extract the message $m$, but not the randomness $r$ of encryption). We show how to use $\mathcal{E}, \Pi$ to construct an $\ell$-LR-CCA encryption scheme $\mathcal{E}^*$. Define $\mathcal{E}^* = (\mathtt{KeyGen}^*, \mathtt{Enc}^*, \mathtt{Dec}^*)$ by:

– $\mathtt{KeyGen}^*(1^\lambda)$: Output $pk = (pk_0, \mathrm{CRS})$, $sk = sk_0$ where $(pk_0, sk_0) \leftarrow \mathtt{KeyGen}(1^\lambda)$ , $(\mathrm{CRS}, \mathrm{TK}, \mathrm{EK}) \leftarrow \mathtt{Setup}(1^\lambda)$.
– $\mathtt{Enc}^*_{pk}(m; r)$: Output $C = (c, \pi)$ where $c \leftarrow \mathtt{Enc}_{pk_0}(m; r)$ , $\pi \leftarrow \mathtt{Prove}_{\mathrm{CRS}}((pk_0, c), (m, r))$.
– $\mathtt{Dec}^*_{sk}(C)$: Parse $C = (c, \pi)$. If $\pi$ verifies output $\mathtt{Dec}_{sk}(c)$, else output $\perp$.

**Theorem 3.** *Assume that* $\mathcal{E}$ *is* $\ell$*-LR-CPA secure, and* $\Pi$ *is a strong one-time* $f$*-tSE NIZK argument for the relation* $R_{enc}$ *where, for any witness* $(m, r)$*, we define* $f(m, r) = m$*. Then the scheme* $\mathcal{E}^*$ *defined above is* $\ell$*-LR-CCA secure.*

We note that if the tSE NIZK construction allows labels, then we can naturally extend our construction above to yield a $\ell$-LR-CCA encryption *with labels*, by simply putting the encryption labels into the NIZK proofs (and using them to verify the proofs).

TRUE-SIMULATION $f$-EXTRACTABLE ($f$-TSE) NIZK. Let $f$ be any efficiently computable function, and let $R(x, y)$ be an NP relation. We show how to construct an $f$-tSE NIZK argument $\Psi$ from any labeled CCA-secure encryption scheme, and (standard)

NIZK arguments. Let $\mathcal{E} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ be a CCA-secure encryption scheme supporting labels, and let $\Pi = (\mathtt{Setup}_\Pi, \mathtt{Prove}_\Pi, \mathtt{Verify}_\Pi)$ be an NIZK argument for the relation $R_\Pi = \{(\ (x,r),(y,c,pk,L)\ ) \mid R(x,y) = 1 \wedge c = \mathtt{Enc}^L_{pk}(f(x);r)\}$. We define $f$-tSE NIZK argument $\Psi$ (supporting labels) as follows:

- $\mathtt{Setup}(1^\lambda)$ : Output $\mathrm{CRS} = (\mathrm{CRS}_\Pi, pk)$, $\mathrm{TK} = \mathrm{TK}_\Pi$, $\mathrm{EK} = sk$ where $(pk, sk) \leftarrow \mathtt{KeyGen}(1^\lambda)$ , $(\mathrm{CRS}_\Pi, \mathrm{TK}_\Pi) \leftarrow \mathtt{Setup}_\Pi(1^\lambda)$.
- $\mathtt{Prove}^L(x, y; r)$: Output $\varphi = (c, \pi)$ where $c \leftarrow \mathtt{Enc}^L_{pk}(f(x); r)$ , $\pi \leftarrow \mathtt{Prove}_\Pi((x,r),(y,c,pk,L))$.
- $\mathtt{Verify}^L(y, \varphi)$: Parse $\varphi = (c, \pi)$ and run $\mathtt{Verify}_\Pi((y,c,pk,L),\pi)$.

**Theorem 4.** *If $\mathcal{E}$ is a labeled CCA-secure encryption scheme and $\Pi$ is an NIZK argument for relation $R_\Pi$, then $\Psi$ is a $f$-tSE NIZK argument for relation $R$.*

COMPARISON OF OUR GENERIC CONSTRUCTIONS TO PRIOR WORK. The idea of using an SPR relation to construct leakage-resilient hard relations was implicit in [4,42], and explicitly described in [5] for the case of leakage-resilient one-way functions.

Our constructions of leakage-resilient CCA encryption and signatures from tSE NIZKs bear significant resemblance to prior constructions. In particular, we observe that an alternate construction of tSE NIZK could be achieved by using a CPA-encryption scheme instead of a CCA one, and a ss-NIZK argument system [56] instead of a standard one. In fact, the resulting construction would yield an *any*-simulation extractable (aSE) NIZK argument. This instantiation of aSE NIZKs is implicitly used by [42] in their construction of leakage-resilient signatures. It is also used implicitly in the Naor-Yung "double-decryption" paradigm [49,55,56,45] for CCA security, which was later used in [48] to construct leakage-resilient CCA-encryption. However, as we have seen, tSE is sufficient for constructing *both* leakage-resilient signatures and CCA-encryption and thus, the stronger notion of aSE is not needed. Furthermore, given the current state of efficient encryption schemes and NIZK, the difference in efficiency between ss-NIZK and standard NIZK is *significantly* greater than the difference between CCA and CPA-secure encryption[7], thus making tSE superior in both simplicity and efficiency.

We note that our construction of tSE NIZKs (based on CCA-encryption and standard NIZKs) was implicitly used by [31] to construct signatures of group elements, and by [11] to construct efficient CCA-encryption with key-dependent message (KDM) security from KDM-secure CPA-encryption. Still, the abstraction of tSE has not been explicitly defined in prior work despite its apparent usefulness.

## 5 Instantiations

ASSUMPTIONS. We review several standard hardness assumptions on which we will base our constructions.

**Decisional Diffie-Hellman (DDH).** Let $\mathbb{G}$ be a group of prime order $q$. Let $g_1, g_2 \xleftarrow{\$} \mathbb{G}$ and $r, r_1, r_2 \xleftarrow{\$} \mathbb{Z}_q$. The decisional Diffie-Hellman (DDH) assumption states that the

[7] Informally, the difference between CCA and CPA-secure encryption is only 2 group elements, whereas the size of a ss-NIZK proof is *more than twice* the size of a standard NIZK proof.

following two distributions are computationally indistinguishable: $(\mathbb{G}, g_1, g_2, g_1^{r_1}, g_2^{r_2})$ and $(\mathbb{G}, g_1, g_2, g_1^{r}, g_2^{r})$.

Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be groups of prime order $q$ and let $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a non-degenerate efficiently computable bilinear map.

**Symmetric External Diffie-Hellman (SXDH) [59,9,6,26,61].** The symmetric external Diffie-Hellman assumption (SXDH) is that the DDH problem is hard in *both* $\mathbb{G}_1$ and $\mathbb{G}_2$. The assumption is clearly invalid for symmetric pairings (when $\mathbb{G}_1 = \mathbb{G}_2$), but is believed to hold when there is no efficiently computable mapping between $\mathbb{G}_1$ and $\mathbb{G}_2$.

**$K$-Linear [37,60] and DLIN [9].** Let $\mathbb{G}$ be a group of primer order $q$ and let $K \geq 1$ be constant. Let $g_0, g_1, \ldots, g_K \xleftarrow{\$} \mathbb{G}$ and $x_0, x_2, \ldots, x_K \xleftarrow{\$} \mathbb{Z}_q$. The $K$-Linear assumption states that the following two distributions are computationally indistinguishable: $(\mathbb{G}, g_0, g_1, \ldots, g_K, g_1^{x_1}, \ldots, g_K^{x_K}, g_0^{x_0})$, and $(\mathbb{G}, g_0, g_1, \ldots, g_K, g_1^{x_1}, \ldots, g_K^{x_K}, g_0^{X})$, with $X = \sum_{i=1}^{K} x_i$. Note that for $K = 1$, the $K$-Linear is the same as DDH, and that it does not hold when working with symmetric pairings. In that setting, the 2-Linear assumption is usually assumed to hold, and is often referred to as the Decisional Linear (DLIN) assumption. *Throughout this paper we assume the $K$-Linear assumption holds in both $\mathbb{G}_1$ and $\mathbb{G}_2$, which is the case when working with symmetric pairings, and slightly abuse notation when $K = 1$ and assume SXDH holds in that case.*

OUR INSTANTIATIONS. We show efficient instantiations of the leakage-resilient signature and CCA-secure encryption constructions described in Section 4. For each scheme, we give two instantiations based on bilinear maps: one secure under the SXDH assumption, and a second, secure under the DLIN assumption. The first can be used with asymmetric pairings, while the second applies to the case of symmetric pairings. We give details of all instantiations in the full version [18] but give a high-level idea below.

**Signatures.** Recall that to instantiate the signature scheme from Section 4, we need a leakage-resilient hard relation $R$ (which we will derive from an SPR relation) and a true-simulation extractable (tSE) NIZK argument, which we build from CCA-secure encryption and a standard NIZK argument for the relation $\{((x, r), (y, c, pk, L)) \,|R(x, y) = 1 \wedge c = \mathtt{Enc}_{pk}^{L}(f(x); r)\}$. We show our choice of instantiations for these components:

- *CCA-Secure Encryption:* Under both the SXDH and DLIN assumptions, we use efficient encryption schemes in the style of Cramer-Shoup [14,60].
- *NIZK Argument:* We use the Groth-Sahai proof system [33], which can be instantiated both under SXDH and DLIN.
- *SPR Relation:* Previous constructions of leakage-resilient primitives use the SPR function $g_1^{x_1} g_2^{x_2} \ldots g_n^{x_n}$. However, this function has the problem that the witness lies in the exponent. This means that we cannot combine it with an encryption scheme for elements in $\mathbb{G}$ (unless each witness component is committed bit by bit which, among other things, results in proofs growing linearly with the security parameter), and unfortunately encryption schemes for messages in $\mathbb{Z}_q$ cannot be combined with the Groth-Sahai system. We therefore construct two new SPR relations based on pairing-product equations. For our SXDH instantiation, we use the relation $e(h_1, x_1)\ e(h_2, x_2) \ldots e(h_n, x_n) = e(y, \tilde{g})$, where $\tilde{g}$ is a generator of $\mathbb{G}_2$. We prove that this relation is SPR under the SXDH assumption. In the DLIN case, we

use the relation: $e(h_1, x_1)\ e(h_2, x_2) \ldots e(h_n, x_n) = e(y_1, g)\ ,\ e(\hbar_1, x_1)\ e(\hbar_2, x_2) \ldots e(\hbar_n, x_n) = e(y_2, g)$, where $g$ is a generator of $\mathbb{G}$. We prove that this relation is SPR under the DLIN assumption. To achieve a $(1 - \epsilon)$ leakage ratio, we let $n$ (the number of witness components) in the SPR relation be inversely proportional to $\epsilon$.

**Theorem 5.** *Let $\mathbb{G}_1, \mathbb{G}_2$ be groups of primer order $q$. For any $\epsilon > 0$, there exists a $(1 - \epsilon)|sk|$-leakage resilient signature scheme, secure under the SXDH assumption, using signatures consisting of $(9/\epsilon)(1 + \omega(\log \lambda)/\log q) + 24$ group elements and 2 elements in $\mathbb{Z}_q$. Similarly, for any $\epsilon > 0$, there exists a $(1 - \epsilon)|sk|$-leakage resilient signature scheme, secure under the DLIN assumption, using signatures consisting of $(19/\epsilon)(2 + \omega(\log \lambda)/\log q) + 70$ group elements and 6 elements in $\mathbb{Z}_q$.*

**CCA-Secure Encryption.** Recall that for leakage-resilient encryption, we need leakage-resilient CPA-secure encryption, standard CCA-secure encryption and strong tSE NIZK, which we can get from combining regular tSE NIZK with a strong one-time signature. We build regular tSE NIZK from CCA-secure encryption and regular NIZK. We describe our choices for each of these below.

- *LR-CPA-Secure Encryption:* We construct a new leakage-resilient CPA-secure encryption scheme for our purpose in the style of ElGamal (similar to ones used in [48,11] but making it more efficient). The leakage that our new CCA-secure encryption tolerates is the same as the leakage tolerated by the CPA-secure scheme. Informally, we achieve a $(1-\epsilon)$ leakage ratio in the CPA-secure scheme by increasing the number of generators used in the public key and ciphertext. This number will be inversely proportional to $\epsilon$.
- *CCA-Secure Encryption:* Under both the SXDH and DLIN assumptions, we use efficient encryption schemes in the style of Cramer-Shoup [14,60].
- *NIZK Argument:* We use the Groth-Sahai proof system [33], which can be instantiated both under SXDH and DLIN.
- *One-Time Signature:* We observe that *any* strong one-time signature secure under these assumptions can be used. Here, we opt for the scheme of [31], secure under the Discrete Log assumption (implied by both SDXH and DLIN), because its signature size is small, namely 2 elements in $\mathbb{Z}_q$.

**Theorem 6.** *Let $\mathbb{G}_1, \mathbb{G}_2$ be groups of primer order $q$. For any $\epsilon > 0$, there exists a $(1 - \epsilon)|sk|$-leakage resilient encryption scheme, secure under the SXDH assumption, using ciphertexts consisting of $(2/\epsilon)(2 + \lambda/\log q) + 15$ group elements and 2 elements in $\mathbb{Z}_q$. Similarly, for any $\epsilon > 0$, there exists a $(1 - \epsilon)|sk|$-leakage resilient encryption scheme, secure under the DLIN assumption, using ciphertexts consisting of $(3/\epsilon)(3 + \lambda/\log q) + 34$ group elements and 2 elements in $\mathbb{Z}_q$.*

## 6 Other Applications

Once we have efficient leakage-flexible signature schemes, we observe that the standard signature-based ID scheme, where the verifier asks the prover to sign a random message, easily extends to the leakage setting. Moreover, the resulting actively secure ID scheme inherits its relative leakage from the corresponding signature scheme, and satisfies the

strongest notion of "anytime-leakage" [4], where the leakage can occur even during the impersonation attack. Although our method is pretty simple, we notice that the other two popular methods of building ID schemes — the use of $\Sigma$-protocols for hard relations analyzed in [4] (see first two rows of Tables 3), and the use of CCA-secure encryption (where the prover decrypts a random challenge ciphertext) — inherently do not allow us to obtain optimal results, even when instantiated with leakage-flexible hard relations or CCA-encryption schemes.

Finally, we obtain two efficient leakage-flexible AKA protocols. First, similarly to the case of ID schemes, we can obtain leakage-resilient AKA schemes from any leakage-resilient signature scheme, as formally explained in [4]. The idea is to essentially sign every flow of a standard Diffie-Hellman-based protocol, but with a leakage-resilient signature scheme. We notice, though, that the resulting protocol is not *deniable*. Namely, the transcript of the protocol leaves irrefutable evidence that the protocol took place. Motivated by this deficiency, we design another general AKA protocol based on CCA-encryption. The details are given in the full version [18], but, intuitively, the parties encrypt the flows of the standard Diffie-Hellman-based protocol, effectively proving their identities by successfully re-encrypting the appropriate flows. Although we do not formalize this, this protocols is "deniable", because the transcript of the protocol can be simulated without the knowledge of parties' secret keys. To the best of our knowledge, this protocol was not suggested and analyzed even in the leakage-free setting, where it appears interesting already. Here we actually show that our (new) deniable AKA protocol works even in the presence of leakage.

## Aknowledgments

We would like to thank Victor Shoup for helpful discussions, as well as the anonymous reviewers for their useful suggestions.

## References

1. In: Proceedings of 44th Symposium on Foundations of Computer Science (FOCS 2003), IEEE Computer Society, Los Alamitos (2003)
2. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009)
3. Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-key encryption in the bounded-retrieval model. Gilbert [28], pp. 113–134 (2010)
4. Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. Halevi [35], pp. 36–54 (2009)
5. Alwen, J., Dodis, Y., Wichs, D.: Survey: Leakage resilience and the bounded retrieval model. In: Kurosawa, K. (ed.) Information Theoretic Security. LNCS, vol. 5973, pp. 1–18. Springer, Heidelberg (2010)
6. Ballard, L., Green, M., de Medeiros, B., Monrose, F.: Correlation-resistant storage via keyword-searchable encryption. Cryptology ePrint Archive, Report 2005/417 (2005)
7. Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 513–525. Springer, Heidelberg (1997)

8. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: STOC, pp. 103–112. ACM, New York (1988)
9. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
10. Boneh, D., DeMillo, R.A., Lipton, R.J.: On the importance of checking cryptographic protocols for faults (extended abstract). In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 37–51. Springer, Heidelberg (1997)
11. Camenisch, J., Chandran, N., Shoup, V.: A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. Joux [39], pp. 351–368 (2009)
12. Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., Sahai, A.: Exposure-resilient functions and all-or-nothing transforms. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 453–469. Springer, Heidelberg (2000)
13. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: STOC, pp. 494–503 (2002)
14. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
15. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
16. Crescenzo, G.D., Lipton, R.J., Walfish, S.: Perfectly secure password protocols in the bounded retrieval model. Halevi and Rabin [36], pp. 225–244 (2006)
17. Dodis, Y., Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Public-key encryption schemes with auxiliary inputs. Micciancio [47], pp. 361–381 (2010)
18. Dodis, Y., Haralambiev, K., Lopez-Alt, A., Wichs, D.: Efficient public-key cryptography in the presence of key leakage. Cryptology ePrint Archive, Report 2010/154 (2010)
19. Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: Mitzenmacher, M. (ed.) STOC, pp. 621–630. ACM, New York (2009)
20. Dodis, Y., Sahai, A., Smith, A.: On perfect and adaptive security in exposure-resilient cryptography. Pfitzmann [52], pp. 301–324 (2001)
21. Dziembowski, S.: Intrusion-resilience via the bounded-storage model. Halevi and Rabin [36], pp. 207–224 (2006)
22. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS, pp. 293–302. IEEE Computer Society, Los Alamitos (2008)
23. Faust, S., Kiltz, E., Pietrzak, K., Rothblum, G.N.: Leakage-resilient signatures. Micciancio [47], pp. 343–360 (2010)
24. Faust, S., Rabin, T., Reyzin, L., Tromer, E., Vaikuntanathan, V.: Protecting circuits from leakage: the computationally-bounded and noisy cases. Gilbert [28], pp. 135–156 (2010)
25. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
26. Galbraith, S.D., Rotger, V.: Easy decision-diffie-hellman groups. LMS Journal of Computation and Mathematics 7 (2004)
27. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: Concrete results. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 251–261. Springer, Heidelberg (2001)
28. Gilbert, H. (ed.): EUROCRYPT 2010. LNCS, vol. 6110. Springer, Heidelberg (2010)
29. Goldwasser, S., Kalai, Y., Peikert, C., Vaikuntanathan, V.: Robustness of the learning with errors assumption. In: Innovations in Computer Science, ICS (2010)

30. Goldwasser, S., Kalai, Y.T.: On the (in)security of the fiat-shamir paradigm. FOCS [1], p. 102 (2003)
31. Groth, J.: Simulation-sound nizk proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)
32. Groth, J.: Fully anonymous group signatures without random oracles. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 164–180. Springer, Heidelberg (2007)
33. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
34. Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: cold-boot attacks on encryption keys. ACM Commun. 52(5), 91–98 (2009)
35. Halevi, S. (ed.): CRYPTO 2009. LNCS, vol. 5677. Springer, Heidelberg (2009)
36. Halevi, S., Rabin, T. (eds.): TCC 2006. LNCS, vol. 3876. Springer, Heidelberg (2006)
37. Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (2007)
38. Ishai, Y., Sahai, A., Wagner, D.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (2003)
39. Joux, A. (ed.): EUROCRYPT 2009. LNCS, vol. 5479. Springer, Heidelberg (2009)
40. Juma, A., Vahlis, Y.: Protecting cryptographic keys against continual leakage. In: Rabin, T. (ed.) Advances in Cryptology – CRYPTO 2010. LNCS, vol. 6223, pp. 41–58. Springer, Heidelberg (2010)
41. Kamp, J., Zuckerman, D.: Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. FOCS [1], pp. 92–101 (2003)
42. Katz, J., Vaikuntanathan, V.: Signature schemes with bounded leakage resilience. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 703–720. Springer, Heidelberg (2009)
43. Kocher, P.C.: Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
44. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
45. Lindell, Y.: A simpler construction of cca2-secure public-keyencryption under general assumptions. J. Cryptology 19(3), 359–377 (2006)
46. Micali, S., Reyzin, L.: Physically observable cryptography (extended abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004)
47. Micciancio, D. (ed.): TCC 2010. LNCS, vol. 5978. Springer, Heidelberg (2010)
48. Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. Halevi [35], pp. 18–35 (2009)
49. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC, pp. 427–437. ACM, New York (1990)
50. Pass, R., Rosen, A.: Concurrent non-malleable commitments. In: FOCS, pp. 563–572. IEEE Computer Society, Los Alamitos (2005)
51. Pass, R., Rosen, A.: New and improved constructions of non-malleable cryptographic protocols. In: Gabow, H.N., Fagin, R. (eds.) STOC, pp. 533–542. ACM, New York (2005)
52. Pfitzmann, B. (ed.): EUROCRYPT 2001. LNCS, vol. 2045. Springer, Heidelberg (2001)
53. Pietrzak, K.: A leakage-resilient mode of operation. Joux [39], pp. 462–482 (2009)
54. Quisquater, J.-J., Samyde, D.: Electromagnetic analysis (ema): Measures and countermeasures for smart cards. In: Attali, I., Jensen, T.P. (eds.) E-smart 2001. LNCS, vol. 2140, pp. 200–210. Springer, Heidelberg (2001)

55. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
56. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: FOCS, pp. 543–553 (1999)
57. Santis, A.D., Crescenzo, G.D., Ostrovsky, R., Persiano, G., Sahai, A.: Robust non-interactive zero knowledge. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 566–598. Springer, Heidelberg (2001)
58. Santis, A.D., Persiano, G.: Zero-knowledge proofs of knowledge without interaction (extended abstract). In: FOCS, pp. 427–436. IEEE, Los Alamitos (1992)
59. Scott, M.: Authenticated id-based key exchange and remote log-in with simple token and pin number. Cryptology ePrint Archive, Report 2002/164 (2002)
60. Shacham, H.: A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074 (2007)
61. Verheul, E.R.: Evidence that xtr is more secure than supersingular elliptic curve cryptosystems. J. Cryptology 17(4), 277–296 (2004)