Signature Schemes Secure against Hard-to-Invert Leakage *

Sebastian Faust, Carmit Hazay, Jesper Buus Nielsen,
Peter Sebastian Nordholt, Angela Zottarel

Aarhus University, Denmark

Abstract. In the auxiliary input model an adversary is allowed to see a *computationally hard-to-invert function* of the secret key. The auxiliary input model weakens the bounded leakage assumption commonly made in leakage resilient cryptography as the hard-to-invert function may information-theoretically reveal the entire secret key. In this work, we propose the *first* constructions of digital signature schemes that are secure in the auxiliary input model. Our main contribution is a digital signature scheme that is secure against *chosen message attacks* when given an *exponentially hard-to-invert function* of the secret key. As a second contribution, we construct a signature scheme that achieves security for *random messages* assuming that the adversary is given a *polynomial-time* hard to invert function. Here, polynomial-hardness is required even when given the entire public-key – so called *weak* auxiliary input security. We show that such signature schemes readily give us auxiliary input secure identification schemes.

* Supported by the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation.

1 Introduction

Modern cryptography analyzes the security of cryptographic algorithms in the *black-box* model. An adversary may view the algorithm's inputs and outputs, but the secret key as well as all the internal computation remains perfectly hidden. Unfortunately, the assumption of perfectly hidden keys does not reflect practice where keys frequently get compromised for various reasons. An important example is side-channel attacks that exploit *information leakage* from the implementation of an algorithm. Side-channel attacks do not only allow the adversary to gain partial knowledge of the secret key thereby making security proofs less meaningful, but in many cases may result in complete security breaches.

In the last years, significant progress has been made within the theory community to incorporate information leakage into the black-box model (cf. [MR04, DP08, AGV09, ADW09, DKL09] [NS09, FKPR10, DHLAW10a, LRW11] and many more). To this end, these works develop new models to formally describe the information leakage, and design new schemes that can be proven secure therein. The leakage is typically characterized by a *leakage function* h that takes as input the secret key sk and reveals $h(\mathsf{sk})$—the so-called *leakage*—to the adversary. Of course, we cannot allow h to be any function as otherwise it may just reveal the complete secret key. Hence certain restrictions on the class $\mathcal{H}$ of admissible leakage functions are necessary.

With very few exceptions (outlined in the next section) most works assume some form of quantitative restriction on the amount of information leaked to an adversary. More formally, in the *bounded* leakage model, it is assumed that $\mathcal{H}$ is the set of all polynomial-time computable functions $h : \{0,1\}^{|\mathsf{sk}|} \to \{0,1\}^{\lambda}$ with $\lambda \ll |\mathsf{sk}|$. This restriction can be weakened in many cases. Namely, instead of requiring a concrete bound λ on the amount of leakage, it often suffices that given the leakage $h(\mathsf{sk})$ the secret key still has a "sufficient" amount of min-entropy left [DP08, Pie09, NS09, DHLAW10b]. This so-called *noisy leakage* models real-world leakage functions more accurately as now the leakage can be arbitrarily large. Indeed, real-world measurements of physical phenomenons are usually described by several megabytes or even gigabytes of information rather than by a few bits.

While security against bounded or noisy leakage often provides a first good indication for the security of a cryptographic implementation, in practice leakage typically information theoretically determines the entire secret key [Sta11]. The only difficulty of a side-channel adversary lies in extracting the relevant key information efficiently. Formally, this can be modeled by assuming that $\mathcal{H}$ is the set of all polynomial-time computable functions such that given $h(\mathsf{sk})$ it is still computationally "hard" to compute sk. Such *hard-to-invert* leakage are a very natural generalization of both the bounded leakage model and the noisy leakage model, and is the focus of this work. More concretely, we will analyze the security of digital signature schemes in the presence of *hard-to-invert* leakage. We show somewhat surprisingly that simple variants of constructions for the bounded leakage setting [KV09, DHLAW10b, DHLAW10a, BSW11, MTVY11] achieve also security with respect to the more *general* class of hard-to-invert leakage.

1.1 The Auxiliary Input Model

The *auxiliary input model* of Dodis, Kalai and Lovett [DKL09] introduced the notion of security of cryptographic schemes in the presence of computationally hard-to-invert leakage. They propose constructions for secret key encryption with IND-CPA and IND-CCA security against an adversary who obtains an arbitrary polynomial-time computable hard-to-invert leakage $h(\mathsf{sk})$. Security is shown to hold under a non-standard LPN-related assumption with respect to any *exponentially* hard-to-invert function. We say that h is an exponentially hard-to-invert function of the secret key sk, if there exists a constant $c > 0$ such that, for sufficiently large $k = |\mathsf{sk}|$, any PPT adversary $\mathcal{A}$ has probability of at most 2^{-ck} in inverting $h(\mathsf{sk})$. Notice that the result gets stronger, and the class of admissible leakage function gets larger, if c is smaller.

In a follow-up paper, and most relevant for our work, Dodis et al. [DGK+10] study the setting of public key encryption. They show that the BHHO encryption scheme [BHHO08] based on DDH and variants of the GPV encryption scheme [GPV08] based on LWE are secure with respect to auxiliary input leakage. All their schemes remain secure under *sub-exponentially* hard-to-invert leakage. That is, a function h is sub-exponentially hard-to-invert if there exists a constant $1 > c > 0$ such that $h(\mathsf{sk})$ can be inverted with probability at most 2^{-k^c}.

In the public key setting, some important subtleties arise which are also important for our work.

1. We shall allow the leakage to depend also on the corresponding public key pk. One approach to model this is to let the adversary adaptively choose the leakage function after seeing the public key pk [AGV09]. An alternative that is taken in the work of Dodis et al. [DGK+10] assumes admissible leakage functions $h : \{0,1\}^{|\mathsf{sk}|+|\mathsf{pk}|} \to \{0,1\}^*$, where it is hard to compute sk given $h(\mathsf{pk}, \mathsf{sk})$.
2. The public key itself may leak information about the secret key. To illustrate this, consider a contrived scheme, where the public key pk contains the first $k/2$ bits of the secret key in clear. Suppose we want to prove security for leakage functions h with the property that given $h(\mathsf{pk}, \mathsf{sk})$, it is at least $2^{-k/2}$ hard to compute the secret key sk. Given the public key pk and such leakage that reveals the last $k/2$ bits of the secret key, the scheme from above gets completely insecure. To handle this issue, Dodis et al. propose a *weaker* notion of auxiliary input security, which assumes that a function is an admissible leakage if it is hard to compute the secret key *even* when given the public key.

For ease of presentation, we mainly consider in this work this weaker notion of auxiliary input security. As shown in [DGK+10], when the public key is short this notion implies security for functions h solely under the assumption that given $h(\mathsf{pk}, \mathsf{sk})$ it is computationally hard to compute sk (i.e., without defining hardness with respect to pk). The underlying idea is that the public key can be guessed within the proof, which implies that the hardness assumption gets stronger when applying this proof technique. Specifically, security is obtained in the presence of exponentially hard-to-invert leakage functions. We further note that this weaker notion already suffices for composition of different cryptographic schemes using the same public key. For instance, consider an encryption and signature scheme sharing the same public key. If the encryption scheme is weakly secure with respect to any polynomially hard-to-invert leakage function,[1] then the scheme remains secure even if the adversary sees arbitrary signatures, as these signatures can be viewed as hard-to-invert leakage.

Recently, Brakerski and Goldwasser [BG10] and Brakerski and Segev [BS11] proposed further constructions of public key encryptions secure against auxiliary input leakage. In the former, the authors show how to construct a public key encryption scheme secure against sub-exponentially hard-to-invert leakage, based on the QR and DCR hardness assumptions. In the latter, the concept of security against auxiliary input has been introduced in the context of deterministic public key encryption, and several secure constructions were proposed based on DDH and subgroup indistinguishability assumptions.

1.2 Our Contributions

Despite significant progress on constructing encryption schemes in the auxiliary input model, the question of whether digital signature schemes can be built with security against hard-to-invert leakage has remained open so far. This is somewhat surprising as a large number of constructions for the bounded and noisy leakage setting are known [ADW09, KV09, DHLAW10b, DHLAW10a, BSW11, MTVY11]. In this paper, we close this gap and propose the first constructions for digital

[1] A function h is *polynomially* hard-to-invert auxiliary information, if any probabilistic polynomial-time adversary computes sk with negligible probability, given the leakage $h(\mathsf{sk}, \mathsf{pk})$.

signature schemes with security in the auxiliary input model. As a first contribution of our work, we propose new security notions that are attainable in the presence of hard-to-invert leakage. We then show that constructions that have been proven to be secure when the amount of leakage is bounded, also achieve security in the presence of hard-to-invert leakage. In a nutshell, our results can be summarized as follows:

1. As shown below, existential unforgeability is unattainable in the presence of polynomially hard-to-invert leakage. We thus weaken the security notion by focusing on the setting where the challenge message is chosen uniformly at random. Our construction uses ideas from [MTVY11] to achieve security against polynomially hard-to-invert leakage when prior to the challenge message the adversary only has seen signatures for random messages. Such schemes can straightforwardly be used to construct identification schemes with security against any polynomially hard-to-invert leakage (cf. Sections 3.2 and 4).
2. We show that the *generic* constructions proposed in [KV09, DHLAW10b, BSW11] achieve the strongest notion of security, namely *existentially unforgeable under chosen message attacks*, if we restrict the adversary to obtain only *exponentially hard-to-invert* leakage. As basic ingredients these schemes use a family of second preimage resistant hash functions, an IND-CCA secure public key encryption scheme with labels and a reusable CRS-NIZK proof system. For our result to be meaningful, we require both the decryption key and the simulation trapdoor of the underlying encryption scheme to be short when compared to the length of the signing key for the signature scheme (cf. Section 3.3).
3. We show an instantiation of this generic transformation that satisfies our requirements on the length of the keys based on the 2-Linear hardness assumption in pairing based groups, using the Groth-Sahai proof system [GS08] (cf. Section 5).

We elaborate on these results in more detail below.

Polynomially hard-to-invert leakage and random challenges. Importantly, security with respect to polynomially hard-to-invert leakage is impossible if the message for which the adversary needs to output a forgery, is fixed at the time the leakage function is chosen. This is certainly the case for the standard security notion of existential unforgeability. One potential weakening of the security definition is by requiring the adversary to forge a signature on a random challenge message. In the case when the challenge messages is sampled uniformly at random, even though the leakage may reveal signatures for some messages, it is very unlikely that the adversary hits a forgery for the challenge message.

Specifically, inspired by the work of Malkin et al. [MTVY11], we propose a construction that guarantees security in the presence of *any polynomially hard-to-invert* leakage, when the challenge message is chosen uniformly at random. The scheme uses the message as the CRS for a non-interactive zero-knowledge proof of knowledge ($\mathsf{NIZKPoK}$). To sign, we use the CRS to prove knowledge of sk such that $\mathsf{vk} = H(\mathsf{sk})$, where H is a second preimage resistant hash function. Therefore, if an adversary forges a signature given vk and the leakage $h(\mathsf{vk}, \mathsf{sk})$ with non-negligible probability, we can use this forgery to extract a preimage of vk which either contradicts the second preimage resistance of H or the assumption that h is polynomially hard-to-invert. An obvious drawback of this scheme is that prior to outputting a forgery for the challenge message the adversary only sees signatures on random messages. Finally, as a natural application of such schemes, we show that auxiliary input security for signatures carries over to auxiliary input security of identification schemes. Hence, our scheme can be readily used to build simple identification schemes with security against any polynomially hard-to-invert leakage function.

Exponentially hard-to-invert leakage and existential unforgeability. The standard security notion for signature schemes is existential unforgeability under adaptive chosen-message attacks [GMR88]. Here, one requires that an adversary cannot forge a signature of any message

m, even when given access to a signing oracle. We strengthen this notion and additionally give the adversary leakage $h(\mathsf{vk}, \mathsf{sk})$, where h is some admissible function from class $\mathcal{H}$. It is easy to verify that no signature scheme can satisfy this security notion when the only assumption that is made about $h \in \mathcal{H}$, is that it is polynomially hard to compute sk given $h(\mathsf{vk}, \mathsf{sk})$. The reason for this is as follows. Since the secret key must be polynomially hard to compute even given some set of signatures (and the public key), a signature is an admissible leakage function with respect to $\mathcal{H}$. Hence, a forgery is a valid leakage. This observation holds even when we define the hardness of h with respect to the public key as well.

Our first observation towards constructing signatures with auxiliary input security is that the above issues do not necessarily arise when we consider the more restricted class of functions that maintain (sub)-exponentially hardness of inversion. Suppose, for concreteness, that there exists a constant $1 > c > 0$ such that there exists a probabilistic polynomial-time algorithm, taking as input a signature and the public key and outputting sk with probability p. Here, we assume that $\mathsf{negl}(k) \geq p \gg 2^{-k^c}$ for some negligible function $\mathsf{negl}(\cdot)$. Then, if we let $\mathcal{H}$ be the class of functions with hardness at least 2^{-k^c}, the signing algorithm is not in $\mathcal{H}$ and hence the artificial counterexample from above does not work anymore! We instantiate this idea by adding an encryption $C = \mathsf{Enc}_{\mathsf{ek}}(\mathsf{sk})$ of the signing key sk to each signature. The encryption key ek is part of the verification key of the signature scheme, but the decryption key dk associated with ek is not part of the signing key. However, we set up the scheme such that dk can be guessed with probability p. Interestingly, it turns out that recent constructions of leakage resilient signatures [KV09, DHLAW10b, BSW11], which originally were designed to protect against *bounded* leakage, use as part of the signature an encryption of the secret key. This enables us to prove that these schemes also enjoy security against exponentially hard-to-invert leakages.

One may object that artificially adding an encryption of the secret key to the signature is somewhat counter-intuitive as it seems to reduce the security of the signature scheme. However, all that is needed for this trick is that guessing dk is significantly easier than guessing sk. For a given security level we can therefore pick the length of dk first, as to achieve that security level. After that we can then pick the length of sk as to achieve meaningful leakage bounds. Our concrete security analysis allows to choose these keys as to achieve a given security. Note, also, that adding trapdoors to cryptographic schemes for what superficially only seems to be proof reasons is common in the field, non-interactive zero-knowledge being another prominent example.

For readers familiar with the security proof of the Katz-Vaikuntanathan scheme, we note that the crux of our new proof is that in the reduction we cannot generate a CRS together with its simulation trapdoor. Instead, to simulate signatures for chosen messages we will guess the simulation trapdoor. Fortunately, we can show that the loss from guessing the simulation trapdoor only effects the tightness in the reduction to the inversion hardness of the leakage functions. As we use a NIZK proof system with a short simulation trapdoor and only aim for exponential hard-to-invert leakage functions, we can successfully complete the reduction.

As a concrete example, we show how to instantiate our generic transformation using the Groth-Sahai proof system based on the 2-linear assumption. This gives us security with respect to any $2^{-6k'}$-hard-to-invert leakage. If we do not want to define the hardness also with respect to the public key, we can guess the public key and loose an additional factor of $2^{-3k'}$ in the hardness assumption. Here, $k' := \log(p)$ for a prime p that denotes the order of the group for which the 2-linear assumption holds, and the secret key of our scheme has length $k := \ell \cdot k'$ bits for some constant $\ell \in \mathbb{N}$.

1.3 A Road Map

In Section 2 we specify basic security definitions and our modeling for the auxiliary input setting. In Section 3 we present our signature schemes for random messages (Section 3.2) and chosen massage attack security (Section 3.3). In Section 4 we show how to use signatures on random messages to construct identification schemes with security against any polynomially hard-to-invert leakage. Finally, in Section 5 we show an instantiation of the later signature scheme under the 2-linear hardness assumption.

2 Preliminaries

Basic Notations. We denote the security parameter by k and by PPT a probabilistic polynomial-time algorithm. For a set S we write $x \leftarrow S$ to denote that x is sampled uniformly from S. We write $y \leftarrow \mathcal{A}(x)$ to indicate that y is the output of an algorithm $\mathcal{A}$ when running on input x. We denote by $\langle a, b \rangle$ the inner product of field elements a and b. We use $\mathsf{negl}(\cdot)$ to denote a negligible function $f : \mathbb{N} \to \mathbb{R}$ and we use the $\approx$ notation to denote computational indistinguishability of families of random variables.

2.1 Public Key Encryption Schemes

We introduce the notion of a labeled public key encryption scheme following the notation used in [DHLAW10b].

Definition 1 (LPKE). *We say that* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is a* labeled public key encryption scheme (LPKE) *with perfect decryption if:*

- KeyGen, *given a security parameter* k, *outputs keys* $(\mathsf{ek}, \mathsf{dk})$, *where* ek *is a public encryption key and* dk *is a secret decryption key. We denote this by* $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^k)$.
- Enc, *given the public key* ek, *a label* L *and a plaintext message* m, *outputs a ciphertext* c *encrypting* m. *We denote this by* $c \leftarrow \mathsf{Enc}^L(\mathsf{ek}, m)$.
- Dec, *given a label* L, *the secret key* dk *and a ciphertext* c, *with* $c = \mathsf{Enc}^L(\mathsf{pk}, m)$, *then with probability* 1 *outputs* m. *We denote this by* $m \leftarrow \mathsf{Dec}^L(\mathsf{dk}, c)$.

Definition 2 (IND-LCCA secure encryption scheme). *We say that a labeled public key encryption scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* IND-LCCA secure encryption scheme *if, for every admissible PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, *there exists a negligible function* negl *such that the probability* $\textsf{IND-LCCA}_{\Pi,\mathcal{A}}(k)$ *that* $\mathcal{A}$ *wins the IND-LCCA game as defined below is at most* $\textsf{IND-LCCA}_{\Pi,\mathcal{A}}(k) \leq \frac{1}{2} + \mathsf{negl}(k)$.

- IND-LCCA game.

$$
\begin{aligned}
&(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^k) \\
&(L, m_0, m_1, history) \leftarrow \mathcal{A}_1^{\mathsf{Dec}^{(\cdot)}(\mathsf{dk},\cdot)}(\mathsf{ek}),\ s.t.\ |m_0| = |m_1| \\
&c \leftarrow \mathsf{Enc}^L(\mathsf{ek}, m_b),\ where\ b \leftarrow \{0,1\} \\
&b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}^{(\cdot)}(\mathsf{dk},\cdot)}(c, history) \\
&\mathcal{A}\ wins\ if\ b' = b.
\end{aligned}
$$

An adversary is admissible if it does not query $\mathsf{Dec}^{(\cdot)}(\mathsf{dk}, \cdot)$ *with* (L, c)

In this work we require a weaker notion, called IND-WLCCA, where the adversary cannot query the decryption oracle with label L. Namely, we change the definition of admissible to mean that the adversary never queries $\mathsf{Dec}^{(\cdot)}(\mathsf{dk}, \cdot)$ with any input of the form $(L, \cdot)$, where L is the label picked to compute the challenge. We discuss further details why this security notion is needed for our construction in Section 3.3.

2.2 Non-Interactive Proofs

A non-interactive proof (NIP) for a language L is a tuple of PPT algorithms $(\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$, where CRSGen generates a common reference string crs. The prover P takes as input $(\mathsf{crs}, x, \omega)$ for $(x, \omega) \in R_L$, the witness relation of L, and outputs a proof π. The verifier V takes as input (crs, x, π) and outputs 0 or 1 (respectively rejecting or accepting the proof). More formally,

Definition 3 (NIP). *A* non-interactive proof (NIP) *for a language L is a tuple of three PPT algorithms* $(\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$*, such that the following properties are satisfied:*

Completeness: *For every $(x, \omega) \in R_L$:* $\Pr[\mathsf{V}(\mathsf{crs}, x, \mathsf{P}(\mathsf{crs}, x, \omega)) = 1] = 1$.
Soundness *: For all PPT algorithms $\mathcal{A}$, $\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^k)$ and $x \notin L$:*

$$\Pr_{(x,\pi)\leftarrow\mathcal{A}(\mathsf{crs})}[\mathsf{V}(\mathsf{crs}, x, \pi) = 1] \leq \mathsf{negl}(k).$$

A non-interactive proof of knowledge (NIPoK) for a relation R_L is a NIP $(\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$ requiring the existence of a knowledge extractor that runs the prover P to get a witness ω. More formally,

Definition 4 (NIPoK). *A* non-interactive proof of knowledge (NIPoK) *for a relation R_L is a tuple of three PPT algorithms* $(\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$*, such that the following properties are satisfied:*

Completeness: *As in Def. 3.*
Knowledge soundness: *There exists a PPT knowledge extractor $E = (E_1, E_2)$ such that:*
a) for all PPT algorithms $\mathcal{A}$:

$$\Pr_{(\mathsf{crs},\mathsf{td}_e)\leftarrow E_1(1^k)}[\mathcal{A}(\mathsf{crs}) = 1] = \Pr_{\mathsf{crs}\leftarrow\mathsf{CRSGen}(1^k)}[\mathcal{A}(\mathsf{crs}) = 1].$$

b) for all PPT algorithms $\mathcal{A}$:

$$\Pr_{w\leftarrow E_2(\mathsf{crs},x,\mathsf{td}_e,\pi),(x,\pi)\leftarrow\mathcal{A}(\mathsf{crs}),(\mathsf{crs},\mathsf{td}_e)\leftarrow E_1(1^k)}[\mathsf{V}(\mathsf{crs}, x, \pi) = 0 \vee (x, \omega) \in R]$$

Definition 5 (NIZK). *A* non-interactive zero-knowledge proof (NIZK) *for a language L is a tuple of three PPT algorithms* $(\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$*, such that the following properties are satisfied:*

Completeness and Soundness: *As in Def. 3.*
Zero-knowledge: *There exist a PPT simulator $S = (S_1, S_2)$ such that*

$$|\Pr_{\mathsf{crs}\leftarrow\mathsf{CRSGen}(1^k)}[\mathcal{A}^{\mathcal{O}_0^{\mathsf{crs}}(\cdot)}(\mathsf{crs}) = 1] - \Pr_{(\mathsf{crs},\mathsf{td}_s)\leftarrow S_1(1^k)}[\mathcal{A}^{\mathcal{O}_1^{\mathsf{crs},\mathsf{td}_s}(\cdot)}(\mathsf{crs}) = 1]| \leq \frac{1}{2} + \mathsf{negl}$$

for all PPT adversaries $\mathcal{A}$, where $\mathcal{O}_0^{\mathsf{crs}}(\cdot)$ is the oracle with state crs*, where $\mathcal{O}_0^{\mathsf{crs}}(x, \omega) = P(\mathsf{crs}, x, \omega)$ if $(x, \omega) \in R$ and $\mathcal{O}_0^{\mathsf{crs}}(x, \omega) = \perp$ otherwise, and $\mathcal{O}_1^{\mathsf{crs},\mathsf{td}_s}$ is the oracle with state* $(\mathsf{crs}, \mathsf{td}_s)$*, where $\mathcal{O}_1^{\mathsf{crs},\mathsf{td}_s}(x, \omega) = S_2(\mathsf{crs}, x, \mathsf{td}_s)$ if $(x, \omega) \in R$ and $\mathcal{O}_1^{\mathsf{crs},\mathsf{td}_s}(x, \omega) = \perp$ otherwise. We say that the scheme is ZK if $\mathcal{A}$ may only query its oracle once. We say that it is reusable-CRS ZK if there is no restriction on how many times $\mathcal{A}$ can query its oracle.*

Definition 6 (NIZKPoK). *A* non-interactive zero-knowledge proof of knowledge (NIZKPoK) *for a relation R_L is a non-interactive proof of knowledge system with the properties completeness, knowledge soundness and zero-knowledge, defined above.*

2.3 Second Preimage Resistant Hash Functions

A family of hash functions $H = \{H_i\}_{i\in\{0,1\}^{\ell(k)}}$, where $\ell(k)$ is some function of the security parameter, is a family of polynomial time computable functions together with a PPT algorithm Gen_H. On input 1^k, Gen_H generates a key s for a function $H_s \in H$ where $H_s : \{0,1\}^{\ell'(k)} \to \{0,1\}^{\ell''(k)}$, for $\ell'(k) > \ell''(k)$. Loosely speaking, a family of hash functions H is second preimage resistant if, given $s \leftarrow \mathsf{Gen}_H(1^k)$ and an input x, it is infeasible for any PPT adversary to find x' such that $x \neq x'$ and $H_s(x) = H_s(x')$.

2.4 Signature Schemes

A signature scheme is a tuple of polynomial-time (PPT) algorithms $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ defined as follows. The key generation algorithm Gen, on input 1^k, where k is a security parameter, outputs a signing and a verification key $(\mathsf{sk}, \mathsf{vk})$. The signing algorithm Sig takes as input a message m and a signing key sk and outputs a signature σ. The verification algorithm Ver, on input (vk, m, σ) outputs either 0 or 1 (respectively rejecting or accepting the signature). A signature scheme has to satisfy the following correctness property: for any message m and keys $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^k)$

$$\Pr[\mathsf{Ver}(\mathsf{vk}, m, \mathsf{Sig}(\mathsf{sk}, m)) = 1] = 1$$

The standard security notion for a signature scheme is existentially unforgeability under chosen message attacks. A scheme is said to be secure under this notion if even after seeing signatures for chosen messages, no adversary can come up with a forgery for a new message. In this paper, we extend this security notion and give the adversary additional auxiliary information about the signing key. To this end, we define a set of admissible leakage functions $\mathcal{H}$ and allow the adversary to obtain the value $h(\mathsf{sk}, \mathsf{vk})$ for any $h \in \mathcal{H}$. Notice that by giving vk as input to the leakage function, we capture the fact that the choice of h may depend on vk.

Definition 7 (Existential Unforgeability under Chosen Message and Auxiliary Input Attacks (EU-CMAA)). *We say that a signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is* existential unforgeable against chosen message and auxiliary input attacks (EU-CMAA) with respect to $\mathcal{H}$ *if for all PPT adversaries $\mathcal{A}$ and any function $h \in \mathcal{H}$, the following probability $\Pr[\mathsf{CMA}_{\Sigma,\mathcal{A},h}(k) = 1]$ is negligible in k, where $\mathsf{CMA}_{\Sigma,\mathcal{A},h}(k)$ is defined as follows:*

Experiment	Oracle
Experiment $\mathsf{CMA}_{\Sigma,\mathcal{A},h}(k)$	***Oracle*** $\mathcal{O}(\mathsf{sk}, m)$
$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^k)$	*Return* $(m, \mathsf{Sig}(\mathsf{sk}, m))$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}(\mathsf{sk},\cdot)}(1^k, h(\mathsf{vk}, \mathsf{sk}), \mathsf{vk})$	
If $\mathsf{Ver}(\mathsf{vk}, m^*, \sigma^*) = 1$ *then return* 1, *else return* 0.	

We note that the leakage may also depend on $\mathcal{A}$'s signature queries as the function h may internally run $\mathcal{A}$, using the access to the secret key in order to emulate the entire security game, including the signature queries made by $\mathcal{A}$.

As outlined in the introduction, we are also interested in a weaker security notion where the adversary is required to output a forgery for a random message after seeing signatures for *random* messages. To this end, we extend the definition from above and let the signing oracle reply with random messages, as well as pick the challenge message at random. This is formally described in the following definition.

Definition 8 (Random Message Unforgeability under Random Message and Auxiliary Input Attacks (RU-RMAA)). *We say that a signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is* random message unforgeable against random message and auxiliary input attacks (RU-RMAA) with respect to $\mathcal{H}$ *if for all PPT adversary $\mathcal{A}$ and any function $h \in \mathcal{H}$, the probability $\Pr[\mathsf{RMA}_{\Sigma,\mathcal{A},h}(k) = 1]$ is negligible in k, where $\mathsf{RMA}_{\Sigma,\mathcal{A},h}(k)$ is defined as follows:*

Experiment	Oracle
Experiment $\mathsf{RMA}_{\Sigma,\mathcal{A},h}(k)$	***Oracle*** $\mathcal{O}(\mathsf{sk}, \cdot)$
$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^k)$	$m \leftarrow \mathcal{M}$
$m^* \leftarrow \mathcal{M}$, *where $\mathcal{M}$ is the message space*	*Return* $(m, \mathsf{Sig}(\mathsf{sk}, m))$
$\sigma^* \leftarrow \mathcal{A}^{\mathcal{O}(\mathsf{sk},\cdot)}(1^k, h(\mathsf{vk}, \mathsf{sk}), \mathsf{vk}, m^*)$	
If $\mathsf{Ver}(\mathsf{vk}, m^*, \sigma^*) = 1$ *then return* 1, *else return* 0.	

We notice that this notion of security is useful in some settings. For instance, it suffices to construct 2-round identification schemes w.r.t auxiliary inputs. In Section 4 we propose formal definitions and a simple construction of an identification scheme with security in the presence of auxiliary input leakage.

One way to enhance the security notion obtained by Definition 8 is to allow chosen message attacks, i.e., random message unforgeability under chosen message and auxiliary input attacks (RU-CMAA). In this game the adversary can pick the messages to be signed by itself but still need to forge a signature on a random message; see Section 3.2 for further discussion.

2.5 Classes of Auxiliary Input Functions

Clearly, the above notions of security require to specify the set of admissible functions $\mathcal{H}$. In the public key setting one can define two different types of classes of leakage functions. In the first class, we require that given the leakage $h(\mathsf{sk}, \mathsf{vk})$ it is computationally hard to compute sk, while in the latter we require hardness of computing sk when additionally given the public key vk. We follow the work of Dodis et al. [DGK$^+$10] to formally define this difference. Let in the following $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^k)$ be generated randomly.

- Let $\mathcal{H}_{\mathsf{ow}}(\ell(k))$ be the class of polynomial-time computable functions $h : \{0,1\}^{|\mathsf{sk}|+|\mathsf{vk}|} \to \{0,1\}^*$ such that given $h(\mathsf{sk}, \mathsf{vk})$, no PPT adversary can find sk with probability greater than $\ell(k) \geq 2^{-k}$, i.e., for any PPT adversary $\mathcal{A}$: $\Pr_{(\mathsf{sk},\mathsf{vk})\leftarrow\mathsf{Gen}(1^k)}[\mathcal{A}(h(\mathsf{sk}, \mathsf{vk})) = \mathsf{sk}] < \ell(k)$.
- Let $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$ be the class of polynomial-time computable functions $h : \{0,1\}^{|\mathsf{sk}|+|\mathsf{vk}|} \to \{0,1\}^*$ such that given $(\mathsf{vk}, h(\mathsf{sk}, \mathsf{vk}))$, no PPT adversary can find sk with probability greater than $\ell(k) \geq 2^{-k}$, i.e., for any PPT adversary $\mathcal{A}$: $\Pr_{(\mathsf{sk},\mathsf{vk})\leftarrow\mathsf{Gen}(1^k)}[\mathcal{A}(\mathsf{vk}, h(\mathsf{sk}, \mathsf{vk})) = \mathsf{sk}] < \ell(k)$.

Security with respect to auxiliary input gets stronger if $\ell(k)$ is larger. Our goal is typically to make $\ell(k)$ as close as possible to $\mathsf{negl}(k)$. If a scheme is EU-CMAA for $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$ according to Definition 7, we say for short that it is $\ell(k)$-EU-CMAA. Similarly, if a scheme is RU-RMAA for $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$, then we say that it is an $\ell(k)$-RU-RMAA signature scheme. If the class of admissible leakage functions is $\mathcal{H}_{\mathsf{ow}}(\ell(k))$, we will mention it explicitly.

As outlined in the introduction, we typically prove security with respect to the class $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$. The stronger security notion where hardness is required to hold *only* given the leakage, i.e., for the class of admissible functions $\mathcal{H}_{\mathsf{ow}}(\ell(k))$, can be achieved by a relation between $\mathcal{H}_{\mathsf{ow}}(\cdot)$ and $\mathcal{H}_{\mathsf{vkow}}(\cdot)$ proven by Dodis et al. [DGK$^+$10].

Lemma 1 ([DGK$^+$10]). *If $|\mathsf{vk}| = t(k)$ then for any $\ell(k)$, we have*

1. $\mathcal{H}_{\mathsf{vkow}}(\ell(k)) \subseteq \mathcal{H}_{\mathsf{ow}}(\ell(k))$
2. $\mathcal{H}_{\mathsf{ow}}(2^{-t(k)}\ell(k)) \subseteq \mathcal{H}_{\mathsf{vkow}}(\ell(k))$

The first point of Lemma 1 says that if no PPT adversary finds sk given $(\mathsf{vk},\ h(\mathsf{sk}, \mathsf{vk}))$ with probability $\ell(k)$ or better, then no PPT adversary finds sk given only $h(\mathsf{sk}, \mathsf{vk})$ with probability $\ell(k)$ or better. Clearly this is the case since knowing vk cannot make it harder to guess sk. The second point states that if no PPT adversary finds sk given $h(\mathsf{sk}, \mathsf{vk})$ with probability $2^{-t(k)}\ell(k)$ or better, then any PPT adversary has advantage at most $\ell(k)$ in guessing sk when given additionally vk. To see this consider a PPT adversary $\mathcal{A}$ that finds sk given $(\mathsf{vk}, h(\mathsf{sk}, \mathsf{vk}))$ with probability $\ell'(k) \geq \ell(k)$. $\mathcal{A}$ then implies a PPT adversary $\mathcal{B}$ that given $h(\mathsf{sk}, \mathsf{vk})$ simply tries to guess vk and uses it to run $\mathcal{A}$. Since $\mathcal{B}$ can guess vk with probability at least $2^{-t(k)}$, $\mathcal{B}$ has probability at least $2^{-t(k)}\ell'(k)$ of finding sk. Thus contradicting $h \in \mathcal{H}_{\mathsf{ow}}(2^{-t(k)}\ell(k))$.

3 Designing Signature Schemes with Auxiliary Input Security

3.1 A Warm-Up Construction

In order to illustrate the difficulties encountered in designing cryptographic primitives in the auxiliary input setting we present a warm-up construction of a signature scheme that may seem secure at first glance but, unfortunately, proving its security is impossible. Essentially, the problem arises due to the computational hardness of the leakage and does not occur in other leakage models, where given the leakage the secret key is still information theoretically hidden. For ease of understanding, in this warm-up construction we only aim for the simpler one-time security notion on random messages, where the adversary only views a single signature before it outputs its forgery on a random message. We consider two building blocks for the following scheme.

- A family of SPR hash functions H.
- A non-interactive zero-knowledge proof of knowledge system $\Pi = (\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$.

Informally, the signature scheme is built as follows. The signing key sk is a random element x in the domain of the hash function, whereas the verification key vk is $y = H(x)$. vk also contains a common reference string crs for Π. A signature on a message m is the bit $b = \langle m, \mathsf{sk} \rangle$ together with a non interactive proof with respect to crs proving that b was computed as the inner product of the preimage of y and the message m. More precisely, define the signature scheme $\Sigma = (\mathsf{Gen}_\Sigma, \mathsf{Sig}_\Sigma, \mathsf{Ver}_\Sigma)$ as follows:

Key Generation, $\mathsf{Gen}_\Sigma(1^k)$**:** Sample a SPR hash function H, a random element x in the domain of H and $\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^k)$. Output $\mathsf{sk} = x, \quad \mathsf{vk} = (H(x), \mathsf{crs})$.

Signing, $\mathsf{Sig}_\Sigma(\mathsf{sk}, m)$**:** Parse vk as $(H(\mathsf{sk}), \mathsf{crs})$. Compute $b = \langle m, \mathsf{sk} \rangle$. Use the crs to generate a non-interactive zero-knowledge proof of knowledge π, demonstrating that $b = \langle m, \mathsf{sk} \rangle$ and $H(\mathsf{sk}) = y$. Output $\sigma = (b, \pi)$.

Verifying, $\mathsf{Ver}_\Sigma(\mathsf{vk}, m, \sigma)$**:** Parse vk as $(H(\mathsf{sk}), \mathsf{crs})$ and σ as (b, π). Use crs to verify the proof π. Output 1 if the proof is verified correctly and 0 otherwise.

We continue with an attempt to prove security. Note first that by the properties of Π, the ability to generate a forgery (σ', m') reduces to the ability using the extraction trapdoor to either find a second preimage for the hash function or break the hardness assumption of the leakage function. As the difficulties arise in the reduction to the hardness of the leakage function, we focus in this outline on that part. Assume there is an adversary $\mathcal{A}$ attacking signature scheme Σ given auxiliary input leakage $h(\mathsf{sk}, \mathsf{vk})$ and (y, crs). Then, an attempt to construct $\mathcal{B}$ that breaks the hardness assumption of the leakage function by invoking $\mathcal{A}$ works as follows. $\mathcal{B}$ obtains (y, crs) and the leakage $h(\mathsf{sk}, \mathsf{vk})$ from its challenge oracle. It forwards them to $\mathcal{A}$ who will ask for signature query. Unfortunately, at that point we are not able to answer this query as we cannot simulate a proof without knowing the witness or the trapdoor.

An alternative approach may be to directly prove security with respect to the leakage class $\mathcal{H}_{\mathsf{ow}}(\ell(k))$ and let $\mathcal{B}$ sample the crs herself using the NIZK simulator to know a trapdoor. Unfortunately, also this approach is deemed to fail as in this case there is no way to learn a $y = H(\mathsf{sk})$ that is consistent with the leakage. Moreover this results into several difficulties in defining the set of admissible leakage functions as they must be different now for $\mathcal{A}$ and $\mathcal{B}$. This can be illustrated as follows. Suppose that the CRS is a public key for an encryption scheme and the trapdoor is the corresponding secret key. As $\mathcal{A}$ only knows the CRS but not the trapdoor a leakage function h that outputs an encryption of $\mathsf{sk} = x$ is admissible. On the other hand, however, for $\mathcal{B}$ who knows the trapdoor (hence the secret key of the encryption scheme) such leakage cannot be admissible. This shows that we need to consider different approaches when analyzing the security of digital signature schemes in the presence of auxiliary input. In what follows, we demonstrate two different approaches for such constructions, obtaining two different notions of security.

3.2 A RU-RMAA Signature Scheme

In this section we present our construction of a RU-RMAA signature scheme as defined in Definition 8. For this scheme we assume the following building blocks:

1. A family H of second preimage resistant hash functions with input length k_1 and key sampling algorithm Gen_H.
2. A NIZKPoK system $\Pi = (\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$ (cf. Definition 6) for proving knowledge of a secret value x so that $y = H_s(x)$ given s and y. We further require that the common reference strings of Π are uniformly random strings of some length $p(k)$ for security parameter k and some polynomial $p(\cdot)$. Denote the message space $\mathcal{M}$ by $\{0,1\}^{p(k)}$.

The main idea for the scheme is inspired by the work by Malkin et al. [MTVY11] where we view each message m as a common reference string for the proof system Π. Since m is uniformly generated, we are guaranteed that the crs is generated correctly and knowledge soundness holds. Intuitively since each new message induces a new crs, each proof is given with respect to an independent crs. This implies that in the security proof the simulator (playing the role of the signer) *can* use the trapdoor of the crs that corresponds to the challenge message m^*.

We formally define our scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ as follows.

Key Generation, $\mathsf{Gen}(1^k)$**:** Sample $s \leftarrow \mathsf{Gen}_H(1^k)$. Choose $x \leftarrow \{0,1\}^{k_1}$ and compute $y = H_s(x)$. Output $\mathsf{sk} = (x, s)$ and $\mathsf{vk} = (y, s)$.
Signing, $\mathsf{Sig}(\mathsf{sk}, m)$**:** To sign $m \leftarrow \mathcal{M}$, let $\mathsf{crs} = m$ and sample the signature $\sigma \leftarrow \mathsf{P}(\mathsf{crs}, \mathsf{vk}, \mathsf{sk})$ as a proof of knowledge of x such that $y = H_s(x)$.
Verifying, $\mathsf{Ver}(\mathsf{vk}, m, \sigma)$**:** To verify σ on $m = \mathsf{crs}$, output $\mathsf{V}(\mathsf{crs}, \mathsf{vk}, \sigma)$.

Theorem 1. *Assume that H is a second preimage resistant family of hash functions and $\Pi = (\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$ is a* NIZKPoK *system. Then $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is a $\mathsf{negl}(k)$-RU-RMAA signature scheme.*

The intuition of the proof is that if one can efficiently forge a signature on a random m^* after getting signatures on random messages m, then one can also efficiently compute x, contradicting the fact that that the leakage is hard to efficiently invert. During the simulated attack the signatures on random messages m are simulated by sampling $m = \mathsf{crs}$, where crs is sampled along with the simulation trapdoor. At the end one samples $m^* = \mathsf{crs}$, where crs is sampled along with the extraction trapdoor. Upon getting a forgery on m^*, we can extract x using the extraction trapdoor.

In the standard setting, a simple modification using Chameleon hash functions [KR00] enables to achieve a stronger notion of security. Recall first that Chameleon hash functions are collision resistance hash functions such that given a trapdoor one can efficiently find collisions for every given preimage and its hashed value. Thereby, instead of signing random messages the scheme can be modified so that the signer signs the hashed value of the message. This achieves chosen message attacks security so that the adversary picks the messages to be signed during the security game, yet the challenge is still picked at random. Nevertheless, when introducing hard-to-invert leakage into the system this approach does not enable to obtain security against polynomially hard-to-invert leakage, because we run into the same problem specified in Section 3.1. Moreover, in Section 3.3 we show how to obtain the strongest security notion of existential unforgeability under chosen message and auxiliary input attacks.

Proof. Let $\mathsf{Exp}_{\Sigma,\mathcal{A},h}$ be as defined in Definition 8 for PPT adversary $\mathcal{A}$ and leakage function $h \in \mathcal{H}_{\mathsf{vkow}}(\mathsf{negl}(k))$. Furthermore let W be the event that $\mathcal{A}$ wins the game. We show that $\Pr[W]$ is negligible. Denote this probability by p_0. Consider the following modification to $\mathsf{Exp}_{\Sigma,\mathcal{A},h}(k)$.

1. Generate $(\mathsf{vk}, \mathsf{sk})$ as in $\mathsf{Exp}_{\Sigma,\mathcal{A},h}(k)$.
2. Instead of sampling the challenge m^* as $m^* \leftarrow M$ sample $(m', \mathsf{td}_e) \leftarrow E_1(1^k)$ and let $m^* = m'$, where $E = (E_1, E_2)$ is the extractor for Π implied by Definition 4.
3. Give input to $\mathcal{A}$ as in $\mathsf{Exp}_{\Sigma,\mathcal{A},h}(k)$.

4. To answer the oracle queries of $\mathcal{A}$, sample $(m', \mathsf{td}_s) \leftarrow S_1(1^k)$, let $m = m'$ and return the signature $(m, S_2(m, \mathsf{vk}, \mathsf{td}_s))$, where $S = (S_1, S_2)$ is the simulator for Π implied by Definition 6.
5. Receive a forgery σ^* from $\mathcal{A}$ as in $\mathsf{Exp}_{\Sigma,\mathcal{A},h}(k)$.
6. Output as in $\mathsf{Exp}_{\Sigma,\mathcal{A},h}(k)$.

Let p_1 be the probability that the modified experiment above outputs 1. Also consider $x' = E_2(m^*, \mathsf{vk}, \mathsf{td}_e, \sigma^*)$. I.e. x' is a signing key extracted from $\mathcal{A}$'s forgery. By Definition 6 we have that distributions of messages and signatures in the modified experiment are indistinguishable from the distributions in the original experiment $\mathsf{Exp}_{\Sigma,\mathcal{A},h}(k)$. Thus it follows that p_1 is negligibly close to p_0. Let p_2 be the probability that $H_s(x') = y$. By the knowledge soundness of Π it follows that p_2 is negligibly close to p_0.

Note then that, since S and E are both PPT algorithms, the modified experiment describes a PPT algorithm which computes x' where with probability p_2 it holds that $y = H_s(x')$. Let p_3 be the probability that $y = H_s(x')$ and $x' \neq x$ and let p_4 be the probability that $x' = x$. Note that $p_2 = p_3 + p_4$.

The Event X Consider the PPT algorithm $\mathcal{B}$ that given vk and leakage $h(\mathsf{sk}, \mathsf{vk})$, where $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^k)$, runs steps 2-5 of the modified experiment above and outputs $x^* = E_2(m^*, \mathsf{vk}, \mathsf{td}_e, \sigma^*)$. Denote by X the event in which $\mathcal{B}$ outputs $x^* = x$. Since $(\mathsf{vk}, \mathsf{sk})$ is generated as in $\mathsf{Exp}_{\Sigma,\mathcal{A},h}(k)$ $\Pr[X] \geq p_4$. Thus by definition of $\mathcal{H}_{\mathsf{vkow}}(\mathsf{negl}(k))$, p_4 is negligible.

The Event C On the other hand, consider the PPT algorithm $\mathcal{B}$ that is given s, x and $y = H_s(x)$. $\mathcal{B}$ lets $\mathsf{vk} = (y, s)$ and runs steps 2-5 of the modified experiment above (notice that $\mathcal{B}$ is given x, so it can compute the leakage h) and outputs $x^* = E_2(m^*, \mathsf{vk}, \mathsf{td}_e, \sigma^*)$. Denote by C the event in which $\mathcal{B}$ outputs $x^* \neq x$ so that $H_s(x^*) = H_s(x)$. Notice again that $((H_s, y), x)$ are generated as in $\mathsf{Exp}_{\Sigma,\mathcal{A},h}(k)$ and therefore $\Pr[C] \geq p_3$. Thus by the second preimage resistance hardness of the family H, p_3 is negligible.

This implies that p_3 and p_4 are negligible and so is $p_2 = p_3 + p_4$. Since p_0 is negligibly close to p_2, p_0 must also be negligible. By definition $p_0 = \Pr[\mathsf{Exp}_{\Sigma,\mathcal{A},h}(k) = 1]$ and so by Definition 8, Σ is a $\mathsf{negl}(k)$-RU-RMAA signature scheme. □

Notice that in the above we assume that the CRS of the NIZKPoK Π is a uniformly random bit string. As an example of a NIZKPoK with this property we can use the construction of [RS91]. In their construction the CRS is a pair (ek, r) where r is a random string and ek is an encryption key for some semantically secure public-key encryption scheme. Thus, we can use the construction of [RS91] with a public-key encryption scheme where uniformly random bit strings can act as public-keys, like Regev's LWE scheme [Reg05].

3.3 A EU-CMAA Signature Scheme

In this section we build a EU-CMAA signature scheme. We use k to denote the security parameter. We need the following tools:

1. A family of second pre-image resistant hash functions H with key sampling algorithm Gen_H, where the input length can be set to be any $k_4 = \text{poly}(k)$ and where the length of the randomness used by $s \leftarrow \mathsf{Gen}_H(1^k)$ is some $l_1 = \text{poly}(k)$ independent of k_4 and where the length of an output $y = H_s(x)$ is some $l_4 = \text{poly}(k)$ independent of k_4. I.e., it is possible to increase the input length of H_s without increasing the randomness used to generate s or the output length.

2. An IND-WLCCA secure labeled public-key encryption scheme $\Gamma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with perfect decryption (cf. Definition 2), where the length of dk is some $l_2 = \text{poly}(k)$ independent of the length of the messages that Γ can encrypt.

3. A reusable-CRS NIZK system $\Pi = (\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$ (cf. Definition Def. 5), where the length of the simulation trapdoor td_s at security level k is some $l_3 = \mathrm{poly}(k)$ independent of the size of the proofs that the NIZK can handle.

The IND-WLCCA secure encryption scheme might be replaced by a IND-CPA secure scheme, but at the price of then instead using a simulation sound NIZK: We expect a general proof via true simulation extractability to work along the lines of [DHLAW10b]. We chose the above tools as they lean themeselves nicely towards our concrete instantiation.

The reason why we use IND-WLCCA is that our signature scheme requires to encrypt its secret key that is much longer than the decryption key. For that we need to break the secret key into blocks and encrypt each block separately under the *same* label (looking ahead, the label would be the signed message). Note that labeled public-key encryption schemes for arbitrary length massages is not implied by LCCA secure scheme for fixed length messages. This is because the adversary can change the order of the ciphertexts within a specific set of ciphertexts and ask for a decryption. We therefore work with the weaker notion that is sufficient for our purposes to design secure signature schemes, and is easier to instantiate as demonstrated in Section 5.

Our scheme Σ works as follows:

Key Generation, $\mathsf{Gen}(1^k)$**:** Sample $s \leftarrow \mathsf{Gen}_H(1^k)$ and $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^k)$. Furthermore, sample $(\mathsf{crs}, \mathsf{td}_s) \leftarrow S_1(1^k)$ and $x \leftarrow \{0,1\}^{k_4}$, where $S = (S_1, S_2)$ is the simulator for Π.[2] Compute $y = H_s(x)$. Set $(\mathsf{sk}, \mathsf{vk}) = (x, (y, s, \mathsf{ek}, \mathsf{crs}))$.

Signing, $\mathsf{Sig}(\mathsf{sk}, m)$**:** Compute $C = \mathsf{Enc}^m(\mathsf{ek}, x)$. Using crs and Π, generate a NIZK proof π proving that $\exists x (C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. Output $\sigma = (C, \pi)$.

Verifying, $\mathsf{Ver}(\mathsf{vk}, m, \sigma)$**:** Parse σ as C, π. Use crs and V to verify the NIZK proof π. Output 1 if the proof verifies correctly and 0 otherwise.

As explained in [DHLAW10b], a NIZK proof system together with a CCA-secure encryption scheme are a specific instantiation of *true-simulation extractable (tSE)*. An alternative instantiation would be to compose a simulation-sound NIZK with a CPA-secure encryption scheme. This approach was used in [KV09]. We note that our proof follows similarly for this instantiation as well.

Theorem 2. *If H, $\Gamma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ and $\Pi = (\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$ have the properties listed above, then Σ is 2^{-k_5}-EU-CMAA where $k_5 = k + l_2 + l_3$ and where*

- *k is the security parameter of Σ,*
- *l_1 is the length of the randomness used to sample s at security parameter k_1 for H,*
- *l_2 is the length of the decryption key* dk *at security parameter k_2 for Γ,*
- *l_3 is the length of the simulation trapdoor* td_s *at security parameter k_3 for Π,*

If we consider the class $\mathcal{H}_{\mathsf{ow}}(\ell(k))$, then our scheme is 2^{-k_6}-EU-CMAA where $k_6 = k + l_1 + l_2 + l_3 + l_4$ and where l_4 is the length of $y = H_s(x)$ at security parameter k_1 for H.
Specifically, the best success against Σ in the forging game with 2^{-k_5}-hard leakage by a PPT adversary $\mathcal{A}$ is $2^{-k} + \sum_{i=0}^{3} \varepsilon_i + u\varepsilon_4$, where u is a polynomial and

- *ε_0 and ε_3 are the advantages of some PPT adversaries in the ZK game against Π at security parameter k_3,*
- *ε_1 is the success probability of some PPT adversary in the soundness game against Π at security parameter k_3,*
- *ε_2 is the probability that some PPT adversary wins the second pre-image game against H on security parameter k_1 and $x \leftarrow \{0,1\}^{k_4}$,*

[2] It is deliberate that we use a simulated CRS as part of the public key. This makes the set of admissible leakage functions defined relative to a simulated CRS, which we use in the proof. The scheme might be secure for a normal CRS too, but the proof would be more complicated.

– ε_4 *is the advantage of some PPT adversary in the IND-WLCCA game against Γ at security parameter k_2.*

The intuition behind the proof of security is that a forged signature contains an encryption of the secret key x, so forging leads to extracting x using dk, giving a reduction to the assumption that it is hard to compute x given the leakage. In doing this reduction the signing oracle is simulated by encrypting 0^{k_4} and simulating the proofs using the simulation trapdoor td_s. This will clearly still lead to an extraction of x, using reusable-CRS ZK and IND-WLCCA. The only hurdle is that given $(\mathsf{vk}, h(\mathsf{sk}, \mathsf{vk}))$, we do not know dk or td_s. We can, however, guess these with probability 2^{-l_2} respectively 2^{-l_3}. This is why we only get security $k_{\mathrm{W}} = k + l_2 + l_3$. When we prove security for $\mathcal{H}_{\mathsf{ow}}(\ell(k))$ the reduction is not given vk either, so we additionally have to guess s and y, leading to $k_{\mathrm{S}} = k + l_1 + l_2 + l_3 + l_4$.

If we set $k_4 = k+l_2+l_3+l_4+L$, then the min-entropy of x given $y = H_s(x)$ is $k+l_2+l_3+L$, so leaking L bits would be an admissible leakage in the $2^{-k_{\mathrm{W}}}$ security game. Since, by assumption on our primitives, l_2 and l_3 and l_4 does not grow with k_4, it follows that we can set L to be any polynomial and be secure against leaking any fraction $(1 - k^{-O(1)})$ of the secret key.

Proof. Let $\mathcal{A}$ be any PPT adversary attacking our scheme. Let W be the event that $\mathcal{A}$ wins the game. We derive a bound on $\Pr[W]$. We start by writing out the forging game for our particular scheme:

Key generation: Sample $s \leftarrow \mathsf{Gen}_H$ and $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^{k_2})$. Also, sample $(\mathsf{crs}, \mathsf{td}_s) \leftarrow S_1(1^{k_3})$ and $x \leftarrow \{0,1\}^{k_4}$. Compute $y = H_s(x)$. Set $(\mathsf{sk}, \mathsf{vk}) = (x, (y, s, \mathsf{ek}, \mathsf{crs}))$.
Leakage: Give $\mathsf{vk} = (y, s, \mathsf{ek}, \mathsf{crs})$ to $\mathcal{A}$ along with $h(\mathsf{sk}, \mathsf{vk})$.
Signing Oracle:
1. Get a message m from $\mathcal{A}$.
2. Compute the ciphertext $C = \mathsf{Enc}^m(\mathsf{ek}, x)$. Using crs, generate a NIZK proof π proving that $\exists x (C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. Give $\sigma = (C, \pi)$ to $\mathcal{A}$.

Calling the Game: Get $(m^*, (C^*, \pi^*))$ from $\mathcal{A}$. The adversary wins iff π^* is an acceptable proof that $\exists x^* (C^* = \mathsf{Enc}^{m^*}(\mathsf{ek}, x^*) \wedge y = H_s(x^*))$, and m^* was not queried. Output 1 if $\mathcal{A}$ wins the game and output 0 otherwise.

Let $\Game$ denote the above game. Clearly

$$\Pr[W] = \Pr[\Game = 1] \ .$$

Let $\Game_0$ denote the game which proceeds as $\Game$, with the following change:

Key generation 0: Sample $s \leftarrow \mathsf{Gen}_H$ and $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^{k_2})$. *Furthermore, sample* $\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^{k_3})$ *and* $x \leftarrow \{0,1\}^{k_4}$.[3] Compute $y = H_s(x)$. Set $(\mathsf{sk}, \mathsf{vk}) = (x, (y, s, \mathsf{ek}, \mathsf{crs}))$.

The only change from $\Game$ is that we run with $\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^{k_3})$ instead of a crs sampled along with a simulation trapdoor td_s. Note, however, that a PPT adversary which can distinguish these two distributions with advantage ε_0 can win the ZK game against $(\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$ with advantage ε_0, using 0 queries to the oracle which gives either real proofs or simulated proofs. A simple reduction thus shows that,

$$\Pr[\Game = 1] - \Pr[\Game_0 = 1] \leq \varepsilon_0 \ ,$$

where ε_0 is the advantage of some PPT adversary against the ZK game against $(\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$ at security parameter k_3.

Let $\Game_1$ denote the game which proceeds as $\Game_0$, with the following change:

[3] It might appear odd that we use a simulated CRS in the scheme and then change to real CRS is the first thing in the proof. Recall, however, that the scheme uses a simulated CRS to force the set of admissible leakage functions to be defined relative to a simulated CRS. In the next step of the proof, however, we need a real CRS, to appeal to the soundness of the NIZK. When this is done, we will change the flavour of the CRS back again.

Calling the Game 1: Get $(m^*, (C^*, \pi^*))$ from $\mathcal{A}$. The adversary wins iff π^* is an acceptable proof that $\exists x^*(C^* = \mathsf{Enc}^{m^*}(\mathsf{ek}, x^*) \wedge y = H_s(x^*))$, and m^* was not queried. *If $\mathcal{A}$ wins the game, then compute $x^* = \mathsf{Dec}^{m^*}(\mathsf{dk}, C^*)$. Output* 1 *iff $\mathcal{A}$ wins the game and $(x^* = x \vee y = H_s(x^*))$.*

The only change is that we only output 1 if the extra condition $(x^* = x \vee y = H_s(x^*))$ holds. Note, however, that if $(x^* = x \vee y = H_s(x^*))$ is false, then in particular $y \neq H_s(x^*)$. By the perfect decryption, this implies that $\nexists x^*(C^* = \mathsf{Enc}^{m^*}(\mathsf{ek}, x^*) \wedge y = H_s(x^*))$. So, in $\eth_1$ we compute a proof π^* for a false statement with probability at least $\Pr[\eth_0 = 1] - \Pr[\eth_1 = 1]$. Since we can take a $\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^{k_3})$ as input and in PPT run $\eth_1$ with that specific crs in vk, we have that

$$\Pr[\eth_0 = 1] - \Pr[\eth_1 = 1] = \varepsilon_1 ,$$

where ε_1 is the success probability of some PPT adversary in the soundness game against $\mathit{\Pi}$ at security parameter k_3.

Let $\eth_2$ denote the game which proceeds as $\eth_1$, with the following change:

Calling the Game 2: Get $(m^*, (C^*, \pi^*))$ from $\mathcal{A}$. The adversary wins iff π^* is an acceptable proof that $\exists x^*(C^* = \mathsf{Enc}^{m^*}(\mathsf{ek}, x^*) \wedge y = H_s(x^*))$, and m^* was not queried. If $\mathcal{A}$ wins the game, then compute $x^* = \mathsf{Dec}^{m^*}(\mathsf{dk}, C^*)$. *Output* 1 *iff $\mathcal{A}$ wins the game and $x^* = x$.*

Note that if we run $\eth_2$, then with probability at least $\Pr[\eth_1 = 1] - \Pr[\eth_2 = 1]$ we have that $x^* \neq x$ and $y = H_s(x^*)$. Since we can take a random s and a random x as input and in PPT run $\eth_2$ with that specific s as key for H_s and that specific x as signing key sk, it follows that

$$\Pr[\eth_1 = 1] - \Pr[\eth_2 = 1] \leq \varepsilon_2 ,$$

where ε_2 is the probability that some PPT adversary wins the second pre-image game against H on security parameter k_1 and $x \leftarrow \{0,1\}^{k_4}$.

Let $\eth_3$ be the game which proceeds as follows:

Key generation 3: Sample $s \leftarrow \mathsf{Gen}_H$ and $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^{k_2})$. Furthermore, sample $(\mathsf{crs}, \mathsf{td}_s) \leftarrow S_1(1^{k_3})$ and $x \leftarrow \{0,1\}^{k_4}$. Compute $y = H_s(x)$. Set $(\mathsf{sk}, \mathsf{vk}) = (x, (y, s, \mathsf{ek}, \mathsf{crs}))$.
Leakage 3: Give $\mathsf{vk} = (y, s, \mathsf{ek}, \mathsf{crs})$ to $\mathcal{A}$ along with $h(\mathsf{sk}, \mathsf{vk})$.
Signing Oracle 3:

1. Get a message m from $\mathcal{A}$.
2. Compute $C = \mathsf{Enc}^m(\mathsf{ek}, x)$. Using td_s, generate a simulated NIZK π proving that $\exists x(C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. Give $\sigma = (C, \pi)$ to $\mathcal{A}$.

Calling the Game 3: Get $(m^*, (C^*, \pi^*))$ from $\mathcal{A}$. The adversary wins iff π^* is an acceptable proof that $\exists x^*(C^* = \mathsf{Enc}^{m^*}(\mathsf{ek}, x^*) \wedge y = H_s(x^*))$, and m^* was not queried. If $\mathcal{A}$ wins the game, then compute $x^* = \mathsf{Dec}^{m^*}(\mathsf{dk}, C^*)$. Output 1 iff $\mathcal{A}$ wins the game and $x^* = x$.

The only difference between $\eth_2$ and $\eth_3$ is whether we give real proofs or simulated proofs. Since we can take crs as input plus access to an oracle $\mathcal{O}$ which produces either real proofs under crs or simulated proofs under crs and in PPT produce the output of $\eth_2$ respectively $\eth_3$, it follows that

$$\Pr[\eth_2 = 1] - \Pr[\eth_3 = 1] \leq \varepsilon_3 ,$$

where ε_3 is the advantage of some PPT adversary in the ZK game against $\mathit{\Pi}$ on security parameter k_3.

Let $\eth_4$ be $\eth_3$ with the following change:

Signing Oracle 4:

1. Get a message m from $\mathcal{A}$.
2. *Compute* $C = \mathsf{Enc}^m(\mathsf{ek}, 0^{k_4})$. Using td_s, generate a simulated NIZK π proving that $\exists x(C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. Give $\sigma = (C, \pi)$ to $\mathcal{A}$.

Consider the following adversary $\mathcal{B}^h$ for the labeled IND-WLCCA game against Γ, where h is a natural number.

Key generation 3-4: Sample $s \leftarrow \mathsf{Gen}_H$ and get ek as input from the IND-WLCCA game. Furthermore, sample crs along with a simulation trapdoor td_s and $x \leftarrow \{0,1\}^{k_4}$. Compute $y = H_s(x)$. Set $(\mathsf{sk}, \mathsf{vk}) = (x, (y, s, \mathsf{ek}, \mathsf{crs}))$.

Leakage 3-4: Give $\mathsf{vk} = (y, s, \mathsf{ek}, \mathsf{crs})$ to $\mathcal{A}$ along with $h(\mathsf{sk}, \mathsf{vk})$.

Signing Oracle 3-4: In the i'th signing request, proceed as follows: If $i < h$, then sign as in $\Game_3$. If $i > h$, then sign as in $\Game_4$. If $i = h$, then sign as follows:

1. Get a message m from $\mathcal{A}$.
2. Output $(m, x, 0^{k_4})$ to the encryption oracle and get back C. Using td_s, generate a simulated NIZK π proving that $\exists x(C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. Give $\sigma = (C, \pi)$ to $\mathcal{A}$.

Calling the Game 3-4: Get $(m^*, (C^*, \pi^*))$ from $\mathcal{A}$. The adversary wins iff π^* is an acceptable proof that $\exists x^*(C^* = \mathsf{Enc}^{m^*}(\mathsf{ek}, x^*) \wedge y = H_s(x^*))$, and m^* was not queried. If $\mathcal{A}$ wins the game, then query the decryption oracle on (m^*, C^*) and get back x^*. Output 1 iff $\mathcal{A}$ wins the game and $x^* = x$.

This is an admissible IND-WLCCA adversary, as $|x| = k_4$ and because the label in the (m^*, C^*) submitted to the decryption oracle is different from all the labels in the (m, x) submitted to the encryption oracle, as a condition for $\mathcal{A}$ winning is that m^* was not queried to the signing oracle. Let u be a polynomial upper bound on the number of signing queries of $\mathcal{A}$. By construction we have that the sum of the advantages of adversaries $\mathcal{B}^1, \ldots, \mathcal{B}^u$ is an upper bound on $|\Pr[\Game_3 = 1] - \Pr[\Game_4 = 1]|$. It follows that

$$\Pr[\Game_3 = 1] - \Pr[\Game_4 = 1] \leq u\varepsilon_4 \ ,$$

where ε_4 is the advantage of some PPT adversary in the IND-WLCCA game against Γ at security parameter k_2.

Consider then the following algorithm $\mathcal{B}_5(x)$, which takes $x \in \{0,1\}^{k_4}$ as input.

Key generation 5: Sample $s \leftarrow \mathsf{Gen}_H$ and $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^{k_2})$. Furthermore, sample $(\mathsf{crs}, \mathsf{td}_s) \leftarrow S_1(1^{k_3})$. Get $x \in \{0,1\}^{k_4}$ as input. Compute $y = H_s(x)$. Set $(\mathsf{sk}, \mathsf{vk}) = (x, (y, s, \mathsf{ek}, \mathsf{crs}))$.

Leakage 5: Give $\mathsf{vk} = (y, s, \mathsf{ek}, \mathsf{crs})$ to $\mathcal{A}$ along with $h(\mathsf{sk}, \mathsf{vk})$.

Signing Oracle 5:

1. Get a message m from $\mathcal{A}$.
2. Compute $C = \mathsf{Enc}^m(\mathsf{ek}, 0^{k_4})$. Using td_s, generate a simulated NIZK π proving that $\exists x(C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. Give $\sigma = (C, \pi)$ to $\mathcal{A}$.

Calling the Game 5: Get $(m^*, (C^*, \pi^*))$ from $\mathcal{A}$. Output $x^* = \mathsf{Dec}^{m^*}(\mathsf{dk}, C^*)$.

Clearly

$$\Pr_{x \leftarrow \{0,1\}^{k_4}}[\mathcal{B}_5(x) = x|] \geq \Pr[\Game_4 = 1] \ .$$

Consider then the following algorithm $\mathcal{B}_6(\mathsf{vk}, a)$, which takes a verification key for Σ and some auxiliary input $a \in \{0,1\}^*$ as input.

Key generation 6: Get the input (vk, a) and parse vk as $\mathsf{vk} = (y, s, \mathsf{ek}, \mathsf{crs})$. Sample $\mathsf{dk}' \leftarrow \{0,1\}^{l_2}$ and $\mathsf{td}'_s \leftarrow \{0,1\}^{l_3}$.

Leakage 6: Give $\mathsf{vk} = (y, s, \mathsf{ek}, \mathsf{crs})$ to $\mathcal{A}$ along with a.

Signing Oracle 6:

1. Get a message m from $\mathcal{A}$.
2. Compute $C = \mathsf{Enc}^m(\mathsf{ek}, 0^{k_4})$. Using td'_s, generate a simulated NIZK proof π proving that $\exists x(C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. Give $\sigma = (C, \pi)$ to $\mathcal{A}$.

Calling the Game 6: Get $(m^*, (C^*, \pi^*))$ from $\mathcal{A}$. Output $x^* = \mathsf{Dec}^{m^*}(\mathsf{dk}', C^*)$.

Let V denote the distribution on $(\mathsf{vk}, \mathsf{sk}, \mathsf{dk}, \mathsf{td}_s)$ produced by sampling as in Gen. I.e. V is produced as follows: Sample $s \leftarrow \mathsf{Gen}_H(1^{k_1})$ and $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^{k_2})$. Furthermore, sample $(\mathsf{crs}, \mathsf{td}_s) \leftarrow S_1(1^{k_3})$ and $x \leftarrow \{0,1\}^{k_4}$. Compute $y = H_s(x)$. Set $(\mathsf{sk}, \mathsf{vk}) = (x, (y, s, \mathsf{ek}, \mathsf{crs}))$. Output $(\mathsf{vk}, \mathsf{sk}, \mathsf{dk}, \mathsf{td}_s)$. By construction

$$\Pr_{(\mathsf{vk},\mathsf{sk},\mathsf{dk},\mathsf{td}_s)\leftarrow V}[\mathcal{B}_6(\mathsf{vk}, h(\mathsf{sk}, \mathsf{vk})) = \mathsf{sk} | \mathsf{dk}' = \mathsf{dk} \wedge \mathsf{td}'_s = \mathsf{td}_s] = \Pr_{x \leftarrow \{0,1\}^{k_4}}[\mathcal{B}_5(x) = x] ,$$

where dk' and td'_s are the values sampled by $\mathcal{B}_6$. Since $\Pr[\mathsf{dk}' = \mathsf{dk} \wedge \mathsf{td}'_s = \mathsf{td}_s] = 2^{-l_2 - l_3}$ it follows that

$$\Pr_{(\mathsf{vk},\mathsf{sk},\mathsf{dk},\mathsf{td}_s)\leftarrow V}[\mathcal{B}_6(\mathsf{vk}, h(\mathsf{sk}, \mathsf{vk})) = \mathsf{sk}] \geq 2^{-l_2 - l_3} \Pr_{x \leftarrow \{0,1\}^{k_4}}[\mathcal{B}_5(x) = x] ,$$

or

$$\Pr_{x \leftarrow \{0,1\}^{k_4}}[\mathcal{B}_5(x) = x] \leq 2^{l_2 + l_3} \Pr_{(\mathsf{vk},\mathsf{sk},\mathsf{dk},\mathsf{td}_s)\leftarrow V}[\mathcal{B}_6(\mathsf{vk}, h(\mathsf{sk}, \mathsf{vk})) = \mathsf{sk}] .$$

The distribution on $(\mathsf{sk}, \mathsf{vk})$ induced by V is identical to that induced by the key generation of Σ, and $\mathcal{B}_6$ is PPT. Consequently, when we are proving security we can assume that

$$\Pr_{(\mathsf{vk},\mathsf{sk},\mathsf{dk},\mathsf{td}_s)\leftarrow V}[\mathcal{B}_6(\mathsf{vk}, h(\mathsf{sk}, \mathsf{vk})) = \mathsf{sk}] \leq 2^{-k_5} .$$

Combining our inequalities so far, we get that

$$\Pr[W] \leq 2^{l_2 + l_3 - k_5} + \sum_{i=0}^{3} \varepsilon_i + u\varepsilon_4 = 2^{-k} + \sum_{i=0}^{3} \varepsilon_i + u\varepsilon_4 .$$

Now, if we want prove security for the class $\mathcal{H}_{\mathsf{ow}}(\ell(k))$, we consider the following algorithm $\mathcal{B}_7(a)$ which takes some auxiliary input $a \in \{0,1\}^*$.

Key generation 7: Get the input a. Sample $\mathsf{dk}' \leftarrow \{0,1\}^{l_2}$ and $\mathsf{td}'_s \leftarrow \{0,1\}^{l_3}$, and let ek' be the public key corresponding to dk' and let crs' be the CRS corresponding to td'_s[4]. Sample $r \leftarrow \{0,1\}^{l_1}$ and let $s = \mathsf{Gen}_H(1^{k_1}; r)$, and sample $y' \leftarrow \{0,1\}^{l_4}$. Let $\mathsf{vk}' = (y', s', \mathsf{ek}', \mathsf{crs}')$.
Leakage 7: Give vk' to $\mathcal{A}$ along with a.
Signing Oracle 7:

1. Get a message m from $\mathcal{A}$.
2. Compute $C = \mathsf{Enc}^m(\mathsf{ek}', 0^{k_4})$. Using td'_s, generate a simulated NIZK proof π proving that $\exists x (C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. Give $\sigma = (C, \pi)$ to $\mathcal{A}$.

Calling the Game 7: Get $(m^*, (C^*, \pi^*))$ from $\mathcal{A}$. Output $x^* = \mathsf{Dec}^{m^*}(\mathsf{dk}', C^*)$.

Let V' be the same distribution as V except that it also outputs s and y, i.e., it outputs $(\mathsf{vk}, \mathsf{sk}, \mathsf{dk}, \mathsf{td}_s, s, y)$. By construction

$$\Pr_{(\mathsf{vk},\mathsf{sk},\mathsf{dk},\mathsf{td}_s,s,y)\leftarrow V'}[\mathcal{B}_7(h(\mathsf{sk}, \mathsf{vk})) = \mathsf{sk} | \wedge s' = s \wedge y' = y] =$$
$$= \Pr_{(\mathsf{vk},\mathsf{sk},\mathsf{dk},\mathsf{td}_s)\leftarrow V}[\mathcal{B}_6(\mathsf{vk}, h(\mathsf{sk}, \mathsf{vk})) = \mathsf{sk}] .$$

So,

$$\Pr_{(\mathsf{vk},\mathsf{sk},\mathsf{dk},\mathsf{td}_s)\leftarrow V}[\mathcal{B}_6(\mathsf{vk}, h(\mathsf{sk}, \mathsf{vk})) = \mathsf{sk}] \leq 2^{l_1 + l_4} \Pr_{(\mathsf{vk},\mathsf{sk},\mathsf{dk},\mathsf{td}_s,s,y)\leftarrow V'}[\mathcal{B}_7(h(\mathsf{sk}, \mathsf{vk})) = \mathsf{sk}] .$$

The distribution on $(\mathsf{sk}, \mathsf{vk})$ induced by V' is identical to that induced by the key generation of Σ, and $\mathcal{B}_7$ is PPT. Consequently, we can assume that

$$\Pr_{(\mathsf{vk},\mathsf{sk},\mathsf{dk},\mathsf{td}_s,s,y)\leftarrow V'}[\mathcal{B}_7(h(\mathsf{sk}, \mathsf{vk})) = \mathsf{sk}] \leq 2^{k_6} .$$

[4] Finding ek' and crs' can be done if we, without loss of generality, assume that the decryption key and simulation trapdoor are the randomness used in the respective generation algorithms.

Combining our inequalities so far, we get that

$$\Pr[W] \leq 2^{l_1+l_2+l_3+l_4-k_6} + \sum_{i=0}^{3} \varepsilon_i + u\varepsilon_4 = 2^{-k} + \sum_{i=0}^{3} \varepsilon_i + u\varepsilon_4 \ .$$

□

The following is a corollary to Thm. 2.

Theorem 3. *If H, Γ and Π have the properties listed above, then Σ is $2^{-k_{\mathrm{W}}}$-EU-CMAA where $k_{\mathrm{W}} = k + l_2 + l_3$ and l_1 is the length of the randomness used to sample s, l_2 is the length of the decryption key dk for Γ, l_3 is the length of the simulation trapdoor td_s. In particular, Σ is $2^{-k_{\mathrm{W}}}$-EU-CMAA for $k_{\mathrm{W}} = \mathrm{poly}(k)$ which do not grow with k_4, i.e, the input length of the hash function.*

If we consider the class $\mathcal{H}_{\mathsf{ow}}(\ell(k))$, then Σ is $2^{-k_{\mathrm{S}}}$-EU-CMAA where $k_{\mathrm{S}} = k + l_1 + l_2 + l_3 + l_4$ and where l_4 is the length of $y = H_s(x)$.

Our concrete instantiation has all the needed properties, except that s has a length which depends on k_4. This, however, can be handled generically as follows.

Lemma 2. *If there exists an ε-secure family of second pre-image resistant hash functions H, with key sampling algorithm Gen_H, and a δ-secure pseudo-random generator prg, then there exists an $(\varepsilon+\delta)$-secure family of second pre-image resistant hash function H, with key sampling algorithm Gen'_H, where $s \leftarrow \mathsf{Gen}'_H(1^k)$ can be guessed with probability 2^{-k_0}, where $k_0 = \mathrm{poly}(k)$ is the seed length of prg at security level k.*

Proof. Let $\mathsf{Gen}'_H(1^k; r \in \{0,1\}^{k_0}) = \mathsf{Gen}_H(1^k; \mathsf{prg}(r))$. It is clear that an output of $\mathsf{Gen}'_H(r \in \{0,1\}^{k_0})$ can be guessed with probability 2^{-k_0}, by guessing r. Let

$$\varepsilon = \Pr_{x^* \leftarrow \mathcal{A}(s,x) \wedge x \leftarrow \{0,1\}^{k_4} \wedge s \leftarrow \mathsf{Gen}_H}[H_s(x^*) = H_s(x) \wedge x^* \neq x]$$

, and let $\varepsilon' = \Pr_{x^* \leftarrow \mathcal{A}(s,x) \wedge x \leftarrow \{0,1\}^{k_4} \wedge s \leftarrow \mathsf{Gen}'_H}[H_s(x^*) = H_s(x) \wedge x^* \neq x]$. Consider the algorithm $\mathcal{B}(s)$ which samples $x \leftarrow \{0,1\}^{k_4}$ and $x^* \leftarrow \mathcal{A}(s,x)$ and outputs 1 iff $H_s(x^*) = H_s(x)$. This algorithm is PPT, and $\varepsilon' = \Pr[\mathcal{B}(\mathsf{Gen}_H(\mathsf{prg}(r \leftarrow \{0,1\}^{k_0}))) = 1]$ and $\varepsilon = \Pr[\mathcal{B}(\mathsf{Gen}_H(r \leftarrow \{0,1\}^*)) = 1]$. By the prg being a δ-pseudo-random generator, it follows that $|\varepsilon' - \varepsilon| \leq \delta$. □

Remark. We can also prove security in the stronger model, where the leakage function h sees not only sk, but the randomness used by Gen to generate $(\mathsf{vk}, \mathsf{sk})$. In that case we need that the distribution on ek induced by sampling $(\mathsf{ek}, \mathsf{dk})$ with KeyGen_Γ,the distribution of a crs sampled along with a trapdoor and that the distribution on s induced by sampling $s \leftarrow \mathsf{Gen}_H$ can all be sampled with invertible sampling. This is indeed the case for our concrete instantiation. The only problematic point is Lemma 2. Even if $\mathsf{Gen}_H(\{0,1\}^*)$ has invertible sampling, it would be very surprising if $\mathsf{Gen}_H(\mathsf{prg}(\{0,1\}^{k_0}))$ has invertible sampling. So, if the probability of guessing a random $s \leftarrow \mathsf{Gen}_H$ is not independent of the input of H_s, we cannot generically add this property. One can circumvent this problem as in [DHLAW10b] and consider s as a public parameter of the scheme. This is modeled by sampling s in a parameter generation phase prior to the key generation phase and give s as input to all entities. This would in turn make s an input to the reduction (called $\mathcal{B}_7$ in the appendix), circumventing the problem of having to guess s. We would get security when considering the class $\mathcal{H}_{\mathsf{ow}}(\ell(k))$ for $k_{\mathrm{S}} = k + l_2 + l_3 + l_4$.

4 Applications: Auxiliary Input Secure Identification Schemes

In an identification scheme ID a prover attempts to prove its identity to a verifier. For a security parameter k, ID consists of three PPT algorithms $\mathsf{ID} = (\mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$:

– $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^k)$: Outputs the public parameters of the scheme and a valid key pair.
– $(\mathsf{P}(\mathsf{pk}, \mathsf{sk}), \mathsf{V}(\mathsf{pk}))$: A (possibly) interactive protocol in which P tries to convince V of its identity by using his secret key sk. The verifier V outputs either 1 for accept or 0 for reject.

We require that ID is *complete*. This means that an honest prover will always be accepted by the verifier. The standard security definition of an identification scheme ID considers a polynomial-time adversary $\mathcal{A}$ that inputs the public key pk and interacts with the prover $\mathsf{P}(\mathsf{pk}, \mathsf{sk})$ playing the role of a verifier. Then, $\mathcal{A}$ tries to impersonate $\mathsf{P}(\mathsf{pk}, \mathsf{sk})$ by engaging in an interaction with $\mathsf{V}(\mathsf{pk})$. An identification scheme ID is *secure* if every polynomial-time adversary $\mathcal{A}$ impersonating the prover only succeeds with negligible probability.

We extend this standard security to incorporate leakage from the prover's secret key. To this end, we let the adversary obtain $h(\mathsf{pk}, \mathsf{sk})$ for an admissible leakage function $h \in \mathcal{H}$, and let it afterwards take the role of V in the execution of the protocol $(\mathsf{P}(\mathsf{pk}, \mathsf{sk}), \mathsf{V}(\mathsf{pk}))$. More formally, we have

Definition 9 (Security under Impersonation Attacks w.r.t. Auxiliary Inputs (ID-AUX security)). *An identification scheme* $\mathsf{ID} = (\mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ *is secure under impersonation attacks w.r.t. to auxiliary inputs from* $\mathcal{H}$ *if for any PPT adversary* $\mathcal{A}$ *and any function* $h \in \mathcal{H}$ *there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that, for sufficiently large* $k \in \mathbb{N}$*, the experiment below outputs* 1 *with probability at most* $\mathsf{negl}(k)$*:*

1. *Sample* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^k)$ *and give* pk *and the leakage* $h(\mathsf{pk}, \mathsf{sk})$ *to* $\mathcal{A}$*.*
2. $\mathcal{A}$ *gets access to an honest prover and runs the protocol* $(\mathsf{P}(\mathsf{pk}, \mathsf{sk}), \mathcal{A}(\mathsf{pk}))$ *for a polynomial number of times.*
3. $\mathcal{A}$ *impersonates the prover and interacts with an honest verifier* $\mathsf{V}(\mathsf{pk})$*. If* $\mathsf{V}(\mathsf{pk})$ *accepts, then output* 1*; otherwise output* 0*.*

As we explain in Section 2.5, it is possible to consider two different classes of leakage functions. We will then refer to $\ell(k) - \mathsf{IDAUX}_{\mathsf{ow}}$ security if the identification scheme is secure in the presence of leakage for functions from the class $\mathcal{H}_{\mathsf{ow}}(\ell(k))$, while we will talk about $\ell(k) - \mathsf{IDAUX}_{\mathsf{vkow}}$ security when the leakage function is from the class $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$. Such identification schemes will be secure only in presence of leakage for functions that are hard to invert given also the public key of the scheme.

4.1 Non-Interactive Identification from Signatures

Identification schemes that tolerate leakage in the bounded-retrieval model were presented in [ADW09]. We give here a construction for both notions of auxiliary input secure identification schemes. More specifically, it is a well known fact that non-interactive identification can straightforwardly be constructed from signature schemes. We show that the same holds when considering signature scheme with auxiliary input security as a building block. Formally, let $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ be a signature scheme that is secure for random messages and auxiliary input attacks with respect to class of functions $\mathcal{H}$ (cf. Definition 8). Then, the following identification scheme $\mathsf{ID}_1 = (\mathsf{KeyGen}_1, \mathsf{P}_1, \mathsf{V}_1)$ is secure auxiliary input attacks with respect to $\mathcal{H}$.

Key generation, $\mathsf{KeyGen}_1(1^k)$**:** Sample keys $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^k)$ by running the underlying key generation of Σ.

Protocol, $(\mathsf{P}(\mathsf{pk}, \mathsf{sk}), \mathsf{V}(\mathsf{pk}))$**:** The non-interactive identification uses Sig and Ver from the underlying signature scheme Sig:

1. The verifier sends a random challenge c from the message space of Σ.
2. The prover $\mathsf{P}(\mathsf{pk}, \mathsf{sk})$ computes $\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}, c)$ and sends it to the verifier.
3. The verifier accepts if $\mathsf{Ver}(\mathsf{pk}, \sigma, c) = 1$; otherwise it rejects.

Theorem 4. *Let $k \in \mathbb{N}$ be the security parameter. For any $\ell(\cdot)$, if $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is $\ell(k)$-RU-RMAA for $\mathcal{H}_{\mathsf{ow}}(\ell(k))$ (resp. for $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$) according to Definition 8, then ID_1 is $\ell(k) - \mathsf{IDAUX}_{\mathsf{ow}}$ (resp. $\ell(k) - \mathsf{IDAUX}_{\mathsf{vkow}}$) secure.*

Proof (sketch). We give a brief sketch of the proof, as the theorem follows immediately from a straightforward reduction to the security of Σ. Notice that the leakage for ID_1 can easily be simulated since the RU-RMAA security of Σ allows for leakage on the keys that are used for ID_1. □

5 Security under the K-Linear Assumption

In this section we demonstrate how to instantiate our scheme from Section 3.3 with particular primitives that yield an efficient implementation of a EU-CMAA signature scheme. Specifically, we replace each cryptographic building block we use in this scheme with an efficient construction. The hardness of our instantiated scheme follows from the K-linear assumption defined below (which essentially implies the hardness of discrete logarithms.) Notably, although we use the same building blocks, our instantiation is different and simpler than the one presented in [DHLAW10b].

Hardness assumptions. Our construction relies on the K-linear assumption. Let $\mathcal{G}$ be a group generation algorithm, which outputs $(p, \mathbb{G}, g)$ given 1^k, where $\mathbb{G}$ is the description of a cyclic group of prime order p and g is a generator of $\mathbb{G}$.

Definition 10 (K-linear Assumption). *Let $K \geq 1$ be constant. The K-linear assumption on $\mathbb{G}$ states that*

$$(\mathbb{G}, g_0, g_1, \ldots, g_K, g_1^{r_1}, \ldots, g_K^{r_K}, g_0^{\sum_{i=1}^{K} r_i}) \approx_c (\mathbb{G}, g_0, g_1, \ldots, g_K, g_1^{r_1}, \ldots, g_K^{r_K}, g_0^{r_0})$$

for $(p, \mathbb{G}, g) \leftarrow \mathcal{G}(1^k), g_0, \ldots, g_K \leftarrow \mathbb{G}$, and $r_0, r_1, \ldots, r_K \leftarrow \mathbb{Z}_p$.

For $K = 1$ we get the DDH assumption and for $K = 2$ the decisional-linear assumption. Note that K-linear implies $(K+1)$-linear. For ease of presentation, from here on we only consider the special case with $K = 2$. Hardness of K-linear implies hardness of the discrete logarithm problem:

Definition 11 (DL). *We say that* the discrete logarithm (DL) problem is hard relative to $\mathbb{G}$ *if for all PPT adversaries $\mathcal{A}$ there exists a negligible function* negl *such that*

$$\Pr[\mathcal{A}(\mathbb{G}, q, g, g^x) = x] \leq \mathsf{negl}(k) \ ,$$

where p is the order of $\mathbb{G}$ and the probabilities are taken over the choices of g and $x \leftarrow \mathbb{Z}_p$.

Definition 12 (Bilinear pairing.). *Let $\mathbb{G}$, $\mathbb{G}_T$ be multiplicative cyclic groups of prime order p and let g be a generator of $\mathbb{G}$. A map $e \colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map of $\mathbb{G}$ and $\mathbb{G}_T$ if it has the following properties:*

1. *Bi-linearity: $\forall u, v \in \mathbb{G}$, $\forall a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.*
2. *Non-degeneracy: $e(g, g)$ generates $\mathbb{G}_T$.*
3. *e is efficiently computable.*

We assume that the K-linear assumption holds in $\mathbb{G}$.

We continue with a list of building blocks used in our scheme from Section 3.3 and their instantiations:

1. Second Pre-Image Resilient Hash Function H. Let $\mathbb{G}$ be a group of prime order p and let $g_1, \ldots, g_\ell$ be generators for $\mathbb{G}$. Define $H_{\mathbb{Z}_p^\ell} = \{H_{g_1,\ldots,g_\ell}\}_{g_1,\ldots,g_\ell \in \mathbb{G}^l}$ as follow. Let $g_1, \ldots, g_\ell \leftarrow \mathsf{Gen}_H(1^k)$. For $x \leftarrow \mathbb{Z}_p^\ell$, let $H_{g_1,\ldots,g_\ell}(x) = \prod_{i=1}^{\ell} g_i^{x_i}$. Second pre-image resilience is implied by the hardness of discrete logarithm in $\mathbb{G}$. Loosely speaking, finding a collision with respect to $y \in \mathbb{G}$ is sufficient to compute $\log_g g_\ell$, given $\log_g g_1, \ldots, \log_g g_{\ell-1}$.
2. IND-WLCCA-Secure Encryption Scheme $\Pi_{\mathsf{cca}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. We use a modification of the linear Cramer-Shoup scheme [Sha07] that is IND-LCCA secure (cf. Def. 2) under the 2-linear assumption and supports labels [CCS09, DHLAW10a]. We recall that the security notion required for our proof is IND-WLCCA where the adversary cannot query the decryption oracle with the label used to compute the challenge. Furthermore, IND-LCCA on fixed length messages implies IND-WLCCA on arbitrary length messages; see Appendix A for further discussion.
   We adopt notation from [DHLAW10a] and use the notation of (a_1, a_2) to denote the elements of a vector a of length 2. For $a \in \mathbb{G}$ and $\alpha \in \mathbb{Z}_p$ we denote by $a \cdot \alpha = \alpha \cdot a$ the exponentiation a^α, vector multiplications are computed component-wise. We further distinct group elements in $\mathbb{G}$ (denoted in boldface) from exponents in $\mathbb{Z}_p$. For public parameters $(H_{\mathsf{CL}}, p, \mathbb{G}, g_0, g_1, g_2)$ where $(p, \mathbb{G}, g_0) \leftarrow \mathcal{G}(1^k), g_1, g_2 \leftarrow \mathbb{G}$, and $H_{\mathsf{CL}} : \{0,1\}^* \rightarrow \mathbb{Z}_p$ is a collision resistance hash function. Define a matrix
$$\mathbf{A} = \begin{pmatrix} g_0 & g_1 & 1 \\ g_0 & 1 & g_2 \end{pmatrix}$$
   Key Generation, KeyGen. Choose 3 random vectors $\mathsf{u}, \mathsf{v}, \mathsf{w} \leftarrow \mathbb{Z}_p^3$. Compute the multiplications
$$\mathbf{d} = \mathbf{A} \cdot \mathsf{v}, \;\; \mathbf{e} = \mathbf{A} \cdot \mathsf{w}, \;\; \mathbf{h} = \mathbf{A} \cdot \mathsf{u}.$$
   Set $(\mathsf{dk}, \mathsf{ek}) = ((\mathsf{u}, \mathsf{v}, \mathsf{w}), (\mathbf{d}, \mathbf{e}, \mathbf{h}))$.
   Encryption, Enc. To encrypt a message $m \in \mathbb{G}$ under label L, choose $\mathsf{r} \leftarrow \mathbb{Z}_p^2$ and compute the multiplication $\mathbf{y} = \mathsf{r}^\top \cdot \mathbf{A}$. Set
$$a := h_1^{\mathsf{r}_1} \cdot h_2^{\mathsf{r}_2}, \quad z = a \cdot m, \quad c = (\mathbf{d}_1(\mathbf{e}_1^t))^{r_1} \cdot (\mathbf{d}_2(\mathbf{e}_2^t))^{r_2},$$
   where $t = H_{\mathsf{CL}}(\mathbf{y}, z, L)$, $\mathbf{d} = (\mathbf{d}_1, \mathbf{d}_2)$, $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$ and $\mathsf{r} = (\mathsf{r}_1, \mathsf{r}_2)$.
   Output $C = (\mathbf{y}, z, c)$.
   Decryption, Dec. To decrypt ciphertext C under label L, parse $C = (\mathbf{y}, z, c)$. Compute
$$t = H_{\mathsf{CL}}(\mathbf{y}, z, L), \quad \tilde{c} = \mathbf{y}^\top \cdot (\mathsf{v} + t\mathsf{w}).$$
   If $\tilde{c} = c$, then output $z/(\mathbf{y}^\top \cdot \mathsf{u})$. Else output $\perp$.
   Another potential instantiation is the recent work by Camenisch et al. [CHK+11] that presents a structure preserving CCA public-key encryption under the 2-linear assumption. This scheme differs from the scheme presented above in that the consistency check is implemented using bilinear groups and thus preserves the structure of the group (i.e., it only uses algebraic operations). Another advantage is that this scheme can also be extended to encrypt multiple messages at the same time at a lower cost then encrypting each ciphertext individually, while supporting labels.
3. NIZK Argument for NP. We consider the NIZK argument of Groth-Sahai [GS08] which shows how to prove in zero-knowledge and under the 2-linear assumption that a linear system has a solution (our notations here follow from [DHLAW10a] as well). Namely, a system $(\mathbf{B}, \mathbf{c})$, where $\mathbf{B}$ is a matrix with coefficients in $\mathbb{G}^{M \times N}$ and $\mathbf{c}$ is a target vector from $\mathbb{G}^M$, is satisfiable if there exists a vector $\mathsf{u} \in \mathbb{Z}_p^N$ such that
$$\mathbf{b}_{i,1}^{\mathsf{u}_1} \mathbf{b}_{i,2}^{\mathsf{u}_2} \ldots \mathbf{b}_{i,N}^{\mathsf{u}_N} = \mathbf{c}_i$$
   for $i \in [1, \ldots, M]$, $\mathbf{B} = \{\mathbf{b}_i = (\mathbf{b}_{i_1}, \ldots, \mathbf{b}_{i_1})\}_{i=1}^M$, $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_M)$ and $\mathsf{u} = (\mathsf{u}_1, \ldots, \mathsf{u}_N)$. The simulation trapdoor for this proof system is in $\mathbb{Z}_p^2$.

Instantiation. Our instantiation for the scheme in Section 3.3 requires the following. First, for a signing key $x \in \{0,1\}^\ell$, consider the bit representation $x_1, \ldots, x_\ell$ of x and compute $H(x)$ using the hash function H from above. I.e., $H(x) = \prod_{i=1}^{\ell} g_i^{x_i}$. In order to sign a message m, the signer view m as a label for the labeled encryption scheme specified above and then computes $\mathsf{Enc}^m(\mathsf{ek}, g_1^{x_1}), \ldots, \mathsf{Enc}^m(\mathsf{ek}, g_\ell^{x_\ell})$; all with the same label m. It is easy to see that IND-WLCCA is maintained under such block-wise parallel encryption under the same label. (See Thm. 5 in the appendix for a formal proof). It then computes a NIZK proving that these ciphertexts are consistent with the hashed value $H(x)$ from the public key. Moreover, collision resistance is still hard even when evaluated on the individual bits of x. Specifically, finding a collision x, x' implies that these values differ with at least a single bit. This means that $\prod_{i=1}^{\ell} g_i^{x_i} = \prod_{i=1}^{\ell} g_i^{x'_i}$ induce two linear equations so that it is possible to find $\log_g g_\ell$, given $\log_g g_1, \ldots, \log_g g_{\ell-1}$. We note that computing the hash function on bits is necessary in order to extract x from the forgery given by the adversary.

It is left to show how the NIZK proof is defined. Formally, this can be proved as follows. Observing a ciphertext $c = (c_1, c_2, c_3)$ computed according to Enc above, we note that c_1 and c_2 are (component-wise) multiplicatively homomorphic. This implies that one can compute the (partial) encryption of $H(x)$ from $c_1 = ((c_{1_1}, c_{1_2}, c_{1_3}), \ldots, (c_{\ell_1}, c_{\ell_2}, c_{\ell_3}))$ by computing the products

$$\tilde{c}_1 = \prod_{i=1}^{\ell} c_{i_1} = \left(\prod_{i=1}^{\ell} \mathbf{y}_{i_1}, \prod_{i=1}^{\ell} \mathbf{y}_{i_2}, \prod_{i=1}^{\ell} \mathbf{y}_{i_3} \right) \quad \text{and} \quad \tilde{c}_2 = \prod_{i=1}^{\ell} c_{i_2} \ .$$

This gives us

$$\tilde{c}_2 \Big/ \prod_{i=1}^{\ell} \left(\mathbf{y}_{i_1}^{\mathsf{u}_1} \cdot \mathbf{y}_{i_2}^{\mathsf{u}_2} \cdot \mathbf{y}_{i_3}^{\mathsf{u}_3} \right) = H(x) \ .$$

Note also that this set of ciphertexts induces ℓ linear equations (in the exponent), with coefficients taken from matrix $\mathbf{A}$ and variables $\mathsf{R} = ((\mathsf{r}_{1_1}, \mathsf{r}_{1_2}), \ldots, (\mathsf{r}_{\ell_1}, \mathsf{r}_{\ell_2}))$ so that $\mathsf{R}^\top \cdot \mathbf{A} = \mathbf{Y}$, for $\mathbf{Y} = ((\mathbf{y}_{1_1}, \mathbf{y}_{1_2}, , \mathbf{y}_{1_3}), \ldots, (\mathbf{y}_{\ell_1}, \mathbf{y}_{\ell_2}, \mathbf{y}_{\ell_3}))$ and $c_{i_3} = (\mathbf{d}_1(\mathbf{e}_1^t))^{r_1} \cdot (\mathbf{d}_2(\mathbf{e}_2^t))^{r_2}$ for all i. Furthermore, in order to impose consistency with $H(x)$ we require that

$$\prod_{i=1}^{\ell} \left(\mathbf{h}_{i_1}^{\mathsf{r}_{i_1}} \cdot \mathbf{h}_{i_2}^{\mathsf{r}_{i_2}} \right) = \left(\prod_{i=1}^{\ell} c_{i_2} \right) \Big/ H(x).$$

This gives another linear equation.

As for the concrete parameters, let $k = \log p$. We get that the length of the decryption key is $l_2 = 3k$, the length of the simulation trapdoor is $l_3 = 2k$, and the length of the output of H is $l_4 = k$. The length of the description of H is $l_1 = \ell k$, but this can be brought down to $l_1 = 2k$ using Lemma 2, as it is trivial to build a pseudo-random generator $\mathbb{G}^2 \to \mathbb{G}^\ell$ using the 2-linear assumption. So, by Thm. 3, we get existential unforgeability with $k_{\mathrm{W}} = k + l_2 + l_3 = 6k$. If we consider the class $\mathcal{H}_{\mathsf{ow}}(\ell(k))$, we have security with $k_{\mathrm{S}} = k + l_1 + l_2 + l_3 + l_4 = 9k$.

Acknowledgments. The authors thank Yevgeniy Dodis for discussions at an early stage of this project.

References

[ADW09] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *CRYPTO*, pages 36–54, 2009.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *TCC*, pages 474–495, 2009.

[BG10] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *CRYPTO*, pages 1–20, 2010.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *CRYPTO*, pages 108–125, 2008.

[BS11] Zvika Brakerski and Gil Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In *CRYPTO*, pages 543–560, 2011.

[BSW11] Elette Boyle, Gil Segev, and Daniel Wichs. Fully leakage-resilient signatures. In *EUROCRYPT*, pages 89–108, 2011.

[CCS09] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *EUROCRYPT*, pages 351–368, 2009.

[CHK+11] Jan Camenisch, Kristiyan Haralambiev, Markulf Kohlweiss, Jorn Lapon, and Vincent Naessens. Structure preserving cca secure encryption and applications. In *ASIACRYPT*, pages 89–106, 2011.

[DGK+10] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *TCC*, pages 361–381, 2010.

[DHLAW10a] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Cryptography against continuous memory attacks. In *FOCS*, pages 511–520, 2010.

[DHLAW10b] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Efficient public-key cryptography in the presence of key leakage. In *ASIACRYPT*, pages 613–631, 2010.

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

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

[FKPR10] Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy N. Rothblum. Leakage-resilient signatures. In *TCC*, pages 343–360, 2010.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, pages 415–432, 2008.

[KR00] Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *NDSS*, 2000.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *ASIACRYPT*, pages 703–720, 2009.

[LRW11] Allison B. Lewko, Yannis Rouselakis, and Brent Waters. Achieving leakage resilience through dual system encryption. In *TCC*, pages 70–88, 2011.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *TCC*, pages 278–296, 2004.

[MTVY11] Tal Malkin, Isamu Teranishi, Yevgeniy Vahlis, and Moti Yung. Signatures resilient to continual leakage on memory and computation. In *TCC*, pages 89–106, 2011.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages 18–35, 2009.

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

[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.

[RS91] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO*, pages 433–444, 1991.

[Sha07] Hovav Shacham. A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074, 2007.

[Sta11] Francois-Xavier Standaert. Leakage resilient cryptography: a practical overview. invited talk at ECRYPT Workshop on Symmetric Encryption (SKEW 2011), 2011.

A From WLCCA on Short Messages to WLCAA on Long Messages

In this section we prove that weak CCA secure scheme with labels (WLCCA) on fixed length messages implies weak CCA security with labels on arbitrary length messages. The same proof also holds for the case the underline building block for fixed length message is CCA secure with labels. Formally, let $\varGamma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an LPKE scheme with message space M. Let $\varDelta = (\mathsf{KeyGen}, \mathsf{Encl}, \mathsf{Decl})$ be LPKE with message space M^n, where

Encryption (long message) $\mathsf{Encl}^L(\mathsf{ek}, (x_1, \ldots, x_n)) = (\mathsf{Enc}^L(\mathsf{ek}, x_i))_{i=1}^n$.
Decryption (long message) $\mathsf{Decl}^L(\mathsf{dk}, (c_1, \ldots, c_n)) = (\mathsf{Dec}^{(L,i)}(\mathsf{dk}, c_i))_{i=1}^n$.

Theorem 5. *If $\varGamma$ is IND-WLCCA, then $\varDelta$ is IND-WLCCA.*

Proof. Let $\mathcal{B}$ be a PPT adversary for the IND-WLCCA game against Δ. We build a PPT adversary $\mathcal{A}$ for the IND-WLCCA game against Δ. It runs as follows:

1. Get ek as input, and input ek to $\mathcal{B}$.
2. When $\mathcal{B}$ queries its decryption oracle with $(L, (c_1, \ldots, c_n))$, then for $i = 1, \ldots, n$, query the decryption oracle to get $x_i = \mathsf{Dec}^L(\mathsf{dk}, c_i)$ and return $(m_1, \ldots, m_n)$ to $\mathcal{B}$.
3. When $\mathcal{B}$ outputs the challenge $(L, (x_{0,1}, \ldots, x_{0,n}), (x_{1,1}, \ldots, x_{1,n}))$, sample $i \leftarrow \{1, \ldots, n\}$. For $j = 1, \ldots, i-1$, let $c_j \leftarrow \mathsf{Enc}^L(\mathsf{ek}, x_{0,j})$. For $j = i$, output $(L, x_{0,j}, x_{1,j})$ to the game and get back $c_j \leftarrow \mathsf{Enc}^L(\mathsf{ek}, x_{b,j})$. For $j = i+1, \ldots, n$, let $c_j \leftarrow \mathsf{Enc}^L(\mathsf{ek}, x_{1,j})$. Return $(c_1, \ldots, c_n)$ to $\mathcal{B}$.
4. When $\mathcal{B}$ outputs a guess b', output b'.

In this proof, we only consider a $\mathcal{B}$ which is admissible. This means that it does not ask to decrypt any ciphertext with the label L. This implies that $\mathcal{A}$ will not ask to decrypt any ciphertext with the label L either. Since the challenge ciphertext of $\mathcal{A}$ has label L, it follows that $\mathcal{A}$ is admissible.

To prove security, it is clearly sufficient to prove that the absolute distance between the probability that an adversary guesses $b' = 1$ when $b = 1$ and the probability that it guessed $b' = 1$ when $b = 0$ is negligible. Since we prove security against all adversaries it is sufficient to consider the signed distance, as an adversary can always flip its guess should it have a negative signed advantage.

Let b'_{i_0,b_0} be the distribution of the output of the $\mathcal{B}$ inside $\mathcal{A}$ when the i sampled by $\mathcal{A}$ happens to be $i = i_0$ and the b in the IND-WLCCA game happens to be $b = b_0$. When $b = 0$, then the message encrypted by $\mathcal{A}$ is $(x_{0,1}, \ldots, x_{0,i-1}, x_{0,i}, x_{1,i+1}, \ldots, x_{1,n})$. When $b = 1$, then the message encrypted by $\mathcal{A}$ is $(x_{0,1}, \ldots, x_{0,i-1}, x_{1,i}, x_{1,i+1}, \ldots, x_{1,n})$. It follows that $b'_{i,0} = b'_{i+1,1}$. By the IND-WLCCA security of Γ we have that $\Pr[b'_{i,0} = 1] - \Pr[b'_{i,1} = 1] \leq \mathsf{negl}$. So, we have that $\Pr[b'_{i+1,1} = 1] - \Pr[b'_{i,1} = 1] \leq \mathsf{negl}$. Using telescoping we get that $\Pr[b'_{n,1} = 1] - \Pr[b'_{1,1} = 1] \leq (n-1) \cdot \mathsf{negl} \leq \mathsf{negl}$. By the IND-WLCCA security of Γ we have that $\Pr[b'_{1,1} = 1] - \Pr[b'_{1,0} = 1] \leq \mathsf{negl}$. So, $\Pr[b'_{n,1} = 1] - \Pr[b'_{1,0} = 1] \leq 2 \cdot \mathsf{negl} \leq \mathsf{negl}$. Let b'_{b_0} be the distribution of the output of $\mathcal{B}$ in the IND-WLCCA game against Γ when the b in the IND-WLCCA game happens to be $b = b_0$. By construction $b'_0 = b'_{1,0}$ and $b'_1 = b'_{n,1}$. So, $\Pr[b'_1 = 1] - \Pr[b'_0 = 1] \leq \mathsf{negl}$, as desired. $\square$