Efficient Signatures with Tight Real World Security in the Random-Oracle Model

Christoph Bader

Horst Görtz Institute, Ruhr-University Bochum, Germany
christoph.bader@rub.de

Abstract. Security for digital signature schemes is most commonly analyzed in an ideal single user setting where the attacker is provided only with a single public key. However, when digital signature schemes are deployed in practice they are often used by many users, each having its own public key, e.g., in authenticated key exchange (AKE) protocols. Common security models for AKE model real world capabilities of an adversary by allowing it (among others) to corrupt secret user keys. For digital signatures it is well known that security in the idealized single user setting implies security in this stronger and more realistic multi user setting with corruptions. However, the security reduction loses a factor which is linear in the number of users. It is not clear how to avoid this loss in general.

In this paper we propose an efficient signature scheme whose security reduction in the above setting is tight. The security reduction loses a factor of about 2. When 80 bits of security are required our signatures are of size roughly 2700 bits.

Keywords: Tight security, digital signatures, Groth-Sahai proofs, Katz-Wang technique, random-oracle heuristic.

1 Introduction

When a new cryptographic scheme is proposed, nowadays the construction comes along with a proof of security. Most commonly, the proof describes an efficient algorithm, the *reduction*, that turns any successful attacker against the scheme (with respect to the considered security notion) into another efficient algorithm that breaks a supposed to be hard problem. The quality of a reduction R is measured in terms of its success probability ϵ_{R} relative to its running time t_{R}. Ideally we have $\frac{\epsilon_{\mathsf{R}}}{t_{\mathsf{R}}} = \mathcal{O}(\frac{\epsilon_{\mathsf{F}}}{t_{\mathsf{F}}})$ where ϵ_{F} and t_{F} denote the success probability and the running time of the forger. In this case the reduction is said to be *tight* and the cryptographic scheme is said to have tight security. Tight reductions are a desirable goal since the quality of a reduction influences the size of the system parameters when they are selected in a theoretically sound way, cf. table 1. There exist implementations of many cryptographic primitives that come along with an (almost) tight reduction in the standard or the random oracle model, e.g., for digital signatures in the single user setting [8,21,9,28,20], for public key encryption in the multi user setting [5,17] and for AKE [3].

Digital Signatures in the Multi User setting. The standard security notion for digital signatures (in the single user setting) is existential unforgeability under chosen message attacks (EUF-CMA-security) [15]. EUF-CMA-security was later extended to the multi user setting *without* corruptions [25]. Recently, [3] introduced the notion of *existential unforgeability under chosen message attacks in the multi user setting* with *adaptive corruptions* (MU-EUF-CMA$^{\mathsf{Corr}}$-security). Here the attacker is considered successful if it manages to produce a signature for a message m (that was not signed before with respect to the target public key) that verifies under an uncorrupted public key (the target public key). While tightness in the single user setting is mostly considered with respect to the number μ of sign queries issued by the attacker, in the multi user setting tightness is additionally considered relative to the number ℓ of public keys the adversary has access to and that it may corrupt. Hence, for digital signatures in the multi user setting there are two dimensions to consider tightness in.

It is well known [25,3] that standard EUF-CMA security (i.e., $\ell = 1$) implies MU-EUF-CMA$^{\mathsf{Corr}}$-security. However, the generic reduction loses a factor of ℓ, i.e., $\frac{\epsilon_{\mathsf{R}}}{t_R} = \mathcal{O}(\ell \cdot \frac{\epsilon_{\mathsf{F}}}{t_F})$ and it is not clear how to avoid this loss in general. On the bright side this means that the proofs from [9,28,20] give rise to digital signature schemes in the multi user setting with corruptions that come along with a proof that only depends (linearly) on ℓ (the number of public keys) but is independent of μ (the number of sign queries issued by the attacker). Recently, standard model schemes that come along with a reduction that is independent of μ *and* ℓ were proposed [3]. However, as the authors remark due to its large signature size the full tight scheme from [3] is rather a feasability result. While the almost tight scheme from [3] supports very short signatures it has public parameters that are linear in the length of messages.

We stress that common security models for *authenticated* key exchange (AKE) or channel establishment (ACCE), e.g. [7,11,19], allow the adversary to corrupt long-term secret keys which often are secret keys of a signature scheme, e.g., in ephemeral Diffie-Hellman Ciphersuites of the TLS-Handshake [14] or when compilers lift a passively secure protocol to meet stronger security notions [6,22,18,23]. Therefore, the MU-EUF-CMA$^{\mathsf{Corr}}$ security notion is implicitly widely used in practice. However, the security proofs for most schemes apply the "polynomial equivalence between EUF-CMA and MU-EUF-CMA$^{\mathsf{Corr}}$ security" argument which incurs a loss of ℓ for the reduction and requires larger parameters when the scheme is implemented in practice. Therefore an efficient signature scheme, i.e., small signatures *and* public parameters, that comes along with a tight MU-EUF-CMA$^{\mathsf{Corr}}$ security reduction is a desirable goal with practical applications. In particular, plugging in a tightly MU-EUF-CMA$^{\mathsf{Corr}}$-secure signature scheme into the tightness preserving compiler from [3] leads to a tightly secure authenticated key exchange protocol the efficiency of which is roughly determined by the efficiency of the signature scheme.

Our Contribution. In this paper we propose a signature scheme that tightly satisfies MU-EUF-CMA$^{\mathsf{Corr}}$ security, i.e., the running time and the success probability of the reduction are roughly the same as the running time and the success

	$\ell = 1$		$\ell = 2^{16}$		$\ell = 2^{45}$		Loss	Assumption
	$\lvert vk \rvert$	$\lvert \sigma \rvert$	$\lvert vk \rvert$	$\lvert \sigma \rvert$	$\lvert vk \rvert$	$\lvert \sigma \rvert$		
ECDSA [29,1,30]	≈ 280	≈ 560	≈ 312	≈ 624	≈ 370	≈ 740	$\mathcal{O}(q_h \ell)$	DLOG
BLS [10]	≈ 2000	≈ 220	≈ 3000	≈ 256	> 4000	≈ 310	$\mathcal{O}(\mu \ell)$	CDH
RSA PSS [8,12]	> 1024	1024	> 1350	≈ 1350	> 3000	≈ 3000	$\mathcal{O}(\ell)$	RSA
Ours	1024	2688	1024	2688	1024	2688	$\mathcal{O}(1)$	SXDH

Table 1: Comparison between our scheme and random oracle signature schemes from the literature. We compare public key size and signature size in bits when parameters are selected to obtain 80 bits of security in a theoretically sound way (i.e., parameter selection considers the security loss) following NIST recommendations [4]. Following [8] we assume $\mu = 2^{30}$ sign-queries and $q_h = 2^{60}$ hash-queries per public key. The BLS and ECDSA schemes as well as our scheme do also require common public parameters. These are omitted in our comparison since they have to be stored only once by each user.

probability of the adversary (and in particular independent of μ and ℓ except for a negligible fraction). The security reduction loses roughly a factor of 2. The scheme works over asymmetric bilinear groups $\mathbb{G} = (\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T)$ equipped with an efficiently computable pairing $e : \mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_T$. Public parameters contain a description of the group, one additional element from $\mathcal{G}_1$, two additional elements from $\mathcal{G}_2$ and the description of a Hash-function. A public key is a single group element from $\mathcal{G}_2$ and signatures live in $\mathcal{G}_1^4 \times \mathcal{G}_2^2$. Table 1 compares our signature scheme to random oracle signature schemes from the literature. We observe that if the number of users is 2^{16} then the signature size of our scheme is roughly twice the size of an RSA PSS signature and 10 times the size of a BLS signature. Our scheme outperforms RSA PSS in both, public key size and signature size, if the number of public keys is about 2^{45} which is a very large number. However, even in this case, BLS and ECDSA signatures are shorter than our signatures. Therefore, for most of today's practical applications our scheme is no better than known solutions. However, due to the loss of the generic reduction (see above) and to some problems that occur by natural approaches (see end of this section) we find it interesting in its own right to construct a signature scheme with tight MU-EUF-CMA$^{\mathsf{Corr}}$ security.

Technical Approach. When designing an MU-EUF-CMA$^{\mathsf{Corr}}$-secure signature scheme with tight reduction we are faced with the following problem: On the one hand we need to be able to reveal the secret key to any public key (note that guessing the target public key would cause a loss of ℓ) and on the other hand we must be able to extract a solution to a hard problem from (almost) any forgery that is output by the adversary. That is, we must be able to extract a solution from a forgery even if we know the secret key corresponding to the target public key. To face this problem, we apply non-interactive proof systems that provide two computationally indistinguishable modes of common reference strings (CRS), perfectly *binding* ones and perfectly *hiding* ones. A perfectly binding CRS allows to extract knowledge from a given proof while a perfectly hiding

CRS does not. A signature will roughly be a proof (using a suitable proof system) that the signer 'knows' a one time signature. Now, to extract a solution from an adversarially generated signature the proof output by the adversary needs to be binding. Note that we do not know the target public key and message up front. At the same time, to hide all critical information from the adversary all proofs output by the reduction need to be hiding.

To achieve this we apply the random oracle in a way similar to Katz-Wang [21] to the (standard model, DLIN-based) linearly homomorphic signature scheme from [24] converted to the SXDH-setting. Namely, public parameters contain part of a Groth-Sahai CRS [16]. To sign a message m, a bit b is sampled uniformly at random. The message is hashed together with b and the public key of the signer to complete the CRS. Finally, using the secret key, a one time signature over m is computed and correctness of the computed signature is proved with respect to the CRS. During the security reduction the random oracle will be programmed such that for each pair of message m and public key vk one out of two possible CRS (recall that m and vk are hashed together with b) is perfectly hiding and the other one is perfectly binding. Both are indistinguishable under a computational assumption. Now, the reduction will make all proofs on a hiding CRS (and thus leak no information about sk) and with high probability the adversary will output a forgery on a binding CRS from which we can extract a solution to a hard problem with overwhelming probability.

A note on schemes from OR proofs. We note that it might look heavy to use the random oracle heuristic in combination with pairings at all and in particular to additionally use Groth-Sahai proofs. Probably the most natural way to construct a tightly secure scheme in the ROM would be to apply OR proofs as introduced in [13] to Fiat-Shamir like signature schemes that have a tight reduction, e.g. [21]. Similar to the fully tight construction from [3] and following the Naor-Yung paradigm [27], a public key in such MU-EUF-CMA$^{\mathsf{Corr}}$-secure scheme would consist of two public keys (vk_0, vk_1) of the underlying signature scheme whereas the secret key would consist only of one of the corresponding secret keys, sk_δ. A signature on message m would be a witness indistinguishable OR proof that the signer 'knows' a signature on m that validates under vk_0 or vk_1. The OR proofs from [13] provide *perfect* witness indistinguishability. Therefore it remains information theoretically hidden from the view of the adversary which secret key is known by the reduction. Unfortunately perfect witness indistinguishability makes the reduction fail to actually extract knowledge from the forgery output by the adversary.

If we are to apply pairings we can resort to Groth-Sahai proofs [16] and could apply a similar technique. However, in this case we need to prove satisfiability of a set of quadratic equations which makes the proofs expensive, i.e., large. Since we are interested in efficient schemes we do not apply this technique. We note however that this technique works even in the standard model [3]. However, it leads to rather long signatures.

2 Preliminaries

Notation. By $[n]$ we denote the set $[n] := \{1, 2, \ldots, n\}$. If A is a set then by $a \leftarrow^{\$} A$ we denote the action of sampling a uniformly from A. If A is an algorithm then $a \leftarrow A(x)$ denotes that A outputs a when run on input x with fresh uniformly random coins. By PPT we will abbreviate probabilistic polynomial time. If an algorithm A has black-box access to an algorithm $\mathcal{O}$, we will write $A^{\mathcal{O}}$.

By $\mathbb{G} = (e, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T, g_1, g_2, p)$ we denote the description of an asymmetric bilinear group. That is, $e : \mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_T$ is a non-degenerate bilinear map, g_b is a generator of $\mathcal{G}_b$ and $|\mathcal{G}_1| = |\mathcal{G}_2| = |\mathcal{G}_T| = p$ where p is prime. It is well known that there is a PPT algorithm that on input 1^κ returns $\mathbb{G}$ such that $2^\kappa < p \leq 2^{\kappa+1}$. We denote this algorithm by $\mathsf{GEN.asym}(1^\kappa)$. Throughout the paper we reasonably assume the non-existence of efficiently computable homomorphisms between $\mathcal{G}_1$ and $\mathcal{G}_2$. Given elements $h \in \mathcal{G}_2$ and $\vec{g} = (g, k) \in \mathcal{G}_1^2$ we denote by $E(\vec{g}, h)$ the vector $(e(g, h), e(k, h))$.

Complexity Assumptions. Let in the sequel be $b \in \{1, 2\}$. Given $g, h \in \mathcal{G}_b^2$ we denote by $\mathsf{DDH}_b(g, h)$ the set $\mathsf{DDH}_b(g, h) := \left\{(\hat{g}, \hat{h}) \in \mathcal{G}_b^2 : \mathsf{log}_g(\hat{g}) = \mathsf{log}_h(\hat{h})\right\}$.

Definition 1. *Let* $\mathbb{G} = (e, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T, g_1, g_2, p) \leftarrow^{\$} \mathsf{GEN.asym}(1^\kappa)$*. We say that an adversary* (t, ϵ)*-breaks the* external Diffie-Hellman *assumption in* $\mathcal{G}_b$ *(*XDH_b *assumption) if it runs in time* t *and*

$$\Big| \Pr\left[\mathcal{A}(\mathbb{G}, g, h, \hat{g}, \hat{h}) = 1 : (g, h) \leftarrow^{\$} \mathcal{G}_b^2 \wedge (\hat{g}, \hat{h}) \leftarrow^{\$} \mathsf{DDH}(g, h)\right] \\ - \Pr\left[\mathcal{A}(\mathbb{G}, g, h, \hat{g}, \hat{h}) = 1 : (g, h) \leftarrow^{\$} \mathcal{G}_b^2 \wedge (\hat{g}, \hat{h}) \leftarrow^{\$} \mathcal{G}_b^2\right] \Big| \geq \epsilon$$

where the probability is over the random choices of $g, h, \hat{g}, \hat{h}$ *and the random coins of* $\mathcal{A}$*.*

We say that an adversary (t, ϵ)*-breaks the* symmetric *external Diffie-Hellman assumption in* $\mathbb{G}$ *if it* (t, ϵ)*-breaks the* XDH_1 *or* XDH_2 *assumption.*

A given instance of the XDH_b problem is efficiently re-randomizable [26,5]. That is, there is an efficient algorithm that, on input $(g, h, \hat{g}, \hat{h}, 1^q)$, outputs q tuples $(g_i, h_i), i \in [q]$ such that

$$(g_i, h_i) \leftarrow^{\$} \mathsf{DDH}(g, h) \text{ if } (\hat{g}, \hat{h}) \in \mathsf{DDH}(g, h) \\ (g_i, h_i) \leftarrow^{\$} \mathcal{G}_b^2 \text{ if } (\hat{g}, \hat{h}) \notin \mathsf{DDH}(g, h)\,.$$

Definition 2. *Let* $\mathbb{G} = (e, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T, g_1, g_2, p) \leftarrow^{\$} \mathsf{GEN.asym}(1^\kappa)$ *and* $(g_z, g_r) \leftarrow^{\$} \mathcal{G}_2^2$*. We say that an adversary* (t, ϵ)*-breaks the* double pairing assumption *in* $\mathcal{G}_2$ *(*DP_2 *assumption) if it runs in time* t *and*

$$\Pr\left[(z, r) \neq (1, 1) \wedge e(z, g_z) \cdot e(r, g_r) = 1 : (z, r) \leftarrow \mathcal{A}(\mathbb{G}, g_z, g_r)\right] \geq \epsilon$$

where the probability is over the random choices of g_z *and* g_r *and the random coins of* $\mathcal{A}$*.*

We define the DP_1 *assumption analogously.*

Lemma 1 ([2]). *For any attacker $\mathcal{A}$ that $(t_{\mathsf{DP_b}}, \epsilon_{\mathsf{DP_b}})$-breaks the* DP *assumption in $\mathcal{G}_b$ (where $b \in \{1, 2\}$) there exists an attacker $\mathcal{B}$ that $(t_{\mathsf{XDH}}, \epsilon_{\mathsf{XDH}})$-breaks the* XDH *assumtion in $\mathcal{G}_b$ where $t_{\mathsf{DP_b}} \approx t_{\mathsf{XDH}}$ and $\epsilon_{\mathsf{XDH}} \geq \epsilon_{\mathsf{DP_b}}$.*

Proof. Let wlog $b = 2$. Algorithm $\mathcal{B}$, given an XDH_2 instance $(\mathbb{G}, g, h, \hat{g}, \hat{h})$, runs $\mathcal{A}$ as a subroutine on input $(\mathbb{G}, g, \hat{g})$. When $\mathcal{A}$ outputs (z, r) such that $e(z, g) \cdot e(r, \hat{g}) = 1$ we know that $\mathsf{log}_z(r) = -\mathsf{log}_{\hat{g}}(g)$ and thus $e(z, h) \cdot e(r, \hat{h}) = 1 \Leftrightarrow (\hat{g}, \hat{h}) \in \mathsf{DDH}(g, h)$.

3 Digital Signature Schemes in the Multi User Setting

Syntax. A digital signature scheme $\mathsf{SIG} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vfy})$ is a four-tuple of PPT algorithms.

Public Parameters. The parameter generation algorithm $\mathit{\Pi} \xleftarrow{\$} \mathsf{Setup}(1^\kappa)$ on input 1^κ returns public parameters. We silently assume that 1^κ is contained in $\mathit{\Pi}$. We note that while Setup often is omitted in the single user setting it is convenient to define it in the multi user setting. If not explicitly required, it just outputs 1^κ.

Key Generation. The key generation algorithm when input $\mathit{\Pi}$ outputs a key pair, $(vk, sk) \xleftarrow{\$} \mathsf{Gen}(\mathit{\Pi})$. Even if not explicitly stated we assume that vk contains at least $\mathit{\Pi}$ and that sk contains vk.

Signature Generation. The signature generation algorithm, given a secret key sk and message m, outputs a signature σ on that message. That is, it returns $\sigma \xleftarrow{\$} \mathsf{Sign}(sk, m)$.

Verification. The verification algorithm accepts or rejects a signature over a message with respect to a given public key, $\mathsf{Vfy}(vk, m, \sigma) \in \{0, 1\}$.

For correctness we require that for all κ, all $\mathit{\Pi} \xleftarrow{\$} \mathsf{Gen}(1^\kappa)$, all $(vk, sk) \xleftarrow{\$} \mathsf{Gen}(\mathit{\Pi})$ and any message m that

$$\Pr\left[\mathsf{Vfy}(vk, m, \sigma) = 1 : \sigma \xleftarrow{\$} \mathsf{Sign}(sk, m)\right] = 1 \,.$$

Security Notion. Consider the following security experiment that is played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ and that is parametrized by μ, the number of overall sign queries the adversary may issue and ℓ the number of public keys the adversary has access to and that it may corrupt.

1. On input 1^κ the challenger runs $\mathit{\Pi} \xleftarrow{\$} \mathsf{Setup}(1^\kappa)$ and samples $(vk_i, sk_i) \xleftarrow{\$} \mathsf{Gen}(\mathit{\Pi}), i \in [\ell]$. Next, it initializes a set $\mathcal{S}^{\mathsf{Corrupt}} \leftarrow \emptyset$ to keep track of corrupted keys and sets $\mathcal{S}^i \leftarrow \emptyset$ to keep track of messages that were signed with respect to public key vk_i. It passes $vk_i, i \in [\ell]$ to $\mathcal{A}$.
2. The adversary may now adaptively issue *sign*-queries (m, i) where m is a message and $i \in [\ell]$ and *corrupt*-queries i (where also $i \in [\ell]$). $\mathcal{C}$ responds to the respective queries as follows. When issued a sign query (m, i), $\mathcal{C}$ updates $\mathcal{S}^i$ to $\mathcal{S}^i \leftarrow \mathcal{S}^i \cup \{m\}$. Next, it returns $\sigma \xleftarrow{\$} \mathsf{Sign}(sk_i, m)$. When issued a corrupt query i, $\mathcal{C}$ updates $\mathcal{S}^{\mathsf{Corrupt}}$ to $\mathcal{S}^{\mathsf{Corrupt}} \leftarrow \mathcal{S}^{\mathsf{Corrupt}} \cup \{i\}$ and returns sk_i. $\mathcal{A}$ is restricted to perform no more than μ overall sign-queries.

3. Finally, $\mathcal{A}$ outputs a forgery (i^*, m^*, σ^*).

Definition 3 (MU-EUF-CMA$^{\mathsf{Corr}}$-security). *We say that an adversary* (t, μ, ℓ, ϵ)*-breaks the* multi user existential unforgeability under chosen message attacks with adaptive corruptions *security of a signature scheme* SIG *if it runs in time* t *in the above security game and*

$$\Pr\left[\mathsf{Vfy}(vk_{i^*}, m^*, \sigma^*) = 1 : i^* \notin \mathcal{S}^{\mathsf{Corrupt}} \wedge m^* \notin \mathcal{S}^{i^*}\right] \geq \epsilon\,.$$

4 Non-interactive Proof Systems

Given a binary relation $R \subseteq X \times W$ and (x, w) such that $R(x, w)$ we call x the statement and w the witness. A non-interactive proof system $\mathsf{NIPS} = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Vfy})$ for witness relation R is a three-tuple of PPT algorithms.

- The common reference string generation algorithm, on input 1^κ, returns a *common reference string*, $\mathsf{CRS} \xleftarrow{\$} \mathsf{Gen}(1^\kappa)$.
- The prove algorithm when input (x, w) such that $R(x, w)$ returns a proof $\pi \xleftarrow{\$} \mathsf{Prove}(\mathsf{CRS}, x, w)$ with respect to CRS.
- The verification algorithm verifies a proof, $\mathsf{Vfy}(\mathsf{CRS}, x, \pi) \in \{0, 1\}$.

Definition 4. *We call* NIPS *a* witness indistinguishable proof of knowledge *(NIWI-PoK) for R, if the following conditions are satisfied:*

Perfect completeness. *For all* $\kappa \in \mathbb{N}$ *it holds that if* $R(x, w)$ *then*

$$\Pr\left[\mathsf{NIPS.Vfy}(\mathsf{CRS}, x, \pi) = 1 : \mathsf{CRS} \xleftarrow{\$} \mathsf{NIPS.Gen}(1^\kappa) \wedge \pi \xleftarrow{\$} \mathsf{Prove}(\mathsf{CRS}, x, w)\right] = 1$$

Perfect Witness Indistinguishability. *Let* $\mathsf{CRS} \xleftarrow{\$} \mathsf{Gen}(1^\kappa)$*. For* $b \in \{0, 1\}$ *we denote by* $\mathcal{O}_b$ *an oracle that when input* (x, w_0, w_1) *such that* $R(x, w_b)$ *returns* $\pi \xleftarrow{\$} \mathsf{Prove}(\mathsf{CRS}, x, w_b)$*. We require*

$$\Pr\left[\mathcal{A}^{\mathcal{O}_0} = 1\right] = \Pr\left[\mathcal{A}^{\mathcal{O}_1} = 1\right]$$

Simulated CRS. *There exists an algorithm* $(\mathsf{CRS}_{\mathsf{sim}}, \tau) \xleftarrow{\$} \mathcal{E}_0$ *that, on input* 1^κ*, outputs a simulated common reference string* $\mathsf{CRS}_{\mathit{sim}}$ *and a trapdoor* τ*.*

Perfect Knowledge Extraction on Simulated CRS. *Let* $(\mathsf{CRS}_{\mathit{sim}}, \tau) \xleftarrow{\$} \mathcal{E}_0(1^\kappa)$*. We require the existence of an algorithm* $\mathcal{E}_1$ *such that for all* $(\pi, x) \leftarrow \mathcal{A}$ *that satisfy* $\mathsf{NIPS.Vfy}(\mathsf{CRS}_{\mathit{sim}}, x, \pi) = 1$ *it holds that*

$$\Pr\Big[w \xleftarrow{\$} \mathcal{E}_1(\mathsf{CRS}_{\mathit{sim}}, \pi, x, \tau) : (x, w) \in R\Big] = 1$$

Secure NIWI-PoK. *Let* $\mathsf{CRS}_{\mathit{real}} \xleftarrow{\$} \mathsf{NIPS.Gen}(1^\kappa)$ *and* $(\mathsf{CRS}_{\mathit{sim}}, \tau) \xleftarrow{\$} \mathcal{E}_0(1^\kappa)$*. We say that an algorithm* $(t, \epsilon_{\mathsf{CRS}})$*-breaks the security of a NIWI-PoK if it runs in time* t *and it holds that*

$$\Pr\Big[\mathcal{A}(\mathsf{CRS}_{\mathit{real}}) = 1)\Big] - \Pr\Big[\mathcal{A}(\mathsf{CRS}_{\mathit{sim}}) = 1\Big] \geq \epsilon_{\mathsf{CRS}}$$

If $\mathsf{CRS} \xleftarrow{\$} \mathsf{Gen}(1^\kappa)$ we call CRS hiding and if $(\mathsf{CRS}_{\mathsf{sim}}, \cdot) \xleftarrow{\$} \mathcal{E}_0(1^\kappa)$ we call $\mathsf{CRS}_{\mathsf{sim}}$ binding. It is easy to verify that perfect witness indistinguishability on a hiding CRS is preserved if many statements are proven.

Defining a Relation. Consider the following equation over (z, r)

$$1 = e(z, k_z) \cdot e(r, k_r) \cdot e(m, k) \tag{1}$$

The core of our signature scheme will be the assumption (which we will justify later) that given (k_z, k_r, k, m) it is hard to compute (z, r) that satisfy equation 1. We define a relation as follows:

$$R\left((k_z, k_r, k, m), (z, r)\right) = \begin{cases} 1, \text{ if } 1 = e(z, k_z) \cdot e(r, k_r) \cdot e(m, k) \\ 0, \text{ else} \end{cases}$$

Suitable Proof Systems. The SXDH-based Groth-Sahai proof system [16] is an (efficient) proof system for witness relation R. Note that equation 1 is *linear* where the variables live in $\mathcal{G}_1$. In this case each commitment costs two elements from $\mathcal{G}_1$ and a proof element costs additional two elements from $\mathcal{G}_2$ (instead of four elements from $\mathcal{G}_1$ and $\mathcal{G}_2$ if we had quadratic equations).

Since we need the notation for our signature scheme we recall SXDH-based Groth-Sahai proofs with efficiency improved verification [24] for relation R here.

$\mathsf{CRS} \xleftarrow{\$} \mathsf{Gen}(1^\kappa)$: The common reference string generation algorithm samples $\mathbb{G} = (e, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T, g_1, g_2, p) \xleftarrow{\$} \mathsf{GEN.asym}(1^\kappa)$, $\vec{v}_1 = (g_1, f_1) \xleftarrow{\$} \mathcal{G}_1^2$ and $\vec{v}_2 = (\hat{g}_1, \hat{f}_1) \notin \mathsf{DDH}(g_1, f_1)$. It returns $(\mathbb{G}, \vec{v}_1, \vec{v}_2)$.

$\pi \xleftarrow{\$} \mathsf{Prove}(\mathsf{CRS}, (k_z, k_r, k, m), (z, r))$: The prove algorithm first commits to z and r via

$$\begin{aligned} C_z &= (1, z) \cdot \vec{v}_1^{\delta_{z,1}} \cdot \vec{v}_2^{\delta_{z,2}} \\ C_r &= (1, r) \cdot \vec{v}_1^{\delta_{r,1}} \cdot \vec{v}_2^{\delta_{r,2}} \end{aligned}$$

where multiplication is done component-wise. Next, it computes proofs that the commitments actually contain a solution to equation 1. These are computed as

$$\pi' = (\pi'_1, \pi'_2) = \left(k_z^{-\delta_{z,1}} \cdot k_r^{-\delta_{r,1}}, k_z^{-\delta_{z,2}} \cdot k_r^{-\delta_{r,2}}\right)$$

The proof is returned as $\pi = (C_z, C_r, \pi') \in \mathcal{G}_1^4 \times \mathcal{G}_2^2$.

$\mathsf{Vfy}(\mathsf{CRS}, (k_z, k_r, k, m), \pi)$: The verification algorithm outputs 1 iff

$$\left(E((1, m), k)\right)^{-1} = E(C_z, k_z) \cdot E(C_r, k_r) \cdot E(\vec{v}_1, \pi'_1) \cdot E(\vec{v}_2, \pi'_2) \tag{2}$$

$(\mathsf{CRS}_{\mathsf{sim}}, td) \xleftarrow{\$} \mathcal{E}_0(1^\kappa)$: The simulated CRS generation algorithm samples $\mathbb{G} = (e, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T, g_1, g_2, p) \xleftarrow{\$} \mathsf{GEN.asym}(1^\kappa)$, $\vec{v}_1 = (g_1, f_1) \xleftarrow{\$} \mathcal{G}_1^2$ and $\vec{v}_2 = (\hat{g}_1, \hat{f}_1) \xleftarrow{\$} \mathsf{DDH}(g_1, f_1)$. It sets $x = \mathsf{log}_{g_1}(f_1)$ and returns $((\mathbb{G}, \vec{v}_1, \vec{v}_2), x)$.

That for any attacker $\mathcal{A}$ that $(t_\mathcal{A}, \epsilon_\mathcal{A})$-breaks the NIWI-PoK security of this proof system there is an attacker $\mathcal{B}$ that $(t_\mathcal{B}, \epsilon_\mathcal{B})$-breaks the SXDH-assumption in $\mathbb{G}$ with $t_\mathcal{A} \approx t_\mathcal{B}$ and $\epsilon_\mathcal{B} \geq \epsilon_\mathcal{A}$ is proven in [16]. We stress that if $\vec{v}_2 \in \mathsf{DDH}(\vec{v}_1)$ then $(\vec{v}_1, \vec{v}_2)$ is a perfectly binding CRS whereas if $\vec{v}_2 \notin \mathsf{DDH}(\vec{v}_1)$ then $(\vec{v}_1, \vec{v}_2)$ yields a perfectly hiding CRS both of which are computationally indistinguishable under the XDH_1 assumption in $\mathbb{G}$.

5 Our new Signature Scheme

Intuition of our scheme. Before we introduce our scheme formally we would like to give some intuition what is behind the scheme. Actually our scheme is similar to the DLIN-based signature scheme from [24] that allows for *linear* OR-proofs. However, we do not need OR-proofs at all.

A signature over m is an SXDH-based Groth-Sahai proof [16] of satisfiability of equation 1 (where (k_z, k_r, k) are given in the public parameters and the public key, respectively). The system parameters contain part of a Groth-Sahai CRS for relation R described in the previous section. The hash of a message, the verification key of the signer and a uniformly random bit completes the CRS. Now, the signer (using sk) computes (z, r) that satisfies equation 1 and generates a proof of this fact using Prove from the proof system of the previous section. We note that there are many possible solutions to equation 1. The secret key of our signature scheme allows to compute exactly one satisfying solution to equation 1. However, two *distinct* solutions yield a solution to the instance (k_z, k_r) of the DP_2 problem.

Description of our scheme. The scheme works as follows.

$\mathsf{SIG.Setup}(1^\kappa)$. The setup algorithm, on input 1^κ, works as follows:

1. Sample $\mathbb{G} = (e, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T, g_1, g_2, p) \xleftarrow{\$} \mathsf{GEN.asym}(1^\kappa)$.
2. Sample $f_1 \xleftarrow{\$} \mathcal{G}_1$ and $k_z, k_r \xleftarrow{\$} \mathcal{G}_2$ and set $\vec{v}_1 = (g_1, f_1)$.
3. Choose a hash-function $H : \{0,1\}^* \to \mathcal{G}_1$. The security analysis will view H as a random oracle.

It returns $\mathit{\Pi} \leftarrow (\mathbb{G}, k_z, k_r, \vec{v}_1, H)$. The message space is $\mathcal{G}_1$.

$\mathsf{SIG.Gen}(\mathit{\Pi})$. The key generation algorithm samples $\chi, \gamma \xleftarrow{\$} \mathbb{Z}_p$ and computes $k = k_z^\chi k_r^\gamma \in \mathcal{G}_2$. The key is returned as $(vk, sk) \leftarrow (k, (\chi, \gamma))$.

$\mathsf{SIG.Sign}(sk, m)$. The sign algorithm first checks if m has been already signed. If this is the case it recovers the bit $b_{vk,m}$ that was previously used to sign m [1]. Else it samples $b_{vk,m} \xleftarrow{\$} \{0,1\}$. Next, it proceeds as follows (recall that $m \in \mathcal{G}_1$).

1. Compute $z = m^{-\chi}$ and $r = m^{-\gamma}$.
2. Compute $\vec{v}_2 = (H(0||vk||m||b_{vk,m}), H(1|vk|||m||b_{vk,m})) \in \mathcal{G}_1^2$ and set $\mathsf{CRS} = (\vec{v}_1, \vec{v}_2)$.
3. Run the prove algorithm for relation R from the previous section and return $\sigma \xleftarrow{\$} \mathsf{Prove}(\mathsf{CRS}, (k_z, k_r, k, m), (z, r)) \in \mathcal{G}_1^4 \times \mathcal{G}_2^2$.

$\mathsf{SIG.Vfy}(vk, m, \sigma)$. The verification algorithm accepts iff $\mathsf{Vfy}(\mathsf{CRS}, (k_z, k_r, k, m), \sigma)$ where $\mathsf{CRS} = (\vec{v}_1, \vec{v}_2)$ and $v_2 = (H(0||vk||m||0), H(1||vk||m||0))$ or $v_2 = (H(0||vk||m||1), H(1||vk||m||1))$.

[1] Note that we could also let the signer evaluate a pseudo-random function on m to determine b. According to [21] another very simple solution is to determine b by evaluating another hash function H' on m and vk (which again will be viewed as a random oracle by the analysis). This way the signer does not need to maintain states.

Remark 1 (On the requirement of a trusted setup). We note that if the scheme is implemented the way we describe it here we require Setup to be run by a trusted party. We can get rid of this requirement if we let the public parameters contain only the description of the group. In this case each user needs to choose H, $\vec{v}_1 \leftarrow^{\$} \mathcal{G}_1^2$ and $k_z, k_r \leftarrow^{\$} \mathcal{G}_2$ itself and publish these as part of its public key. By the random self reducibility of DDH and DP the tightness of the reduction will be preserved. However, this approach leads to longer public keys. Because of this and for ease of readability we chose to describe the scheme as above.

Next, we show that there is a tight reduction from breaking the SXDH-assumption to breaking the unforgeability of the above signature scheme.

Theorem 1. *For any attacker $\mathcal{A}$ that $(t, \mu, \ell, \epsilon_{\mathsf{SIG}})$-breaks the* MU-EUF-CMA$^{\mathsf{Corr}}$*-security of* SIG *there is an attacker $\mathcal{B} = (\mathcal{B}_{\mathsf{XDH}}, \mathcal{B}_{\mathsf{DP}})$ such that $\mathcal{B}_{\mathsf{XDH}}$ $(t_{\mathsf{XDH}}, \epsilon_{\mathsf{XDH}})$-breaks the XDH-assumption in $\mathcal{G}_1$ or $\mathcal{B}_{\mathsf{DP}}$ $(t_{\mathsf{DP}}, \epsilon_{\mathsf{DP}})$-breaks the double pairing assumption in $\mathcal{G}_2$ with $t \approx t_{\mathsf{XDH}} \approx t_{\mathsf{DP}}$ and*

$$\epsilon_{\mathsf{SIG}} < \frac{\ell^2}{2 \cdot p} + 2 \cdot \left(\epsilon_{\mathsf{XDH}} + \epsilon_{\mathsf{DP}} + \frac{\mu + 1}{p} \right) .$$

The analysis will view H as a random oracle.

Proof. The proof is built on the following fact: Given only the public key, there are many possible secret keys and the actual values of χ and γ are information theoretically hidden. However, given a message and a secret key the pair (z, r) is determined. That is, a given secret key allows to compute exactly one pair that satisfies equation 1. At the same time, even if the secret key is available, any other tuple that satisfies equation 1 allows to solve an instance of the DP_2 problem. We argue that since the signer commits to (z, r) via *hiding* commitments the actual values (z, r) are information theoretically hidden from the view of $\mathcal{A}$. Therefore the secret key is also hidden from the adversary. Now, the reduction will manipulate H to produce binding commitment keys for (almost) any adversarially generated signature. From this, we can extract a DP solution with probability $1 - \frac{1}{p}$.

The proof proceeds in a sequence of games. Here, we denote by $\Pr[\chi_i]$ the probability that $\mathcal{A}$ is considered successful in game i. Let us denote by (i^*, m^*, σ^*) the forgery otuput by $\mathcal{A}$ and vk_{i^*} by vk^*.

Game 0. This game is the real MU-EUF-CMA$^{\mathsf{Corr}}$-security game. When issued a hash-query for the string s the reduction $\mathcal{R}$ first checks if s has already been hashed. If this is the case it returns the previously computed value $H(s)$. Otherwise it samples r uniformly at random from $\mathcal{G}_1$ and sets and returns $H(s) = r$. All other queries are answered according to the MU-EUF-CMA$^{\mathsf{Corr}}$-security experiment. This perfectly simulates the challenger in the random-oracle model. Thus, we have:

$$\Pr[\chi_0] = \epsilon_{\mathsf{SIG}}$$

Game 1. Let Q_{vkcoll} denote the following event:

$$Q_{\mathsf{vkcoll}} := \left\{ \exists (i,j) \in [\ell]^2 : i \neq j \wedge vk_i = vk_j \right\}$$

In Game 1 $\mathcal{R}$ aborts (and $\mathcal{A}$ looses) if event Q_{vkcoll} occurs. Since χ and γ are chosen uniformly at random by $\mathcal{R}$, public keys are distributed uniformly random over $\mathcal{G}_2$ which implies $\Pr[Q] = \frac{\ell \cdot (\ell - 1)}{2 \cdot p}$. Thus, we have

$$|\Pr[\chi_0] - \Pr[\chi_1]| \leq \frac{\ell^2}{2 \cdot p}$$

Game 2. Before we introduce the changes made in Game 2 let us fix some notation. Let $b_{vk,m}$ denote the bit that is (lazily) sampled by $\mathcal{R}$ during signing on m under vk. In Game 2, $\mathcal{R}$ aborts if for the forgery (i^*, m^*, σ^*) that is output by $\mathcal{A}$ it holds that $v_2^* = (H(0||vk^*||m^*||b_{vk^*,m^*}), H(1||vk^*||m^*||b_{vk^*,m^*}))$. In other words, $\mathcal{R}$ aborts (and $\mathcal{A}$ looses) if $\mathcal{A}$ chooses for the forgery the same bit b_{vk^*,m^*} that $\mathcal{R}$ would have chosen itself to sign m^* under vk^*. Since $\mathcal{R}$ chooses each bit uniformly at random the actual choice of b_{vk^*,m^*} is information theoretically hidden from the view of $\mathcal{A}$ (recall that all vk are distinct due to Game 1). Thus we have

$$\Pr[\chi_1] \leq 2 \cdot \Pr[\chi_2]$$

Game 3. In Game 3 the reduction proceeds similarly to Game 2 except for the following: $\mathcal{R}$ lazily programs the hash-oracle such that for every pair of m and vk we have that $(H(0||vk||m||1 - b_{vk,m}), H(1||vk||m||1 - b_{vk,m})) \in \mathsf{DDH}(g_1, f_1)$. By the random self reducibility of DDH we get:

$$|\Pr[\chi_2] - \Pr[\chi_3]| < \epsilon_{\mathsf{XDH}}$$

Game 4. This game is similar to Game 3, except that $\mathcal{R}$ aborts (and $\mathcal{A}$ looses), if for any sign query (m, i) issued by $\mathcal{A}$ during the security experiment we have that $(H(0||vk_i||m||b_{vk_i,m}), H(1||vk_i||m||b_{vk_i,m})) \in \mathsf{DDH}(g_1, f_1)$. Since images of H are distributed uniformly over $\mathcal{G}$ we have that

$$|\Pr[\chi_3] - \Pr[\chi_4]| \leq \frac{\mu}{p}$$

Game 5. Game 5 proceeds exactly like Game 4 except for the following. $\mathcal{R}$ aborts if it cannot extract a satisfying assignment for equation 1 from σ^*. Due to Game 3 we know that $(\hat{g}_1, \hat{f}_1) = (H(0||vk^*||m^*||1 - b_{vk^*,m^*}), H(1||vk^*||m^*||1 - b_{vk^*,m^*})) \in \mathsf{DDH}(g_1, f_1)$. Therefore $(\vec{v}_1, \vec{v}_2)$ is in the (first) range of $\mathcal{E}_0(1^\kappa)$ and gives a perfectly binding CRS.

Given the trapdoor $\tau = \log_{g_1}(f_1)$ and using $\mathcal{E}_1$, $\mathcal{R}$ is able to extract (z^*, r^*) that satisfy equation 1 due to the perfect knowledge extraction on simulated CRS [16]. Thus, we have:

$$\Pr[\chi_4] = \Pr[\chi_5]$$

Game 6. Game 6 proceeds exactly as Game 5 except for the following. The reduction aborts (and $\mathcal{A}$ looses) if for the forgery that $\mathcal{A}$ outputs it holds that the satisfying assignment of equation 1, (z^*, r^*), that is extracted by $\mathcal{R}$ from σ^* is equal to $((m^*)^{-\chi}, (m^*)^{-\gamma})$. Since for all sign queries (m, i) issued by $\mathcal{A}$ it holds that $(H(0||vk_i||m||b_{vk_i,m}), H(1||vk_i||m||b_{vk_i,m})) \notin \mathsf{DDH}(g_1, f_1)$ (which is due to Game 4) the signatures output by $\mathcal{R}$ are perfectly hiding proofs and do not leak any valuable information on (z, r) that are used by $\mathcal{R}$ to compute the respective commitements. From the view of the adversary all (z, r) that satisfy the respective equation 1 are equally likely. In particular the only information that the adversary obtains on χ and γ comes from the public key. However the public key provides the adversary with one linear equation in two unknowns which has p possible solutions. Thus we have:

$$|\Pr[\chi_5] - \Pr[\chi_6]| \leq \frac{1}{p}$$

Lemma 2. $\Pr[\chi_6] < \epsilon_{\mathsf{DP}_2}$.

We will show that any forgery output by the adversary in Game 6 allows $\mathcal{B}_{\mathsf{DP}}$ to solve a given instance of the DP_2 assumption. To this end, assume that $\mathcal{A}$ outputs a valid signature σ^* for m^* that was not signed before under vk. By Game 5 we know that from σ^* we can extract (z^*, r^*) such that $1 = e(k_z, z^*) \cdot e(k_r, r^*) \cdot (k^*, m^*)$. Moreover due to Game 6 we know that $(z^*, r^*) \neq (z, r) = ((m^*)^{-\chi}, (m^*)^{-\gamma})$. However, we do know that (z, r) also satisfies equation 1. Now, $(\frac{z}{z^*}, \frac{r}{r^*}) \neq (1, 1)$ yields a solution to the DP_2 instance $(k_z, k_r) \in \mathcal{G}_2$:

$$\begin{aligned} e(\frac{z}{z^*}, k_z) \cdot e(\frac{r}{r^*}, k_r) =& e(z, k_z) \cdot e(r, k_r) \cdot e(z^*, k_z)^{-1} \cdot e(r^*, k_r)^{-1} \\ =& e(z, k_z) \cdot e(r, k_r) \cdot e(m^*, k^*)^{1-1} \cdot e(z^*, k_z)^{-1} \cdot e(r^*, k_r)^{-1} \\ =& 1 \end{aligned}$$

where the last equation is due to the fact that both, (z, r) and (z^*, r^*), satisfy equation 1. This completes our proof. □

We stress that the reduction is able to reveal the secret key corresponding to a public key in every single game throughout the proof and is nevertheless able to extract a solution to a hard problem from a forgery. We do not even have to re-randomize publicly available values. That is, we can use k_z and k_r, as well as v_1 from $\mathit{\Pi}$ for all users.

Acknowledgements. We would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

1. NIST FIPS 186-4. Digital signature standard (dss). Technical report, NIST, 2013.
2. Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 209–236, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Berlin, Germany.
3. Christoph Bader, Dennis Hofheinz, Tibor Jager, Eike Kiltz, and Yong Li. Tightly secure authenticated key exchange. unpublished manuscript, 2014.
4. Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid. Nist sp 800-57, recommendation for key management – part 1: General (revision 3). Technical report, 2012.
5. Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 259–274, Bruges, Belgium, May 14–18, 2000. Springer, Berlin, Germany.
6. Mihir Bellare, Ran Canetti, and Hugo Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols (extended abstract). In *30th Annual ACM Symposium on Theory of Computing*, pages 419–428, Dallas, Texas, USA, May 23–26, 1998. ACM Press.
7. Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 232–249, Santa Barbara, CA, USA, August 22–26, 1993. Springer, Berlin, Germany.
8. Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416, Saragossa, Spain, May 12–16, 1996. Springer, Berlin, Germany.
9. Daniel J. Bernstein. Proving tight security for Rabin-Williams signatures. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 70–87, Istanbul, Turkey, April 13–17, 2008. Springer, Berlin, Germany.
10. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532, Gold Coast, Australia, December 9–13, 2001. Springer, Berlin, Germany.
11. Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 453–474, Innsbruck, Austria, May 6–10, 2001. Springer, Berlin, Germany.
12. Jean-Sébastien Coron. Optimal security proofs for PSS and other signature schemes. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 272–287, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer, Berlin, Germany.
13. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *Advances in Cryptology – CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187, Santa Barbara, CA, USA, August 21–25, 1994. Springer, Berlin, Germany.

14. T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008. Updated by RFCs 5746, 5878.
15. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.
16. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432, Istanbul, Turkey, April 13–17, 2008. Springer, Berlin, Germany.
17. Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 590–607, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Berlin, Germany.
18. Tibor Jager, Florian Kohlar, Sven Schäge, and Jörg Schwenk. Generic compilers for authenticated key exchange. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 232–249, Singapore, December 5–9, 2010. Springer, Berlin, Germany.
19. Tibor Jager, Florian Kohlar, Sven Schäge, and Jörg Schwenk. On the security of TLS-DHE in the standard model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 273–293, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Berlin, Germany.
20. Saqib A. Kakvi and Eike Kiltz. Optimal security proofs for full domain hash, revisited. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 537–553, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany.
21. Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger, editors, *ACM CCS 03: 10th Conference on Computer and Communications Security*, pages 155–164, Washington D.C., USA, October 27–30, 2003. ACM Press.
22. Jonathan Katz and Moti Yung. Scalable protocols for authenticated group key exchange. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 110–125, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Berlin, Germany.
23. Yong Li, Sven Schäge, Zheng Yang, Christoph Bader, and Jörg Schwenk. New modular compilers for authenticated key exchange. In Ioana Boureanu, Philippe Owesarski, and Serge Vaudenay, editors, *ACNS 14: 12th International Conference on Applied Cryptography and Network Security*, volume 8479 of *Lecture Notes in Computer Science*, pages 1–18, Lausanne, Switzerland, June 10–13, 2014. Springer, Berlin, Germany.
24. Benoit Libert, Thomas Peters, Marc Joye, and Moti Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive nizk proofs and cca2-secure encryption from homomorphic signatures. In Phong Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - Eurocrypt 2014*. Springer, 2014.
25. Alfred Menezes and Nigel P. Smart. Security of signature schemes in a multi-user setting. *Des. Codes Cryptography*, 33(3):261–274, 2004.
26. Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th Annual Symposium on Foundations of Computer Science*, pages 458–467, Miami Beach, Florida, October 19–22, 1997. IEEE Computer Society Press.

27. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on Theory of Computing*, pages 427–437, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press.
28. Sven Schäge. Tight proofs for signature schemes without random oracles. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 189–206, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany.
29. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252, Santa Barbara, CA, USA, August 20–24, 1989. Springer, Berlin, Germany.
30. Yannick Seurin. On the exact security of schnorr-type signatures in the random oracle model. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 554–571, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany.