An extended abstract of this paper is published in the proceedings of ASIACRYPT 2012.

Signature Schemes Secure against Hard-to-Invert Leakage

Sebastian Faust[*] Carmit Hazay[†] Jesper Buus Nielsen[‡] Peter Sebastian Nordholt[§]
Angela Zottarel[¶]

Abstract

Side-channel attacks allow the adversary to gain partial knowledge of the secret key when cryptographic protocols are implemented in real-world hardware. The goal of leakage resilient cryptography is to design crytosystems that withstand such attacks. 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 any *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 (where both the challenge as well as the signatures seen prior to that are computed on random messages). Here, polynomial-hardness is required even when given the entire public-key. We further show that such signature schemes readily give us auxiliary input secure identification schemes.

[*]EPFL, Lausanne, Switzerland. Email: sebastian.faust@gmail.com. Received funding from the Marie Curie IEF/FP7 project GAPS, grant number: 626467.

[†]Faculty of Engineering, Bar-Ilan University, Israel. Email: carmit.hazay@biu.ac.il.

[‡]Department of Computer Science, Aarhus University, Denmark. Email: jbn@cs.au.dk. Supported by European Research Commission Starting Grant 279447. Supported by Danish Council for Independent Research Starting Grant 10-081612. The authors acknowledge support from 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, from the CFEM research center (supported by the Danish Strategic Research Council) within which part of this work was performed.

[§]Alexandra Institute, Denmark. Email: peter.s.nordholt@alexandra.dk.

[¶]Department of Computer Science, Aarhus University, Denmark. Email: angela@cs.au.dk.

1 Introduction

Leakage resilient cryptography. Modern cryptography analyzes the security of cryptographic algorithms in the *black-box* model. Namely, an adversary may view the algorithm's inputs and outputs, but the secret key as well as all the internal computation remain perfectly hidden. For instance, consider the classic security definition of signature schemes [GMR88] where the adversary is given the verification key and block-box access to the signing algorithm. Still, the adversary cannot obtain (or exploit) any information about the secret state of the signer during its attack.

Unfortunately, the assumption of perfectly hidden keys does not reflect practice as demonstrated by a large volume of works on side-channel attacks [Koc96, BDL97, BS97, KJJ99, QS01, GMO01, HSH+09], since when implementing cryptographic protocols in real-world hardware some information on the secret key may leak to the adversary. 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; see an example regarding the RSA and AES cryptosystems [HSH+09].

In the last years, significant progress has been made within the theory community to incorporate information leakage into the black-box model. To this end, these works develop new models to formally describe the information leakage [MR04, AGV09, SMY09] and design new schemes that can be proven secure therein. This recent line of works (cf. [MR04, DP08, AGV09, ADW09, DKL09, NS09, FKPR10, LRW11, DF12, HLAWW13] and many more) presents leakage resilient cryptographic primitives with security proven even in the presence of arbitrary (but somewhat restricted) leakage from the secret key. These works design leakage resilient primitives both in the secret key and public key settings, including stream ciphers [DP08, Pie09], MACs [HLAWW13] and public key encryptions [AGV09, NS09]. In this paper we focus on digital signature schemes; see a broader discussion below.

Leakage modeling. Loosely speaking, 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 during the security game.

More formally, in the *bounded* leakage model [AGV09], 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 $\lambda \ll |\mathsf{sk}|$. Namely, in the context of signature schemes, the adversary is allowed to specify the description of a leakage function h as above (that may be chosen based on the verification key), and learns the leakage $h(\mathsf{sk})$ in addition to any information it is meant to learn during the security game (such as the verification key and valid signatures).

This restriction can be weakened in several ways. For instance, 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.

1.1 The Auxiliary Input Model

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]. Namely, 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 "hard" to compute sk. Such *hard-to-invert* leakage is 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 consider two classes of hard-to-invert leakage functions:

1. A function h of the secret key sk is *polynomially* hard-to-invert auxiliary information, if there exists a negligible function negl such that for sufficiently large $k = |\mathsf{sk}|$, any polynomial-time adversary will succeed with probability at most $\mathsf{negl}(k)$ in inverting $h(\mathsf{sk})$.

2. A function h of the secret key sk is *exponentially* hard-to-invert auxiliary information if there exists a constant $c > 0$ such that for sufficiently large $k = |\mathsf{sk}|$, any polynomial-time adversary $\mathcal{A}$ will succeed with probability 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.

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 proposed 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 variant of the learning parity with noise (LPN) assumption with respect to any *exponentially* hard-to-invert function. 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 the Decisional Diffie-Hellman (DDH) hardness assumption and variants of the GPV encryption scheme [GPV08] based on the learning with errors (LWE) hardness assumption, are secure with respect to auxiliary input leakage. All their schemes remain secure under *sub-exponentially* hard-to-invert leakage.[1] As discussed in their work, some important subtleties arise in the public key setting 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, which may make the scheme insecure if the adversary also obtains additional auxiliary input leakage about the secret key. For instance, consider the setting where the public key pk contains the first $k/2$ bits of the secret key. In this case, there is no hope to prove security with respect to $2^{-k/2}$ hard to compute leakage functions. Hence, the definition of the set of admissible leakage functions needs to take into account also the information that is revealed by the public key. To handle this issue, Dodis et al. [DGK$^+$10] proposed a natural notion of auxiliary input security, which says that a leakage function is admissible if it is hard to compute the secret key *even* when given the auxiliary input leakage *together* with the public key (we point out that this notion is called "weak" in [DGK$^+$10] since the class of admissible leakage functions now becomes smaller). A more detailed discussion on this issue can be found in [DGK$^+$10].

Following Dodis et al. [DGK$^+$10], in this paper, we will usually first prove security in the weaker setting where we consider only leakage functions that are hard to invert given also the public key. As shown in [DGK$^+$10], when the public key is short this weaker notion of auxiliary input security 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

[1] A function h of the secret key sk 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}.

guessed within the proof, which implies that the hardness assumption gets stronger when applying this proof technique (in particular, such a guessing strategy always results in an exponential loss in the hardness assumption).

Notice that the distinction between the weak and strong model only affects the size of the set of admissible leakage functions. More precisely, as in the traditional "non-leaky" setting the adversary is allowed to always see the public key as well when it attacks the signature scheme. The distinction between the weak and strong model is that in the weak model we assume that the additional auxiliary input leakage it obtains is *even* hard to invert when given that public key, while in the strong (and desired) model we only assume that the function is hard to invert without considering the public key. Notice that in the latter setting the set of admissible leakage functions can be much larger, and hence the result becomes stronger.

While in general we aim for the stronger notion of auxiliary input security, we further note that as outlined in [DGK+10] the 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, then the scheme remains secure even if the adversary sees arbitrary signatures – as these signatures can be viewed as polynomially hard-to-invert leakage.

More recently, Brakerski and Goldwasser [BG10] and Brakerski and Segev [BS11] proposed further constructions of public key encryption 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 Quadratic Residuosity (QR) and Decisional Composite Residuosity (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. Finally, a more recent work by Yuen et al. [YCZY12] presents the first identity-based encryption scheme with security in the presence of continual auxiliary input leakage, by applying a modified theorem of Goldreich-Levin. Their security model allows leakage from both the master secret key as well as identity-based secret keys.

1.2 Our Contributions

In this work, we will analyze the security of digital signature schemes in the presence of *computationally hard-to-invert* leakage. We show somewhat surprisingly that simple variants of constructions for the bounded and noisy leakage settings also achieve security with respect to the more *general* class of hard-to-invert leakage. We stress that our work is theoretical in nature, and it is unclear to what extent it would offer any protection against real-world side-channel attacks.

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 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 certain 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 discussed above, 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. Next, 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 non-interactive zero-knowledge argument (NIZK) 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, as hinted above, 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 the verification key $\mathsf{vk} = H(\mathsf{sk})$, 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 passive 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 in the weaker model 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 the larger class of exponentially hard-to-invert leakages, and hence provides a strengthening of the security proofs given in the bounded leakage model of [KV09, DHLAW10b, BSW11].

One may object that artificially adding an encryption of the secret key to the signature is somewhat counter-intuitive, as it seems that we obtain provable security by just reducing the security of the signature scheme to a point where signatures are no longer allowed leakage. This is actually not the case. The better way to see the construction is that the encryption forces signatures to be so *long* that leaking them is at least as hard as leaking the secret key, and then we just have to pick the security parameters such that it is hard enough to guess dk and hard enough to leak the secret key (which is the goal for all leakage resilient schemes). To elaborate on this, notice that all that is needed for security of our scheme is that guessing dk is significantly easier than guessing sk. For a given desirable security level we can therefore *first* pick the length of dk, as to achieve the desirable security level, and after that pick the length of sk long enough to get a meaningful leakage bound. For the example above, if we instantiate the scheme with a larger security parameter $k' = k^{1/c}$, then we can allow p to be exponentially small, say 2^{dk} for $0 < d < 1$. In that case the signature scheme can still have exponential security. After that we can then, for instance, pick $|\mathsf{sk}| = 100|\mathsf{dk}|$. This ensures that even after leaking 98% of the bits of the secret key, it is easier to guess dk than sk and hence our leakage class will in particular include the leakage of 98% of the secret key. Instantiating any cryptographic primitive in practice will involve such considerations of how to instantiate the security parameters. As this is particularly the case for our scheme, we provide a concrete security analysis, which allows to conveniently instantiate our scheme with any desirable security level. Note, also, that adding trapdoors to cryptographic schemes for what superficially only seems to be proof reasons is common in the field, e.g., non-interactive zero-knowledge is another prominent example.

For readers familiar with the security proof of the Katz-Vaikuntanathan scheme from [KV09], we note that the crux of our new proof is that in the reduction we cannot generate a CRS together with its simulation trapdoor due to technicality arises in the definition of the admissible class of leakage functions (see more discussion in Section 3.1). 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.

Auxiliary input secure identification schemes. One particular immediate application is non-interactive identification schemes with auxiliary input security. We define two notions of passive security for identifications scheme in the presence of auxiliary input, and show that auxiliary input secure signature schemes, with

random messages and a random challenge, imply these notions. Active security for identification schemes still remains an open question. In particular, known transformations from passive to active security only apply when the underlying building block is Σ-protocols. These do not apply in the auxiliary input setting.

Instantiation under the 2-linear assumption. As a concrete example, we show how to instantiate our generic transformation using the Groth-Sahai proofs system based on the 2-linear assumption [GS08]. This yields security with respect to any 2^{-6k}-hard-to-invert leakage where k is the security parameter. If we do not wish to define the hardness with respect to the public key as well, it is possible to guess it and thus 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 is hard.

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 probabilistic polynomial-time. 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 say that a function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* if for every polynomial $p(\cdot)$ there exists an integer $n_p \in \mathbb{N}$ such that $f(n) < 1/p(n)$, for every $n > n_p$. Finally, we specify the definition of computational indistinguishability.

Definition 2.1 (Computational indistinguishability by circuits) *Let $X = \{X_n(a)\}_{n\in\mathbb{N},a\in\{0,1\}^*}$ and $Y = \{Y_n(a)\}_{n\in\mathbb{N},a\in\{0,1\}^*}$ be distribution ensembles. We say that X and Y are* computationally indistinguishable*, denoted $X \approx Y$, if for every family $\{C_n\}_{n\in\mathbb{N}}$ of polynomial-size circuites, there exists a negligible function* negl *such that for all $a \in \{0,1\}^*$,*

$$|\Pr[C_n(X_n(a)) = 1] - \Pr[C_n(Y_n(a)) = 1]| < \mathsf{negl}(n).$$

2.1 Public Key Encryption Schemes

We specify the notion of a labeled public key encryption scheme following the notation used in [CCS09, DHLAW10b]. In this work we require a weaker notion of security, called IND-WLCCA, where the adversary cannot query the decryption oracle with label L such that L is the label picked for the challenge. (This is in contrast to the IND-LCCA notion where the adversary is not allowed to query the decryption oracle on (L, c), where c is the challenge ciphertext). We further motivate this security notion in Section 3.3.

Definition 2.2 (LPKE) *We say that a tuple of PPT algorithms $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ 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* $c \leftarrow \mathsf{Enc}^L(\mathsf{ek}, m)$, *then with probability* 1 *outputs* m. *We denote this by* $m \leftarrow \mathsf{Dec}^L(\mathsf{dk}, c)$.

Definition 2.3 (IND-WLCCA secure encryption scheme) *We say that a labeled public key encryption scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* IND-WLCCA 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* $\mathsf{IND\text{-}LCCA}_{\Pi,\mathcal{A}}(k)$ *that* $\mathcal{A}$ *wins the* $\mathsf{IND\text{-}WLCCA}$ *game as defined below is at most* $\mathsf{IND\text{-}LCCA}_{\Pi,\mathcal{A}}(k) \leq \frac{1}{2} + \mathsf{negl}(k)$.

- IND-WLCCA game*:*

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

An adversary is admissible if it does not query $\mathsf{Dec}^{(\cdot)}(\mathsf{dk}, \cdot)$ *with* $(L, \cdot)$ *where* L *is the label picked to compute the challenge.*

2.2 Non-Interactive Zero-Knowledge Arguments (of Knowledge)

A non-interactive zero-knowledge argument 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. Finally, the verifier V takes as input (crs, x, π) and outputs 0 or 1 (respectively rejecting or accepting the proof). Moreover, security is formalized in the following definition.

Definition 2.4 (NIZK) *A* non-interactive zero-knowledge argument (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: *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* $x \notin L$

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

Zero-knowledge: *There exist a PPT simulator* $S = (S_1, S_2)$ *such that*

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

for all PPT adversaries $\mathcal{A}$, *where* $\mathcal{O}_0^{\mathsf{crs}}(\cdot)$ *is an oracle with state* crs *such that* $\mathcal{O}_0^{\mathsf{crs}}(x, \omega) = P(\mathsf{crs}, x, \omega)$ *if* $(x, \omega) \in R_L$ *and* $\mathcal{O}_0^{\mathsf{crs}}(x, \omega) = \perp$ *otherwise, whereas* $\mathcal{O}_1^{\mathsf{crs},\mathsf{td}_s}$ *is an 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* $(x, \omega) \in R_L$ *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.*

For our first construction that is describe in Section 3.2, where security holds with respect to random messages, we need the additional property of proof of knowledge. For completeness, we specify below the formal definition for non-interactive zero-knowledge argument of knowledge (NIZKPoK).

Definition 2.5 (NIZKPoK) *A* non-interactive zero-knowledge argument $(\mathsf{NIZKPoK})$ *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 Definition 2.4.*

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 $\mathcal{A}$:

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

Zero-knowledge: *As in Definition 2.4.*

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 a random input x, it is infeasible for any PPT adversary to find x' such that $x \neq x'$ and $H_s(x) = H_s(x')$. Formally,

Definition 2.6 (Second preimage resistance (SPR)) *A family of hash functions H_s is* second preimage resistant *if for all PPT adversaries $\mathcal{A}$ there exists a negligible function* negl *such that*

$$\Pr[\mathsf{Hash}_{\mathcal{A},H}(k) = 1] \leq \mathsf{negl}(k)$$

where game Hash *is defined as follows:*

1. *Key s is sampled by running $s \leftarrow \mathsf{Gen}_H(1^k)$ together with $x \leftarrow \{0,1\}^{\ell'(k)}$.*
2. *The adversary $\mathcal{A}$ is given (s, x) and outputs x'.*
3. *The output of the game is 1 if and only if $x \neq x'$ and $H_s(x) = H_s(x')$. In such a case we say that $\mathcal{A}$ wins the game.*

2.4 Signature Schemes

A *signature scheme* is a tuple of PPT algorithms $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ defined as follows. The key generation algorithm Gen, on input 1^k 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. We formally define our two notions of security.

Definition 2.7 (Existential unforgeability under chosen message and auxiliary input attacks) *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 $h \in \mathcal{H}$, there exists a negligible function* negl *such that*

$$\Pr[\mathsf{CMA}_{\Sigma,\mathcal{A},h}(k) = 1] \leq \mathsf{negl}(k)$$

where game $\mathsf{CMA}_{\Sigma,\mathcal{A},h}(k)$ is defined as follows:

Experiment $\mathsf{CMA}_{\Sigma,\mathcal{A},h}(k)$
$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^k)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}(\mathsf{sk},\cdot)}(1^k, h(\mathsf{vk}, \mathsf{sk}), \mathsf{vk})$ *such that m^* was never submitted to $\mathcal{O}(\mathsf{sk}, \cdot)$*
Return $\mathsf{Ver}(\mathsf{vk}, m^*, \sigma^*)$.

Oracle $\mathcal{O}(\mathsf{sk}, m)$
Return $(m, \mathsf{Sig}(\mathsf{sk}, m))$.

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 2.8 (Random message unforgeability under random message and auxiliary input attacks) *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 adversaries $\mathcal{A}$ and any function $h \in \mathcal{H}$, there exists a negligible function* negl *such that*

$$\Pr[\mathsf{RMA}_{\Sigma,\mathcal{A},h}(k) = 1] \leq \mathsf{negl}(k)$$

where game $\mathsf{RMA}_{\Sigma,\mathcal{A},h}(k)$ is defined as follows:

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

Oracle $\mathcal{O}(\mathsf{sk}, \cdot)$
$m \leftarrow \mathcal{M}$
Return $(m, \mathsf{Sig}(\mathsf{sk}, m))$.

We notice that the notion of unforgeability under random messages is useful in some settings. For instance, it suffices in order to construct 2-rounds identification schemes w.r.t auxiliary inputs. In Section 4 we propose formal definitions and simple constructions of identification schemes with security in the presence of auxiliary input leakage.

2.5 Classes of Auxiliary Input Functions

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 auxiliary input 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 these classes. Concretely,

1. We denote by $\mathcal{H}_{\mathsf{ow}}(\ell(k))$ the class of polynomial-time computable functions $h : \{0,1\}^{|\mathsf{sk}|+|\mathsf{vk}|} \rightarrow \{0,1\}^*$ such that given $h(\mathsf{sk}, \mathsf{vk})$, no PPT adversary can find sk with probability $\ell(k) \geq 2^{-k}$ or greater, i.e., for any PPT adversary $\mathcal{A}$

$$\Pr_{(\mathsf{sk},\mathsf{vk})\leftarrow\mathsf{Gen}(1^k)}[\mathsf{sk} \leftarrow \mathcal{A}(h(\mathsf{sk}, \mathsf{vk}))] < \ell(k).$$

2. We denote by $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$ the class of polynomial-time computable functions $h : \{0,1\}^{|\mathsf{sk}|+|\mathsf{vk}|} \rightarrow \{0,1\}^*$ such that given $(\mathsf{vk}, h(\mathsf{sk}, \mathsf{vk}))$, no PPT adversary can find sk with probability $\ell(k) \geq 2^{-k}$ or greater, i.e., for any PPT adversary $\mathcal{A}$

$$\Pr_{(\mathsf{sk},\mathsf{vk})\leftarrow\mathsf{Gen}(1^k)}[\mathsf{sk} \leftarrow \mathcal{A}(\mathsf{vk}, h(\mathsf{sk}, \mathsf{vk}))] < \ell(k).$$

Security with respect to auxiliary input gets stronger if $\ell(k)$ is larger. Therefore, our goal is typically to make $\ell(k)$ as large as possible as long as it is a negligible function. Moreover, in case $\ell(k) < 2^{-|\mathsf{sk}|}$ then our definitions are trivialized since then no leakage is admissible. If a scheme is EU-CMAA for $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$ according to Definition 2.7, 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.

Note that in the definition of $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$, we ask that it is hard to compute the secret key, when given the public key in addition to the leakage, which means that the allowed leakage functions *depend on the information in the verification key*, which might make it very hard to intuitively understand what leakage functions are allowed. In contrast, when defining $\mathcal{H}_{\mathsf{ow}}(\ell(k))$, we ask that the secret key is hard to compute given only the leakage. Therefore the leakage class $\mathcal{H}_{\mathsf{ow}}(\ell(k))$ is defined *independently of the signature scheme*, and hence it is much easier to understand what leakage functions are allowed. It is therefore primarily security against $\mathcal{H}_{\mathsf{ow}}(\ell(k))$ that we are interested in.

However, 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 then be achieved by a relation between $\mathcal{H}_{\mathsf{ow}}(\cdot)$ and $\mathcal{H}_{\mathsf{vkow}}(\cdot)$ proven by Dodis et al. [DGK+10], given in Lemma 2.1 below. For this relation to make sense it will be important that our public keys have a length which is independent of the length of the secret key. We will elaborate on this issue after each of our main theorems.

Lemma 2.1 ([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 2.1 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 will not 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 inspired by [KV09], that may seem secure at first glance. Unfortunately, proving its security seems 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:

1. A family of second preimage resistance (SPR) hash functions H.
2. A non-interactive zero-knowledge argument 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 $y = H(x)$. The verification key 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 argument 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 $\mathsf{sk} = x$, $\mathsf{vk} = (H(x), \mathsf{crs})$.

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 argument 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 of 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 $\mathsf{vk} = (y, \mathsf{crs})$. 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 a signature query. Unfortunately, at that point $\mathcal{B}$ cannot answer this query as it cannot simulate a proof without knowing the witness or the trapdoor.

An alternative approach for proving security with respect to the leakage class $\mathcal{H}_{\mathsf{ow}}(\ell(k)$ is to let $\mathcal{B}$ sample the CRS itself using the simulator for the non-interactive zero-knowledge argument in order to ensure that $\mathcal{B}$ knows the trapdoor. Unfortunately, this approach is also deemed to fail as in this case $\mathcal{B}$ cannot efficiently find $y = H(\mathsf{sk})$ that is consistent with the leakage. Moreover this results into several additional 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, for $\mathcal{B}$ who knows the trapdoor (and 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 An RU-RMAA Signature Scheme

In this section we present our construction of a RU-RMAA signature scheme as defined in Definition 2.8, where both the message queries as well as the challenge are picked at random in the security game. For this scheme we require the following building blocks:

1. A family H of second preimage resistant hash functions (cf. Definition 2.6) with input length k_1 and key sampling algorithm Gen_H. We require that the output length of H in independent of the input length. We use $q(k)$ to denote the output length of H, where q is a polynomial.

2. A $\mathsf{NIZKPoK}$ system $\Pi = (\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$ (cf. Definition 2.5) for proving knowledge of a secret value x so that $y = H_s(x)$ given s and y. We further require that the CRS's of Π are uniformly random strings of some length $p(k)$ for security parameter k and some polynomial $p(\cdot)$. We require that $p(k)$ depends only k and the scheme, not the length of the witnesses y that the proof can handle. Denote the message space $\mathcal{M}$ by $\{0,1\}^{p(k)}$.

The main idea for this scheme is inspired by the work of Malkin et al. [MTVY11] where we view each message m as a common reference string for the argument system Π. Due to the fact that 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 $s \leftarrow \mathsf{Gen}_H(1^k)$. Sample $x \leftarrow \{0,1\}^{k_1}$ 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 an argument 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 $\mathsf{V}(\mathsf{crs}, \mathsf{vk}, \sigma)$.

We are now ready to prove our theorem.

Theorem 3.1 *Assume that H is a second preimage resistant family of hash functions and that $\Pi = (\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$ is a $\mathsf{NIZKPoK}$ system. Then $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is a $\mathsf{negl}(k)$-RU-RMAA signature scheme.*

Remark. Notice that we only prove security for the leakage class $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$, for $\ell(k) = \mathsf{negl}(k)$. We can, however, use Lemma 2.1 to obtain security for the class $\mathcal{H}_{\mathsf{ow}}(2^{-t(k)}\ell(k))$, where $t(k)$ is the length of our public key. Note that the leakage class $\mathcal{H}_{\mathsf{ow}}(2^{-t(k)}\ell(k))$ only asks that the leakage is hard to invert when the adversary is *not* given the public key. This in particular means that the leakage class is *independent* of the hash function used, except through the length of a hash, which is why we require the length of a digest to be independent of the input x to the hash function. Another consequence is that the bound $2^{-t(k)}\ell(k)$ is independent of the length of the secret key x. We can therefore set the length of x to much longer than $t(k) + \log_2(\ell(k)^{-1})$. If we, e.g., let $|x| = 100(t(k) + \log_2(\ell(k)^{-1}))$, then $\mathcal{H}_{\mathsf{ow}}(2^{-t(k)}\ell(k))$ in particular includes the leakage functions which leak up to 98% of the bits of the secret key. Besides this it *additionally* includes all the leakage functions which information theoretically leaks x but still renders it $2^{-t(k)}\ell(k)$-hard to guess x in polynomial time.

The intuition of the proof is that if one can efficiently forge a signature on a random m^* after getting signatures on random messages then one can also efficiently compute x, contradicting the assumption that the leakage is hard to efficiently invert. Specifically, during the simulated attack the signatures on random messages are simulated by sampling $m = \mathsf{crs}$, where crs is sampled along with the simulation trapdoor. Then, at the challenge phase, one samples $m^* = \mathsf{crs}$, where crs is sampled along with the extraction trapdoor. Consequently, upon receiving a forgery on m^*, it is possible to extract x using the extraction trapdoor.

We note that in the standard setting a simple modification to our construction 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 the same problems specified in Section 3.1 are encountered here as well. In Section 3.3 we demonstrate 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 2.8 for a 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 2.5.
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 2.5.
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 2.5 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 2.8, Σ is a $\mathsf{negl}(k)$-RU-RMAA signature scheme. ■

Remark. 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 An EU-CMAA Signature Scheme

In this section we describe our second construction and build a EU-CMAA signature scheme. Recalling that k denote the security parameter for the signature scheme, our construction employs the following tools:

1. A family of second preimage resistant hash functions H with key sampling algorithm Gen_H (cf. Definition 2.6) where (**i**) the input length can be set to any $l_{\mathsf{in}} = \mathrm{poly}(k)$, (**ii**) the length of the randomness used by $s \leftarrow \mathsf{Gen}_H(1^k)$ is some $l_{\mathsf{s}} = \mathrm{poly}(k)$ independent of l_{in} and (**iii**) the length of an output $y = H_s(x)$ is some $l_{\mathsf{out}} = \mathrm{poly}(k)$ independent of l_{in}. 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 2.3), where the length of dk is some $l_{\mathsf{dk}} = \mathrm{poly}(k)$ 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 2.4), where the length of the simulation trapdoor td_s is some $l_{\mathsf{td}_s} = \mathrm{poly}(k)$ independent of the size of the proofs that the NIZK can handle.

We stress that the reason we use the IND-WLCCA security notion is that our signature scheme requires to encrypt its secret key which 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 the security of labeled public-key encryption schemes for arbitrary length massages is not implied by the security of IND-LCCA encryption 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 of the modified ciphertext. We therefore work with a weaker notion of security that is sufficient for our purposes to design secure signature schemes, and is easier to instantiate as demonstrated in Section 5.

Our signature scheme Σ is formally defined 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 $x \leftarrow \{0,1\}^{l_{\mathsf{in}}}$, 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 $\exists x$ such that $(C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. 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 imply a specific instantiation of *true-simulation extractable (tSE)*. An alternative instantiation of tSE 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 3.2 *Assume* H, $\Gamma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *and* $\Pi = (\mathsf{CRSGen}, \mathsf{P}, \mathsf{V})$ *have the properties listed above. Then the following holds:*

1. *If we consider the class* $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$*, then* Σ *is* $2^{-k_{\mathrm{W}}}$*-EU-CMAA where* $k_{\mathrm{W}} = k + l_{\mathsf{dk}} + l_{\mathsf{td}_s}$.
2. *If we consider the class* $\mathcal{H}_{\mathsf{ow}}(\ell(k))$*, then* Σ *is* $2^{-k_{\mathrm{S}}}$*-EU-CMAA where* $k_{\mathrm{S}} = k + l_{\mathsf{s}} + l_{\mathsf{dk}} + l_{\mathsf{td}_s} + l_{\mathsf{out}}$.

Specifically, we claim that the best success against Σ in the forging game with $2^{-k_{\mathrm{W}}}$-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 with a security parameter k_{td_s},
- ε_1 is the success probability of some PPT adversary in the soundness game against Π with a security parameter k_{td_s},
- ε_2 is the probability that some PPT adversary wins the second preimage game against H with a security parameter k_{s} and $x \leftarrow \{0,1\}^{l_{\mathsf{in}}}$,
- ε_4 is the advantage of some PPT adversary in the IND-WLCCA game against Γ with a security parameter k_{S}.

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 this reduction the signing oracle is simulated by encrypting $0^{l_{\mathsf{in}}}$ and simulating the

[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.

proofs using the simulation trapdoor td_s. This will clearly still lead to an extraction of x, using reusable-CRS NIZK system 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 probabilities $2^{-l_{\mathsf{dk}}}$ and $2^{-l_{\mathsf{td}_s}}$, respectively. This is why we only get security $k_{\mathrm{W}} = k + l_{\mathsf{dk}} + l_{\mathsf{td}_s}$. 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 $k_{\mathrm{S}} = k + l_{\mathsf{s}} + l_{\mathsf{dk}} + l_{\mathsf{td}_s} + l_{\mathsf{out}}$.

We note that it is primarily security against the class $\mathcal{H}_{\mathsf{ow}}$ that we are interested in, as this leakage class is independent of the signature scheme in general and the primitives used by the signature scheme in particular, like the hash function, except via the length of the public key (see the remark after Thm. 3.1). Note also the leakage class $\mathcal{H}_{\mathsf{ow}}(2^{-k_{\mathrm{S}}})$ that we prove security against is independent of the length of the secret key, so we can again obtain any desired leakage bound simply by making the secret key longer. Note in particular that if we set $l_{\mathsf{in}} = k + l_{\mathsf{s}} + l_{\mathsf{dk}} + l_{\mathsf{td}_s} + l_{\mathsf{out}} + L$, then leaking L bits from x would be an admissible leakage. Since, by the assumption on our primitives, $l_{\mathsf{s}}, l_{\mathsf{dk}}, l_{\mathsf{td}_s}$ and l_{out} do not grow with l_{in}, it thus 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 and 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 Game for our particular scheme:

Key generation: Sample $s \leftarrow \mathsf{Gen}_H$ and $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^{k_{\mathrm{S}}})$. Also, sample $(\mathsf{crs}, \mathsf{td}_s) \leftarrow S_1(1^{k_{\mathsf{td}_s}})$ and $x \leftarrow \{0,1\}^{l_{\mathsf{in}}}$. 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 $\exists x$ such that $(C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. 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 $\exists x^*$ such that $(C^* = \mathsf{Enc}^{m^*}(\mathsf{ek}, x^*) \wedge y = H_s(x^*))$, and m^* was not queried. Output 1 if $\mathcal{A}$ wins the game and output 0 otherwise.

Clearly, $\Pr[W] = \Pr[\mathsf{Game} = 1]$.

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

Key generation 0: Sample $s \leftarrow \mathsf{Gen}_H$ and $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^{k_{\mathrm{S}}})$. *Moreover, sample* $\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^{k_{\mathsf{td}_s}})$ and $x \leftarrow \{0,1\}^{l_{\mathsf{in}}}$.[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 $\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^{k_{\mathsf{td}_s}})$ 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,

$$\Pr[\mathsf{Game} = 1] - \Pr[\mathsf{Game}_0 = 1] \leq \varepsilon_0$$

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

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, yet we switch to a real CRS in the first step of our 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 $\exists$ such that x^* such that $(C^* = \mathsf{Enc}^{m^*}(\mathsf{ek}, x^*) \wedge y = H_s(x^*))$, 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 $y = H_s(x^*)$.*

The only change is that we only output 1 if the extra condition $y = H_s(x^*)$ holds. Note, however, that if $y = H_s(x^*)$ is false, then in particular $y \neq H_s(x^*)$. By the perfect decryption, this implies that $\nexists\, x^*$ such that $(C^* = \mathsf{Enc}^{m^*}(\mathsf{ek}, x^*) \wedge y = H_s(x^*))$. This implies that in Game_1 the adversary computes a proof π^* for a false statement with probability at least $\Pr[\mathsf{Game}_0 = 1] - \Pr[\mathsf{Game}_1 = 1]$. Specifically, given a CRS sampled by $\mathsf{crs} \leftarrow \mathsf{CRSGen}(1^{k_{\mathsf{td}_s}})$, Game_1 can be emulated in polynomial-time with that specific crs in vk, implying that

$$\Pr[\mathsf{Game}_0 = 1] - \Pr[\mathsf{Game}_1 = 1] = \varepsilon_1$$

where ε_1 is the probability when some PPT adversary successfully attacks the soundness property of Π with a security parameter k_{td_s}.

Let Game_2 denote the game which proceeds as Game_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 $\exists x^*$ such that $(C^* = \mathsf{Enc}^{m^*}(\mathsf{ek}, x^*) \wedge y = H_s(x^*))$, 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 Game_2, then with probability at least $\Pr[\mathsf{Game}_1 = 1] - \Pr[\mathsf{Game}_2 = 1]$ we have that $x^* \neq x$ and $y = H_s(x^*)$. Specifically, we can take a random s and a random x as input and emulate Game_2 in polynomial-time, with that specific s as key for H_s and that specific x as signing key sk, thus

$$\Pr[\mathsf{Game}_1 = 1] - \Pr[\mathsf{Game}_2 = 1] \leq \varepsilon_2$$

where ε_2 is the probability that some PPT adversary wins the second preimage game against H with a security parameter k_{s} and $x \leftarrow \{0,1\}^{l_{\mathsf{in}}}$.

Let Game_3 be the game which proceeds as follows:

Key generation 3: Sample $s \leftarrow \mathsf{Gen}_H$ and $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^{k_{\mathsf{S}}})$. Furthermore, sample $(\mathsf{crs}, \mathsf{td}_s) \leftarrow S_1(1^{k_{\mathsf{td}_s}})$ and $x \leftarrow \{0,1\}^{l_{\mathsf{in}}}$. 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 $\exists x$ such that $(C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. 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 $\exists x^*$ such that $(C^* = \mathsf{Enc}^{m^*}(\mathsf{ek}, x^*) \wedge y = H_s(x^*))$, 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 Game_2 and Game_3 is whether we give real or simulated proofs. Specifically, 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 produce the output of Game_2, respectively Game_3, in polynomial-time, thus

$$\Pr[\mathsf{Game}_2 = 1] - \Pr[\mathsf{Game}_3 = 1] \leq \varepsilon_3$$

where ε_3 is the advantage of some PPT adversary in the ZK game against Π with a security parameter k_{td_s}.

Let Game_4 be Game_3 with the following change:

Signing Oracle 4:

1. Get a message m from $\mathcal{A}$.
2. *Compute* $C = \mathsf{Enc}^m(\mathsf{ek}, 0^{l_{\mathsf{in}}})$. Using td_s, generate a simulated NIZK π proving that $\exists x$ such that $(C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. 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 $x \leftarrow \{0,1\}^{l_{\mathsf{in}}}$. 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 $(m, x, 0^{l_{\mathsf{in}}})$ to the encryption oracle and get back C. Using td_s, generate a simulated NIZK π proving that $\exists x$ such that $(C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. 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 $\exists x^*$ such that $(C^* = \mathsf{Enc}^{m^*}(\mathsf{ek}, x^*) \wedge y = H_s(x^*))$, 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 $|x| = l_{\mathsf{in}}$ 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 our construction we have that the sum of the advantages of adversaries $\mathcal{B}^1, \ldots, \mathcal{B}^u$ is an upper bound on $|\Pr[\mathsf{Game}_3 = 1] - \Pr[\mathsf{Game}_4 = 1]|$. It follows that

$$\Pr[\mathsf{Game}_3 = 1] - \Pr[\mathsf{Game}_4 = 1] \leq u\varepsilon_4$$

where ε_4 is the advantage of some PPT adversary in the IND-WLCCA game against Γ with a security parameter k_{S}.

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

Key generation 5: Sample $s \leftarrow \mathsf{Gen}_H$ and $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^{k_{\mathrm{S}}})$. Furthermore, sample $(\mathsf{crs}, \mathsf{td}_s) \leftarrow S_1(1^{k_{\mathsf{td}_s}})$. Get $x \in \{0,1\}^{l_{\mathsf{in}}}$ 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 $C = \mathsf{Enc}^m(\mathsf{ek}, 0^{l_{\mathsf{in}}})$. Using td_s, generate a simulated NIZK π proving that $\exists x$ such that $(C = \mathsf{Enc}^m(\mathsf{ek}, x) \wedge y = H_s(x))$. 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,

$$\Pr_{x\leftarrow\{0,1\}^{l_{\mathsf{in}}}}[\mathcal{B}_5(x)=x|]\geq \Pr[\mathsf{Game}_4=1].$$

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 $\mathsf{dk}'\leftarrow\{0,1\}^{l_{\mathsf{dk}}}$ and $\mathsf{td}'_s\leftarrow\{0,1\}^{l_{\mathsf{td}_s}}$.

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 $C=\mathsf{Enc}^m(\mathsf{ek},0^{l_{\mathsf{in}}})$. Using td'_s, generate a simulated NIZK proof π proving that $\exists x$ such that $(C=\mathsf{Enc}^m(\mathsf{ek},x)\wedge y=H_s(x))$. 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 $s\leftarrow\mathsf{Gen}_H(1^{k_{\mathsf{S}}})$ and $(\mathsf{ek},\mathsf{dk})\leftarrow\mathsf{KeyGen}(1^{k_{\mathsf{S}}})$. Furthermore, sample $(\mathsf{crs},\mathsf{td}_s)\leftarrow S_1(1^{k_{\mathsf{td}_s}})$ and $x\leftarrow\{0,1\}^{l_{\mathsf{in}}}$. 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

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

where dk' and td'_s are the values sampled by $\mathcal{B}_6$. This must hold since $\Pr[\mathsf{dk}'=\mathsf{dk}\wedge\mathsf{td}'_s=\mathsf{td}_s]=2^{-l_{\mathsf{dk}}-l_{\mathsf{td}_s}}$, thus either

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

or

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

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

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

Combining our inequalities so far, we get that

$$\Pr[W]\leq 2^{l_{\mathsf{dk}}+l_{\mathsf{td}_s}-k_{\mathrm{W}}}+\sum_{i=0}^{3}\varepsilon_i+u\varepsilon_4=2^{-k}+\sum_{i=0}^{3}\varepsilon_i+u\varepsilon_4.$$

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 $\mathsf{dk}'\leftarrow\{0,1\}^{l_{\mathsf{dk}}}$ and $\mathsf{td}'_s\leftarrow\{0,1\}^{l_{\mathsf{td}_s}}$, and let ek' be the public key corresponding to dk' and let crs' be the CRS corresponding to td'_s.[4] Sample $r\leftarrow\{0,1\}^{l_{\mathsf{s}}}$ and let $s=\mathsf{Gen}_H(1^{k_{\mathsf{s}}};r)$, and sample $y'\leftarrow\{0,1\}^{l_{\mathsf{out}}}$. Let $\mathsf{vk}'=(y',s',\mathsf{ek}',\mathsf{crs}')$.

[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.

Leakage 7: Give vk' to $\mathcal{A}$ along with a.

Signing Oracle 7:

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

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

So,

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

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

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

Combining our inequalities so far, we get that

$$\Pr[W] \leq 2^{l_{\mathsf{s}}+l_{\mathsf{dk}}+l_{\mathsf{td}_s}+l_{\mathsf{out}}-k_{\mathrm{S}}} + \sum_{i=0}^{3}\varepsilon_i + u\varepsilon_4 = 2^{-k} + \sum_{i=0}^{3}\varepsilon_i + u\varepsilon_4 \ .$$

■

Our concrete instantiation has all the needed properties except that the length of the hash function key s depends on the input length l_{in}. This, however, can be handled generically as follows.

Lemma 3.1 *If there exists an ε-secure family of second preimage 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 preimage 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}, for $k_0 = \mathrm{poly}(k)$ the seed length of PRG with security parameter k.*

Proof: Let $\mathsf{Gen}'_H(1^k; r \in \{0,1\}^{k_0}) = \mathsf{Gen}_H(1^k; \mathsf{PRG}(r))$. It is first 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. Next, let

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

and let

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

Then, consider the PPT algorithm $\mathcal{B}(s)$ which samples $x \leftarrow \{0,1\}^{l_{\mathsf{in}}}$ and then invokes $x^* \leftarrow \mathcal{A}(s, x)$, and finally outputs 1 iff $H_s(x^*) = H_s(x)$. Fix $\varepsilon' = \Pr[\mathcal{B}(\mathsf{Gen}_H(\mathsf{PRG}(r \leftarrow \{0,1\}^{k_0}))) = 1]$ and $\varepsilon = \Pr[\mathcal{B}(\mathsf{Gen}_H(r \leftarrow \{0,1\}^*)) = 1]$. Clearly, by the PRG being a δ-pseudo-random generator it follows that $|\varepsilon' - \varepsilon| \leq \delta$. This concludes the proof since it implies that H' is second preimage resistance with a short key s. ■

Remark. We note that 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 3.1. 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 our proof), circumventing the problem of having to guess s. We would then get security when considering the class $\mathcal{H}_{\mathsf{ow}}(\ell(k))$ for $k_{\mathrm{S}} = k + l_{\mathsf{dk}} + l_{\mathsf{td}_s} + l_{\mathsf{out}}$.

4 Applications: Auxiliary Input Secure Identification Schemes

In an identification scheme ID a prover attempts to prove its identity to a verifier. Specifically, it allows a prover to prove to the verifier that it possesses some secret information without revealing anything about it. Identification schemes that tolerate leakage in the bounded-retrieval model were already presented in [ADW09]. In this section we define two notions of identification schemes with security in the presence of auxiliary input, and present two constructions with security that meets these notions. More formally, for a security parameter k, an identification scheme ID consists of three PPT algorithms $\mathsf{ID} = (\mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ defined as follows:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^k)$: 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 its secret key sk. The verifier V outputs either 1 for accept or 0 for reject.

We require that ID is *complete* in the sense that an interaction with an honest prover will always be accepted by the verifier. Passive security of an identification scheme ID considers a polynomial-time adversary $\mathcal{A}$ that takes as input the public key pk and observes an arbitrary number of runs of the protocol. After this phase is completed, $\mathcal{A}$ tries to impersonate $\mathsf{P}(\mathsf{pk}, \mathsf{sk})$ by engaging in an interaction with $\mathsf{V}(\mathsf{pk})$. An identification scheme ID is *passive secure* if every polynomial-time adversary $\mathcal{A}$ impersonating the prover, only succeeds with at most negligible probability. We extend this definition 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 $h \in \mathcal{H}$. More formally, consider the following definition

Definition 4.1 (Secure identification schemes under auxiliary input attacks) *An identification scheme* $\mathsf{ID} = (\mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ *is* passively secure under impersonation attacks w.r.t. to auxiliary inputs *(*$\mathsf{IDAUX}_{\mathsf{vkow}}$*) 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 the protocol* $(\mathsf{P}(\mathsf{pk}, \mathsf{sk}), \mathsf{V}(\mathsf{pk}))$ *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 explained 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 $\ell(k) - \mathsf{IDAUX}_{\mathsf{vkow}}$ implies security when the leakage function is picked from the class $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$. Where the later identification schemes will be secure only in presence of leakage functions that are hard to invert given also the public key of the scheme.

We stress that our construction is insecure in the presence of active attacks, where the adversary gets to determine the challenges for the prover, since our construction is only secure for random messages picked by the challenger in the security game for signature schemes. Whereas in the active scenario, the adversary picks the messages to be signed by itself. This implies that we cannot reduce an active attack into the security of our signature scheme from Section 3.2.

4.1 Non-Interactive Identification from Signature Schemes

In this section, we present constructions for both auxiliary input notions of secure identification schemes. More specifically, it is a well known fact that non-interactive identification can be easily constructed from signature schemes. We demonstrate below that the same argument holds also when considering a 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 a class of functions $\mathcal{H}$ (cf. Definition 2.8). Then, the following identification scheme $\mathsf{ID}_1 = (\mathsf{KeyGen}_1, \mathsf{P}_1, \mathsf{V}_1)$ is passive secure against 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 4.2 *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 2.8, then ID_1 is $\ell(k) - \mathsf{IDAUX}_{\mathsf{ow}}$ (resp. $\ell(k) - \mathsf{IDAUX}_{\mathsf{vkow}}$) secure.*

Proof: Let Σ be random messages unforgeable signature scheme against random message attacks for $\mathcal{H}_{\mathsf{ow}}(\ell(k))$ (resp. for $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$) and let ID_1 be the identification scheme described above. Assume, by contradiction, that ID_1 is not passive secure against impersonation attacks for $\mathcal{H}_{\mathsf{ow}}(\ell(k))$ (resp. for $\mathcal{H}_{\mathsf{vkow}}(\ell(k))$). This implies the existence of a PPT adversary $\mathcal{A}$ that wins the impersonating game with a non-negligible probability $p(k)$ for infinitely many k's. We use $\mathcal{A}$ to break security of Σ. Consider the following adversary $\mathcal{B}$ playing against a challenger in the RU-RMA game for signature schemes.

1. $\mathcal{B}$ receives a verification key pk together with a leakage function $H(\mathsf{pk}, \mathsf{sk})$.
2. $\mathcal{B}$ invokes adversary $\mathcal{A}$ with input $(\mathsf{pk}, H(\mathsf{pk}, \mathsf{sk}))$.
3. When $\mathcal{A}$ wishes to observe an interaction of the identification protocol, $\mathcal{B}$ first asks its oracle for a signature, receiving back (m, σ), where $m \leftarrow \mathcal{M}$ and $\sigma = \mathsf{Sig}(\mathsf{sk}, m)$. $\mathcal{B}$ then proceeds in the simulation of the protocol by setting the random challenge of the verifier $c = m$, and then simulating the prover by sending back σ.

4. Whenever $\mathcal{A}$ is ready to impersonate the prover, $\mathcal{B}$ asks its challenger for a random message m^* to be signed, which it then forwards to $\mathcal{A}$.

5. Finally, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs.

Note first that $\mathcal{B}$ perfectly simulates the identification protocol's execution and that its overall running time is polynomial. Moreover, $\mathcal{B}$ wins the game whenever $\mathcal{A}$ impersonates correctly the prover. In other words:

$$\Pr[\mathcal{B} \text{ wins}] \geq \Pr[\mathcal{A} \text{ wins}] \geq p(k).$$

This is a contradiction to the security of Σ and thus concludes the proof. ■

5 Security under the K-Linear Assumption

In this section we demonstrate how to instantiate our scheme from Section 3.3 with concrete primitives that yield an implementation of a EU-CMAA signature scheme. The hardness of our instantiated scheme follows from the K-linear assumption defined below (which also 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 5.1 (The 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 $K = 2$. We further note that the hardness of K-linear implies the hardness of the discrete logarithm problem defined as follows.

Definition 5.2 (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*

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

where $(p, \mathbb{G}, g) \leftarrow \mathcal{G}(1^k)$ and $x \leftarrow \mathbb{Z}_p$.

Definition 5.3 (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 for $\mathbb{G}$ 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 $H = (\mathsf{Gen}_H, H_s)$. Let $\mathbb{G}$ be a group of prime order p. Define first $g_1, \ldots, g_\ell \leftarrow \mathsf{Gen}_H(1^k)$ such that $g_1, \ldots, g_\ell$ are generators for $\mathbb{G}$ and fix the public key $s = (g_1, \ldots, g_\ell)$. Then, for input $x \leftarrow \mathbb{Z}_p^\ell$ define $H_s(x) := \prod_{i=1}^{\ell} g_i^{x_i}$. It is simple to verify that 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 $(\log_g g_1, \ldots, \log_g g_{\ell-1}, x, y)$. As shown below, second pre-image resilience holds even for a small input domain, such as bits.

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. Definition 2.3) 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) for 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. Notice that we use bold fonts to denote vectors. Formally, for the 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 $H_{\mathsf{CL}} : \{0,1\}^* \to \mathbb{Z}_p$ is a collision resistance hash function, and matrix $\mathbf{A}$ defined by

$$\mathbf{A} = \begin{pmatrix} g_0 & g_1 & 1 \\ g_0 & 1 & g_2 \end{pmatrix}$$

define the following encryption scheme.

Key generation, KeyGen. Choose 3 random vectors $\mathsf{u}, \mathsf{v}, \mathsf{w} \leftarrow \mathbb{Z}_p^3$. Compute the following:

$$\mathbf{d} = \mathbf{A} \cdot \mathsf{v},\ \mathbf{e} = \mathbf{A} \cdot \mathsf{w},\ \mathbf{h} = \mathbf{A} \cdot \mathsf{u}.$$

Notice that $\mathbf{d}, \mathbf{e}, \mathbf{h} \in \mathbb{G}^2$. 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 $\mathbf{y} = \mathsf{r}^\top \cdot \mathbf{A} \in \mathbb{G}^3$. Set

$$a := h_1^{r_1} \cdot h_2^{r_2},\ \ z = a \cdot m,\ \ c = (d_1(e_1^t))^{r_1} \cdot (d_2(e_2^t))^{r_2},$$

where $t = H_{\mathsf{CL}}(\mathbf{y}, z, L)$, $\mathbf{d} = (d_1, d_2)$, $\mathbf{e} = (e_1, e_2)$, $\mathsf{r} = (r_1, r_2)$ and $\mathbf{h} = (h_1, h_2)$. Output $C = (\mathbf{y}, z, c)$.

Decryption, Dec. To decrypt ciphertext C under label L, parse $C = (\mathbf{y}, z, c)$. Compute

$$t = H_{\mathsf{CL}}(\mathbf{y}, z, L),\ \ \tilde{c} = \mathbf{y}^\top \cdot (\mathsf{v} + t\mathsf{w}).$$

If $\tilde{c} = c$, then output $z/(\mathbf{y}^\top \cdot \mathsf{u})$. Else output $\perp$.

3. NIZK ARGUMENT FOR NP. We consider the NIZK argument of Groth-Sahai [GS08] which shows how to prove in zero-knowledge under the 2-linear assumption that a linear system has a solution (our notations here follow from [DHLAW10a] as well). Let $\mathbf{B} \in \mathrm{M}_{M \times N}(\mathbb{G})$ be a matrix whose rows are

$\mathbf{b}_i = (\mathbf{b}_{i,1}, \ldots, \mathbf{b}_{i,N})$ for $i = 1, \ldots, M$. Let $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_M)$ be a target vector in $\mathbb{G}^M$. We say that the system $(\mathbf{B}, \mathbf{c})$ is satisfiable if there exists a vector $\mathsf{u} = (\mathsf{u}_1, \ldots, \mathsf{u}_N) \in \mathbb{Z}_p^N$ 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 every $i \in [1, \ldots, M]$.

Another type of proof we adapt from [GS08] is for proving that an encrypted plaintext is a bit. Namely, when encrypting g^x the prover proves that the quadratic equation $x(1-x)$ equals zero. This proof ensures that a dishonest signer does not encrypt arbitrary plaintexts that are multiplied into the right hashed value, but do not enable extraction (since we cannot efficiently compute the discrete logarithm for arbitrary values in $\mathbb{G}$).

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 views 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 ciphertexts with the same label m. It is easy to see that IND-WLCCA is maintained under such block-wise parallel encryption with respect to the same label. (See Theorem A.1 in the appendix for a formal proof). The signer then computes a NIZK for proving that these ciphertexts encrypt bits and that they are consistent with the hashed value $H(x)$ taken from the public key.

We recall first that second pre-image resilience still holds even when evaluated on individual bits. Specifically, finding a collision x, x' implies that these values differ in at least a single bit. This means that $\prod_{i=1}^{\ell} g_i^{x_i} = \prod_{i=1}^{\ell} g_i^{x'_i}$ induces 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 rather than group elements is necessary in order to extract x from the forgery given by the adversary.

It is left to show how the NIZK proofs are defined. We observe first that for a ciphertext $c = (c_1, c_2, c_3)$ generated by Enc, elements c_1 and c_2 are component-wise multiplicatively homomorphic (where the third element is needed to verify consistency). Thus, given a ciphertext $c = (c_1, c_2, c_3)$ encrypting g^x one can generate a (partial) encryption of g^{1-x} using the homomorphic property of our PKE and prove that the product of the underlying plaintexts equals zero. Specifically, the signer proves that the product of x and $1-x$ is zero which implies that x must be a bit. In addition, it is possible to efficiently compute a (partial) encryption of $H(x)$ from encryptions of individual bits of x, denoted by $c_x = ((c_{1_1}, c_{1_2}, c_{1_3}), \ldots, (c_{\ell_1}, c_{\ell_2}, c_{\ell_3}))$, by computing the following products

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

This implies that if c_x is correctly computed than the following relation holds

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

Note 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. Finally, in order to impose consistency between c_x and $H(x)$ 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 implies another linear equation and concludes the description of the signature.

As for the concrete parameters, for $k = \log p$ 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 can be brought down to $l_1 = 2k$ using Lemma 3.1, as it is trivial to build a pseudo-random generator $\mathbb{G}^2 \to \mathbb{G}^\ell$ using the 2-linear assumption. Therefore, by 3.2 we obtain existential unforgeability with $k_{\mathrm{W}} = k + l_2 + l_3 = 6k$. If we consider the class $\mathcal{H}_{\mathsf{ow}}(\ell(k))$ we obtain 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.

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

[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.

[BS97] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In *CRYPTO*, pages 513–525, 1997.

[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.

[DF12] Stefan Dziembowski and Sebastian Faust. Leakage-resilient circuits without computational assumptions. In Ronald Cramer, editor, *TCC*, volume 7194 of *Lecture Notes in Computer Science*, pages 230–247. Springer, 2012.

[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.

[GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In *CHES*, number Generators, pages 251–261, 2001.

[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.

[HLAWW13] Carmit Hazay, Adriana López-Alt, Hoeteck Wee, and Daniel Wichs. Leakage-resilient cryptography from minimal assumptions. In *EUROCRYPT*, pages 160–176, 2013.

[HSH+09] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest we remember: cold-boot attacks on encryption keys. *Commun. ACM*, 52(5):91–98, 2009.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *CRYPTO*, pages 388–397, 1999.

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO*, pages 104–113, 1996.

[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.

[QS01] Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *E-smart*, pages 200–210, 2001.

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

[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.

[SMY09] François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *EUROCRYPT*, pages 443–461, 2009.

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

[YCZY12] Tsz Hon Yuen, Sherman S. M. Chow, Ye Zhang, and Siu-Ming Yiu. Identity-based encryption resilient to continual auxiliary leakage. In *EUROCRYPT*, pages 117–134, 2012.

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. Formally, let $\Gamma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an LPKE scheme with message space M and let $\Delta = (\mathsf{KeyGen}, \mathsf{Encl}, \mathsf{Decl})$ be LPKE with message space M^n such that

Encryption (long message): $\mathsf{Encl}^L(\mathsf{ek}, (x_1, \ldots, x_n)) = (\mathsf{Enc}^{(L,i)}(\mathsf{ek}, x_i))_{i=1}^n$.

Decryption (long message): $\mathsf{Decl}^L(\mathsf{dk}, (c_1, \ldots, c_n)) = (\mathsf{Dec}^{(L,i)}(\mathsf{dk}, c_i))_{i=1}^n$.

Then we prove the following theorem,

Theorem A.1 *If Γ is IND-WLCCA (cf, Definition 2.3), then Δ 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 Γ as follows:

1. Get ek as input and forward it 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 an adversary $\mathcal{B}$ which is admissible. This means an adversary that does not ask to decrypt any ciphertext with the label L (used for computing the challenge). This implies that $\mathcal{A}$ will not ask to decrypt any ciphertext with this label either. Now, since the challenge ciphertext of $\mathcal{A}$ has a label L, it follows that $\mathcal{A}$ is admissible as well.

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 output distribution of $\mathcal{B}$ when i sampled by $\mathcal{A}$ happens to be $i = i_0$ and $b = b_0$. Recall that when $b = 0$, then the message vector encrypted by $\mathcal{A}$ is $(x_{0,1}, \ldots, x_{0,i-1}, x_{0,i}, x_{1,i+1}, \ldots, x_{1,n})$. Whereas when $b = 1$, then the message vector 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 $\Pr[b'_{i,0} = 1] - \Pr[b'_{i,1} = 1] \leq \mathsf{negl}$. Then 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 $\Pr[b'_{1,1} = 1] - \Pr[b'_{1,0} = 1] \leq \mathsf{negl}$. Thus, $\Pr[b'_{n,1} = 1] - \Pr[b'_{1,0} = 1] \leq 2 \cdot \mathsf{negl} \leq \mathsf{negl}$. Let b'_{b_0} be the output distribution of $\mathcal{B}$ in the IND-WLCCA game against Γ when b sampled in the IND-WLCCA game is $b = b_0$. By construction $b'_0 = b'_{1,0}$ and $b'_1 = b'_{n,1}$. This implies that $\Pr[b'_1 = 1] - \Pr[b'_0 = 1] \leq \mathsf{negl}$, as desired. ■