Tightly Secure Signatures and Public-Key Encryption

Dennis Hofheinz and Tibor Jager

Karlsruhe Institute of Technology, Germany
{dennis.hofheinz,tibor.jager}@kit.edu

Abstract. We construct the first public-key encryption scheme whose chosen-ciphertext (i.e., IND-CCA) security can be proved under a standard assumption *and* does not degrade in either the number of users or the number of ciphertexts. In particular, our scheme can be safely deployed in unknown settings in which no a-priori bound on the number of encryptions and/or users is known.

As a central technical building block, we construct the first structure-preserving signature scheme with a tight security reduction. (This signature scheme may be of independent interest.) Combining this scheme with Groth-Sahai proofs yields a tightly simulation-sound non-interactive zero-knowledge proof system for group equations. If we use this proof system in the Naor-Yung double encryption scheme, we obtain a tightly IND-CCA secure public-key encryption scheme from the Decision Linear assumption.

We point out that our techniques are not specific to public-key encryption security. Rather, we view our signature scheme and proof system as general building blocks that can help to achieve a tight security reduction.

Keywords: Tight security proofs, structure-preserving signatures, public-key encryption, Groth-Sahai proofs.

1 Introduction

Many interesting cryptographic primitives (such as public-key encryption and signature schemes) cannot be proven secure with current techniques, as their security would imply $P \neq NP$. Instead, we usually provide a proof of security under a suitable (computational) assumption (such as the hardness of factoring large integers). Concretely, a *security reduction* shows that any successful adversary $\mathcal{A}$ on the scheme's security can be converted into a successful solver $\mathcal{B}$ of the underlying computational problem. Naturally, we would desire that $\mathcal{B}$'s success $\epsilon_{\mathcal{B}}$ is at least as large as $\mathcal{A}$'s success $\epsilon_{\mathcal{A}}$ in attacking the system. However, security reductions often suffer from a nontrivial multiplicative *security loss* L (such that only $\epsilon_{\mathcal{A}} \leq L \cdot \epsilon_{\mathcal{B}}$ can be guaranteed).

The issue of a nontrivial security loss becomes particularly problematic, e.g., in the case of a realistic public-key encryption (PKE) scenario with many users who encrypt and send many ciphertexts. Standard security notions for PKE

R. Safavi-Naini and R. Canetti (Eds.): CRYPTO 2012, LNCS 7417, pp. 590–607, 2012.

schemes (such as IND-CCA security [44, 20]) only consider one user and one ciphertext. In particular, with very few exceptions ([9, 32]), most security proofs of encryption schemes only prove security in this simplified scenario. This can be justified with general results ([8, 9]) that show that one-user, one-ciphertext PKE security implies security in the much more realistic multi-user, multi-ciphertext case. However, this generic reduction suffers from a reduction loss of $L = n_U \cdot n_C$, where n_U is the number of users, and n_C is the number of ciphertexts per user.

That is, even if a PKE scheme reaches a certain level of security in the commonly considered one-user, one-ciphertext setting, its security level may be significantly lower in a realistic setting. (In fact, Bellare et al. [7] give a concrete example of such a scheme in the symmetric-key setting.) This is particularly problematic, since it may not be clear at deployment time for how many users and encryptions a PKE scheme will be used. We thus note that the analysis of cryptographic primitives in the multi-user setting is necessary to derive concrete security guarantees for realistic settings.

Let us say that a security reduction (in the multi-user setting) is *tight* if the corresponding reduction loss L is a (preferably small) constant. In particular, the security of a tightly secure scheme does not deteriorate in the number of users (or encryptions). For some security notions and constructions, tightly secure reductions can be constructed relatively painlessly. For instance, the random self-reducibility of the Decisional Diffie-Hellman problem allows to tightly prove the IND-CPA security of ElGamal encryption [21] even with many users and ciphertexts [9]. However, for other security notions, it seems inherently difficult to derive a tight security reduction.

For instance, there is no known PKE scheme with a tight (IND-CCA) security reduction from a standard assumption.[1] Diving into the technical details for a moment, one reason for this apparent difficulty is that an IND-CCA security reduction must be able to decrypt all of $\mathcal{A}$'s decryption queries, but must not be able to decrypt its own IND-CCA challenge. One way to resolve this dilemma is to partition the set of ciphertexts into those that can be decrypted, and those that cannot. (For instance, one can set up the proof simulation for $\mathcal{A}$ such that the reduction can decrypt all ciphertexts except for one single challenge ciphertext; examples of this approach are [11, 34, 35].) This proof technique can only argue about a small number of ciphertexts at a time, and a hybrid argument is required to show security in the multi-ciphertexts case. Such a hybrid argument results again in a reduction loss that is linear in the number of ciphertexts.

Another way to show IND-CCA security is to argue with the information the adversary has about the secret key. (Examples of this approach are [18, 19, 38].)

[1] Bellare, Boldyreva, and Micali [9] show that the security loss of Cramer-Shoup encryption [18] does not depend on the number of users; however, their reduction loss still grows linearly in the number of ciphertexts per user. The identity-based encryption schemes [25, 26] enjoy a tight reduction, and can be generically converted into tightly IND-CCA secure PKE schemes [14]; however, they rely on a non-standard multi-challenge assumption. A similar argument holds for the tightly IND-SO-CCA secure PKE scheme of Hofheinz [32].

Since the size of the secret key is limited, its entropy can only be used to argue about the security of a limited number of ciphertexts at a time. Again a hybrid argument (entailing a linear reduction loss) is required to argue about the security of many ciphertexts. One could hope that the described inherent hybrid arguments of partitioning and entropy-based strategies to show IND-CCA security can be circumvented using dual system (identity-based) encryption techniques [47, 39]. In a nutshell, dual system encryption provides a way to subtly and gradually randomize the distribution of challenge ciphertexts (and user secret keys in an IBE scheme) without explicitly partitioning the set of ciphertexts into decryptable and non-decryptable ones. However, while dual system techniques rely on re-randomizable computational problems (such as the Decision Linear problem), and thus in principle should not suffer from the described problems, all known dual systems schemes still have to use a hybrid argument and do not achieve a tight security reduction. Note that one can construct IND-CCA-secure public-key encryption schemes with tight reduction in the random oracle model [5], for instance by applying the Fujisaki-Okamoto transform [23] to the tightly IND-CPA-secure schemes from [9].

In this paper, we present a general technique to construct tightly secure cryptographic primitives. As an example, we construct the first PKE scheme that is tightly IND-CCA secure under a simple assumption. Concretely, all the constructions in this paper build on the Decision Linear (DLIN) assumption.[2]

Our main technical building block is a *structure-preserving* signature scheme with a tight security reduction.[3] Loosely speaking, a structure-preserving signature scheme is one in which verification can be expressed as a sequence of group equations. In particular, structure-preserving schemes are amenable to Groth-Sahai (GS) proofs [31], which are efficient non-interactive proof systems for sets of equations over a group. Following a known paradigm [40, 30, 16], we then turn our signature scheme into a *simulation-sound* non-interactive zero-knowledge proof system for group equations.[4] Since our signature is tightly secure, so is the proof system.

This tightly secure and simulation-sound proof system offers the technical means to achieve tight security. We exemplify this by implementing the Naor-Yung paradigm [42, 40, 13] with our proof system to obtain a tightly IND-CCA secure PKE scheme. This construction appears in Section 5.

[2] However, we expect that our constructions also naturally generalize to the — potentially weaker — K-Linear assumption and to suitable subgroup decision assumptions.

[3] We construct tightly secure structure-preserving signatures. (In fact, our schemes can sign their own public key; such signature schemes are commonly also referred to as automorphic.) While there exist tightly secure signature schemes (e.g., [24, 12, 17, 10, 36, 46]), and structure-preserving signature schemes (e.g., [22, 3, 16]), our scheme seems to be the first to achieve both properties. This combination of properties is crucial for our applications.

[4] By a simulation-sound zero-knowledge proof system, we mean one in which it is infeasible to generate valid proofs for false statements, even when already having observed *many* simulated proofs for possibly false statements.

Our signature scheme is tree-based and inspired by the scheme of Boneh, Mironov, and Shoup [12]. However, their scheme is not structure-preserving, as it uses the hash of group elements as an exponent. To avoid this kind of "domain translation," we construct a one-time signature scheme in which signatures and messages are vectors of group elements. (Since we want to implement a tree-based many-time signature scheme, we require, however, that messages can be longer than public verification keys.) To describe our (one-time) scheme, assume groups $\mathbb{G}, \mathbb{G}_T$ with pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Write $E : \mathbb{G}^3 \times \mathbb{G} \to \mathbb{G}_T^3$ for component-wise pairing. (That is, $E((u_1, u_2, u_3), v) = (e(u_1, v), e(u_2, v), e(u_3, v)$.) In a nutshell, a signature for a message $m = (m_i)_{i=1}^n \in \mathbb{G}^n$ is of the form $(s, t) \in \mathbb{G}^2$ and satisfies

$$\left(\prod_{i=1}^{n} E(U_i, m_i)\right) \cdot E(G, s) \cdot E(H, t) = E(X, z), \tag{1}$$

where the $G, H, X, U_1, \ldots, U_n \in \mathbb{G}^3$, and $z \in \mathbb{G}$ are part of the public verification key.[5] This means that m_i, s, t, z act as coefficients in a linear equation (in the $\mathbb{G}$-exponent) for the vectors U_i, G, H, X. Now if all the vectors U_i, G, H, X are DLIN-tuples of the form (g^x, h^y, k^{x+y}) (for fixed g, h, k), then any message can be signed when knowing all U_i, G, H, X-exponents. Now consider a setup in which one U_i (say, U_j) and X are non-DLIN-tuples, and the other U_i and G, H are DLIN-tuples. Then, only messages with a specific m_j-component can be signed. (This is easiest seen by thinking of DLIN-elements as linearly dependent vectors in the exponent; with this view, X and U_j are the only vectors outside the vector space generated by DLIN-tuples. We note that this idea of "unique signatures" already appears in the ROM-based signature scheme of Katz and Wang [37].) In the proof of (non-adaptive) one-time security, this m_j will be set up as the message signed for the adversary $\mathcal{A}$. Furthermore, if the adversary forges a message, we know that this message must have m_j in its j-th component. Since a forged signature must refer to a message M^* that is different from the message M signed for $\mathcal{A}$, there must be an j with $m_j^* \neq m_j$. A small hybrid argument over $j \in [n]$ thus shows security. (We stress that we employ a hybrid argument only over a small set $[n]$ that will not depend on the number of users or ciphertexts. Specifically, our scheme can be implemented with $n = 8$.) From this one-time secure scheme, we will construct a tree-based many-time secure scheme following ideas from [12]; in particular, we will re-use the U_i for many instances of the one-time scheme. (Such a re-use of public key parts has also been used and made explicit [6].) This will finally yield an adaptively secure structure-preserving signature scheme with a tight security reduction. The remaining steps that lead to a tightly simulation-sound proof system and tightly IND-CCA secure public-key encryption follow existing ideas [30, 13], so we will not outline them here. (Details follow inside.)

We note that plugging our tightly simulation-sound proof system into the construction of [13] yields a PKE scheme that is tightly chosen-ciphertext secure

[5] We highlight that (1) actually consists of three pairing product equations. This can in part be justified by [3, Theorem 2], which states that already any secure structure-preserving two-time signature scheme must have at least two verification equations.

even under *key-dependent* message attacks. Similarly, we expect that proofs of chosen-ciphertext security for *identity-based* encryption schemes can be made independent of the number of challenge ciphertexts. (However, here we do not expect to obtain independence from the number of users, i.e., identities.) Besides, structure-preserving signatures have found applications in several areas (e.g., [15, 22, 29]). We expect that *tightly secure* structure-preserving signature schemes lead to tighter security proofs in these applications.

In [2, Appendix C] (the full version of [1]), Abe et al. describe a DLIN-based structure-preserving one-time signature scheme that is more efficient than ours and has subsequently also been proven compatible with our tree-based approach [4]. In particular, together with our work, their scheme yields a more efficient tightly IND-CCA-secure encryption scheme. (We were not aware of their one-time signature scheme when designing ours.)

2 Preliminaries

Notation. If A is a set, then $a \xleftarrow{\$} A$ denotes that a is distributed uniformly over A. If A is a probabilistic algorithm, then $a \xleftarrow{\$} A$ denotes that a is computed by A using fresh random coins. For $n \in \mathbb{N}$ we write $[n]$ to denote the set $[n] = \{1, \ldots, n\}$. For $j \in [n]$ we write $[n \setminus j]$ to denote the set $\{1, \ldots, n\} \setminus \{j\}$.

Digital Signatures. Generally, we assume a parameter generation algorithm $\mathsf{Sig.Param}$ which takes as input the security parameter κ and generates public parameters $\Pi \xleftarrow{\$} \mathsf{Sig.Param}(\kappa)$.

A digital signature scheme $\mathsf{Sig} = (\mathsf{Sig.Gen}, \mathsf{Sig.Sign}, \mathsf{Sig.Vfy})$ consists of three algorithms. Key generation algorithm $\mathsf{Sig.Gen}$ generates, on input parameters Π, a keypair $(vk, sk) \xleftarrow{\$} \mathsf{Sig.Gen}(\Pi)$ consisting of a secret signing key sk and a public verification key vk. The signing algorithm $\mathsf{Sig.Sign}$ inputs a message and the secret signing key, and returns a signature $\sigma \xleftarrow{\$} \mathsf{Sig.Sign}(sk, m)$ of the message. The verification algorithm $\mathsf{Sig.Vfy}$ takes a verification key and a message with corresponding signature as input, and returns $b \leftarrow \mathsf{Sig.Vfy}(vk, m, \sigma)$ where $b \in \{0, 1\}$. We require the usual correctness properties.

Let us recall the *existential unforgeability against chosen message attacks* (EUF-CMA) security experiment [28], played between a challenger and a forger $\mathcal{A}$.

1. The forger, on input public parameters Π, may ask a *non-adaptive* chosen-message query. To this end, it submits a list of messages $M^{(1)}, \ldots, M^{(q_0)}$ to the challenger.
2. The challenger runs $\mathsf{Sig.Gen}(\Pi)$ to generate a keypair (vk, sk). The forger receives vk and a signature $\sigma^{(i)}$ for each chosen message $M^{(i)}$, $i \in [q_0]$.
3. Now the forger may ask *adaptive* chosen-message queries, by submitting messages $M^{(q_0+1)}, \ldots, M^{(q)}$ to the challenger. The challenger returns a signature $\sigma^{(i)}$ under sk for each message $M^{(i)}$, $i \in [q_0 + 1, q]$.
4. Finally the forger outputs a message M^* and signature σ^*.

Definition 1. *An adversary is* adaptive*, if it asks at least one adaptive chosen-message query. Otherwise it is* non-adaptive*. Let $\mathcal{A}$ be an adversary (adaptive or non-adaptive) that runs in time t, makes q chosen-message queries (in total), and outputs (M^*, σ^*). We say that $\mathcal{A}$ (ϵ, t, q)-breaks the EUF-CMA security of* Sig *if*

$$\Pr[\mathsf{Sig.Vfy}(vk, M^*, \sigma^*) = 1 \wedge M^* \notin \{M^{(1)}, \ldots, M^{(q)}\}] \geq \epsilon.$$

We say that $\mathcal{A}$ (ϵ, t, q)-breaks the strong *EUF-CMA security of* Sig *if*

$$\Pr[\mathsf{Sig.Vfy}(vk, M^*, \sigma^*) = 1 \wedge (M^*, \sigma^*) \notin \{(M^{(1)}, \sigma^{(1)}), \ldots, (M^{(q)}, \sigma^{(q)})\}] \geq \epsilon.$$

Accordingly, a signature scheme Sig *is* (ϵ, t, q)-secure *against adaptive (non-adaptive) EUF-CMA attacks, if there exists no adaptive (non-adaptive) adversary that (ϵ, t, q)-breaks* Sig*.*

Complexity Assumptions. In the following, let $\mathbb{G}$ be a group of prime order p. For a generator $g \in \mathbb{G}$ and arbitrary $h \in \mathbb{G}$, let $\log_g(h) \in \mathbb{Z}_p$ be the discrete logarithm of h (to base g), such that $g^{\log_g(h)} = h$.

Definition 2. *Let $g, h \in \mathbb{G}$ be random generators of $\mathbb{G}$. We say that an adversary $\mathcal{A}$ (ϵ, t)-breaks the* Discrete Logarithm *(DLOG) assumption in $\mathbb{G}$, if $\mathcal{A}$ runs in time t and* $\Pr[\mathcal{A}(g, h) = \log_g(h)] \geq \epsilon$.

Furthermore, for generators $g, h, k \in \mathbb{G}$ let $\mathsf{DLIN}(g, h, k)$ denote the set

$$\mathsf{DLIN}(g, h, k) = \{(g^u, h^v, k^{u+v}) : u, v \in \mathbb{Z}_p\}$$

Let $G = (g, 1, k) \in \mathbb{G}^3$ and $H = (1, h, k) \in \mathbb{G}^3$. For two vectors $V = (v_1, v_2, v_3)$ and $W = (w_1, w_2, w_3)$ in $\mathbb{G}^3$ and $u \in \mathbb{Z}_p$ let $V \cdot W := (v_1 \cdot w_1, v_2 \cdot w_2, v_3 \cdot w_3)$ and let $V^u := (v_1^u, v_2^u, v_3^u)$. Then we can write the set $\mathsf{DLIN}(g, h, k)$ equivalently as

$$\mathsf{DLIN}(g, h, k) = \{U : U = G^u \cdot H^v, u, v \in \mathbb{Z}_p\}.$$

Definition 3. *Let $g, h, k \xleftarrow{\$} \mathbb{G}$ be random generators of group $\mathbb{G}$, and let $U \xleftarrow{\$} \mathsf{DLIN}(g, h, k)$ and $V \xleftarrow{\$} \mathbb{G}^3$. We say that an adversary $\mathcal{B}$ (ϵ, t)-breaks the* Decision Linear *(DLIN) assumption in $\mathbb{G}$, if $\mathcal{B}$ runs in time t and*

$$\Pr[\mathcal{B}(g, h, k, U) = 1] - \Pr[\mathcal{B}(g, h, k, V) = 1] \geq \epsilon.$$

3 Structure-Preserving Signatures

In the sequel let $\mathbb{G}, \mathbb{G}_T$ be groups of prime order p with bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, and let g, h, k be random generators of $\mathbb{G}$. For a vector $V = (v_0, v_1, v_2) \in \mathbb{G}^3$ and a group element $w \in \mathbb{G}$, we write $E(V, w)$ to denote the vector

$$E(V, w) = (e(v_0, w), e(v_1, w), e(v_2, w)).$$

For two vectors $V = (v_0, v_1, v_2)$ and $W = (w_0, w_1, w_2)$ we denote with $V \cdot W$ the component-wise product $V \cdot W = (v_0 \cdot w_0, v_1 \cdot w_1, v_2 \cdot w_2)$. For $w \in \mathbb{Z}_p$ we write V^w to denote $V^w = (v_0^w, v_1^w, v_2^w)$.

3.1 One-Time Signatures for Single Group Elements

Let $\mathsf{OTSig} = (\mathsf{OTSig.Gen}, \mathsf{OTSig.Sign}, \mathsf{OTSig.Vfy})$ be the following signature scheme.

$\mathsf{OTSig.Gen}(g, h, k)$**:** Given random generators $g, h, k \in \mathbb{G}$, choose a random generator $z \xleftarrow{\$} \mathbb{G}$ and integers $u, v, x, y \xleftarrow{\$} \mathbb{Z}_p$. Set $U := (g^u, h^v, k^{u+v})$ and $X := (g^x, h^y, k^{x+y})$. Set $vk := (g, h, k, U, X, z)$ and $sk := (u, v, x, y)$ and return (vk, sk).

$\mathsf{OTSig.Sign}(sk, m)$**:** Given a message $m \in \mathbb{G}$ and a secret key $sk = (u, v, x, y)$, compute $s := z^x m^{-u}$ and $t := z^y m^{-v}$ and return $\sigma = (s, t)$.

$\mathsf{OTSig.Vfy}(vk, m, \sigma)$**:** Given a public key $vk = (g, h, k, U, X, z)$, message m, and signature $\sigma = (s, t)$, let $G := (g, 1, k)$ and $H = (1, h, k)$. Return 1 if equation

$$E(U, m) \cdot E(G, s) \cdot E(H, t) = E(X, z)$$

holds. Otherwise return 0.

Theorem 1. *Suppose there exists a non-adaptive adversary $\mathcal{A}$ that $(\epsilon, t, 1)$-breaks the EUF-CMA security of OTSig. Then there exists an adversary $\mathcal{B}$ that (ϵ', t')-breaks the DLIN assumption in $\mathbb{G}$, where t' is roughly the runtime of the EUF-CMA experiment with $\mathcal{A}$, and $\epsilon' \geq \epsilon - 1/p$.*

Due to space limitations, we have to refer to the full version [33] for the proof.

3.2 One-Time Signatures for Vectors of Group Elements

In this section we extend the message space of OTSig from the previous section to vectors $M = (m_1, \ldots, m_n)$ of n elements of $\mathbb{G}$. The scheme is very similar, except that now the public key and the verification equation contain n elements $U_1, \ldots, U_n$ instead of a single element U.

Scheme $\mathsf{OTSig}^n = (\mathsf{OTSig.Gen}^n, \mathsf{OTSig.Sign}^n, \mathsf{OTSig.Vfy}^n)$ works as follows.

$\mathsf{OTSig.Gen}^n(g, h, k)$**:** Given a generators $g, h, k \in \mathbb{G}$, choose a random generator $z \xleftarrow{\$} \mathbb{G}$ and $2(n+1)$ integers $u_1, v_1, \ldots, u_n, v_n, x, y \xleftarrow{\$} \mathbb{Z}_p$. Set $U_i := (g^{u_i}, h^{v_i}, k^{u_i+v_i})$ for $i \in [n]$ and $X := (g^x, h^y, k^{x+y})$. Define the keys as $vk := (g, h, k, U_1, \ldots, U_n, X, z)$ and $sk := (u_1, v_1, \ldots, u_n, v_n, x, y)$ and return (vk, sk).

$\mathsf{OTSig.Sign}^n(sk, M)$**:** Given a message vector $M = (m_1, \ldots, m_n) \in \mathbb{G}^n$ and secret key sk, compute $s := z^x \prod_{i=1}^n m_i^{-u_i}$ and $t := z^y \prod_{i=1}^n m_i^{-v_i}$ and return $\sigma = (s, t)$.

$\mathsf{OTSig.Vfy}^n(vk, M, \sigma)$**:** Given a public key vk, message vector $M = (m_1, \ldots, m_n)$, and signature $\sigma = (s, t)$, let $G := (g, 1, k)$ and $H = (1, h, k)$. Return 1 if equation

$$\prod_{i=1}^{n} E(U_i, m_i) \cdot E(G, s) \cdot E(H, t) = E(X, z)$$

holds. Otherwise return 0.

Theorem 2. *Suppose there exists a non-adaptive adversary $\mathcal{A}$ that $(\epsilon, t, 1)$-breaks the EUF-CMA security of OTSig^n. Then there exists an adversary $\mathcal{B}$ that (ϵ', t')-breaks the DLIN assumption in $\mathbb{G}$, where t' is roughly with runtime of the EUF-CMA experiment with $\mathcal{A}$, and $\epsilon' \geq \epsilon/n - 1/p$.*

Again we have to refer to the full version [33] for the proof.

3.3 Signatures Secure against Non-adaptive Adversaries

Now we construct a signature scheme based on a binary tree of depth d. The scheme has message space $\mathbb{G}^8$, allows us to issue up to 2^d signatures (where d may be large enough such that 2^d is virtually unbounded, e.g. $d = 80$), and is provably secure against non-adaptive adversaries under the DLIN assumption.

Basic Idea. The construction is based on binary Merkle trees [41], instantiated such that all nodes except for the root can be generated "on the fly." In particular, not the complete tree must be stored (which would clearly be infeasible for large d). Each node of the tree consists of a key pair (vk, sk) of our one-time signature scheme from Section 3.2. The two children of this node are authenticated by a signature over their respective public keys that verifies under vk. The key-pairs corresponding to tree leaves are used to sign actual messages.

Recall that a public key consists of a vector $(g, h, k, U_1, \ldots, U_n, X, z)$, where n is the number of group elements to be signed. In order to obtain a tight security reduction, we re-use the public-key components $(g, h, k, U_1, \ldots, U_n)$ for all nodes of the tree. Only the (X, z)-components are unique for each node. The tight reduction is inspired by the proof of the tree-based signature scheme of Boneh et al. [12]. Let us give some more details on an informal level.

- The tree is parametrized by $(g, h, k) \in \mathbb{G}^3$ and $U_1, \ldots, U_8 \in \mathsf{DLIN}(g, h, k)$, where for each $i \in [8]$ we have $U_i = (g^{u_i}, h^{v_i}, k^{u_i+v_i})$ for random $u_i, v_i \xleftarrow{\$} \mathbb{Z}_p$. (It will later become clear that we will sign vectors of group elements, where each consists of 8 group elements. This is the reason why we choose $n = 8$ here).
- Each tree node N is identified by a four-tuple of group elements $N = (X, z) \in \mathbb{G}^4$, where $z \xleftarrow{\$} \mathbb{G}$ is random and $X = (g^x, h^y, k^{x+y})$ for random $x, y \xleftarrow{\$} \mathbb{Z}_p$.
- To each node $N = (X, z)$ of the tree we assign the public key

$$vk = (g, h, k, U_1, \ldots, U_8, X, z)$$

  with corresponding secret key $sk_N = (u_1, v_1, \ldots, u_8, v_8, x, y)$. Note that this is a valid key pair for the one-time signature scheme from Section 3.2, instantiated such that vectors of 8 group elements can be signed. Note also that each node is identified by $(X, z) \in \mathbb{G}^4$, so that we can sign two child nodes with each public key.
- The tree is constructed — on the fly — as follows. Let $N_L = (X_L, z_L)$ and $N_R = (X_R, z_R)$ be the two children of node $N = (X, z)$. Then a signature of the message $M = (N_L, N_R) = (X_L, z_L, X_R, z_R) \in \mathbb{G}^8$ under secret key sk_N authenticates N_L and N_R as children of N. (This is why we chose $n = 8$).

– This gives the following signature scheme, which can be used to sign 8-tuples of elements of $\mathbb{G}$:
  - The public key of the signature scheme consists of $(g, h, k, U_1, \ldots, U_8)$ and the root node $N_0 = (X_0, z_0)$, the secret key consists of the discrete logarithms $(u_1, v_1, \ldots, u_8, v_8, x_0, y_0)$.
  - In order to sign a message $M = M_d \in \mathbb{G}^8$, select a leaf node N_d which has not been used before. Let $N_{d-1}, \ldots, N_0$ denote the path from N_d to the root N_0, and for all N_i ($i \in \{1, \ldots, d-1\}$), let N_i^{co} denote the sibling of N_i. Let

$$M_{i-1} := \begin{cases} (N_i, N_i^{\mathsf{co}}), & \text{if } N_i^{\mathsf{co}} \text{ is the right-hand sibling of } N_i, \\ (N_i^{\mathsf{co}}, N_i), & \text{if } N_i^{\mathsf{co}} \text{ is the left-hand sibling of } N_i. \end{cases} \tag{2}$$

    A signature for M_d consists of all pairs $M_{d-1}, \ldots, M_0$ and signatures $(\sigma_d, \ldots, \sigma_0)$ such that each signature σ_i authenticates M_i as child of node N_{i-1}.

We note that, strictly speaking, the described scheme is not structure-preserving. (The reason is the case distinction in (2).) We will show later how to implement our scheme in a structure-preserving way. A more precise description as well as a graphical illustration of the scheme can be found in the full version [33].

Theorem 3. *Suppose there exists a non-adaptive adversary $\mathcal{A}$ that (ϵ, t, q)-breaks the EUF-CMA security of* TSig. *Then there exists an adversary $\mathcal{B}$ that (ϵ', t')-breaks the DLIN assumption in $\mathbb{G}$, where t' is roughly the runtime of the EUF-CMA experiment with $\mathcal{A}$, and $\epsilon' = \epsilon/8$.*

Note that the success probability ϵ' of the DLIN-breaker $\mathcal{B}$ is independent of the number q of chosen-message queries issued by $\mathcal{A}$. In the full version [33] we also describe how to construct an adaptively secure scheme with tight reduction.

4 Tightly Simulation-Sound NIZK Proofs for Pairing Product Equations

In this section we use the signature scheme from Section 3.3 to construct a NIZK proof for satisfiability of pairing product equations whose security reduces tightly to the DLIN assumption. "Tight" means here that the success probability of the reduction is independent of the number of simulated proofs the adversary sees. The construction is a special case of Groth-Sahai (GS) proofs [31], and uses a trick from [30, Section 4] to express the disjunction of two sets of pairing product equations as one set.

Non-Interactive Zero-Knowledge Proofs. Let R be a binary relation and let $\mathcal{L} := \{x : \exists w \text{ s.t. } R(x, w) = 1\}$ be the language defined by R. A non-interactive zero-knowledge proof system $\mathsf{NIZK} = (\mathsf{NIZK.Gen}, \mathsf{NIZK.Prove}, \mathsf{NIZK.Vfy})$ for $\mathcal{L}$ consists of three algorithms. The common reference string generation algorithm

$crs \xleftarrow{\$} \mathsf{NIZK.Gen}(\kappa)$ takes as input a security parameter κ and outputs a common reference string crs. Algorithm $\pi \xleftarrow{\$} \mathsf{NIZK.Prove}(crs, x, w)$ takes as input crs, statement x, and a witness w that $x \in \mathcal{L}$, and outputs a proof π. The verification algorithm $\mathsf{NIZK.Vfy}(crs, \pi, x) \in \{0,1\}$ takes as input proof π and statement x. We say that $\mathsf{NIZK.Vfy}$ *accepts* if $\mathsf{NIZK.Vfy}(crs, \pi, x) = 1$. We say that $\mathsf{NIZK.Vfy}$ *rejects* if $\mathsf{NIZK.Vfy}(crs, \pi, x) = 0$.

NIZK is $(\epsilon_{\mathsf{ZK}}, \epsilon_{\mathsf{snd}}, \epsilon_{\mathsf{simsnd}}, t, Q)$-secure, if the following holds.

Perfect completeness. For each $(x, w) \in R$, each parameter κ, and each $crs \xleftarrow{\$} \mathsf{NIZK.Gen}(\kappa)$ holds that

$$\Pr[\mathsf{NIZK.Vfy}(crs, \pi, x) = 1 : \pi \xleftarrow{\$} \mathsf{NIZK.Prove}(crs, x, w)] = 1.$$

Soundness. For all adversaries $\mathcal{A}$ running in time t holds that

$$\Pr[\mathcal{A}(crs) = (x, \pi) : x \notin \mathcal{L} \wedge \mathsf{NIZK.Vfy}(crs, \pi, x) = 1] \leq \epsilon_{\mathsf{snd}}$$

Zero knowledge. There exists a simulator $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1)$, such that $(crs, td) \xleftarrow{\$} \mathcal{S}_0(\kappa)$ generates a common reference string and trapdoor information td, and $\pi \xleftarrow{\$} \mathcal{S}_1(crs_{\mathsf{sim}}, td, x)$ generates a simulated proof π for statement x (where not necessarily $x \in \mathcal{L}$).

Let $crs_{\mathsf{real}} \xleftarrow{\$} \mathsf{NIZK.Gen}(\kappa)$ and let $\mathcal{O}_{\mathsf{real}}$ denote an oracle that takes as input $(x, w) \in R$ and returns $\mathsf{NIZK.Prove}(crs_{\mathsf{real}}, x, w)$. Let $(crs_{\mathsf{sim}}, td) \xleftarrow{\$} \mathcal{S}_0(\kappa)$ and let $\mathcal{O}_{\mathsf{sim}}$ return $\mathcal{S}_1(crs_{\mathsf{sim}}, td, x)$ on input $(x, w) \in R$. We require that

$$\Pr[\mathcal{A}^{\mathcal{O}_{\mathsf{real}}}(crs_{\mathsf{real}}) = 1] - \Pr[\mathcal{A}^{\mathcal{O}_{\mathsf{sim}}}(crs_{\mathsf{sim}}) = 1] \leq \epsilon_{\mathsf{ZK}}$$

for all $\mathcal{A}$ running in time at most t that issue at most Q oracle queries.

Simulation soundness. For $crs \xleftarrow{\$} \mathcal{S}_0(\kappa)$ and for all adversaries $\mathcal{A}$ running in time t that may query $\mathcal{S}_1$ at most Q times for simulated proofs $\pi_1, \ldots, \pi_Q$ of arbitrary statements $x_1, \ldots, x_Q$ (where possibly $x_i \notin \mathcal{L}$ for some or all $i \in [Q]$) holds that

$$\Pr\left[\mathcal{A}^{\mathcal{S}_1}(crs) = (x, \pi) : \begin{array}{c} x \notin \mathcal{L} \wedge (x, \pi) \neq (x_i, \pi_i) \forall i \in [Q] \\ \text{and } \mathsf{NIZK.Vfy}(crs, \pi, x) = 1 \end{array}\right] \leq \epsilon_{\mathsf{simsnd}}$$

We will also use a variant of NIZK proof systems as a technical building block. Namely, a **(perfectly) non-interactive witness-indistinguishable (NIWI) proof system** is defined like a NIZK proof system above, with the following difference. Instead of the zero-knowledge and simulation-soundness properties, we require (perfect) witness-indistinguishability: for all crs in the image of $\mathsf{NIZK.Gen}$, and all $(x, w_1), (x, w_2) \in R$ (for the same x), we require that the distributions induced by $\mathsf{NIZK.Prove}(crs, x, w_1)$ and $\mathsf{NIZK.Prove}(crs, x, w_2)$ are identical.

4.1 Building Blocks

Pairing Product Equations. Following [30, 31], a pairing product equation (PPE) s of length ℓ over $\mathbb{G}$ is an equation of the form

$$\prod_{j=1}^{\ell} e(Q_{j,0}, Q_{j,1}) = 1 \quad \text{with} \quad Q_{j,b} = A_{j,b} \cdot \prod_{i=1}^{\nu} X_i^{\alpha_{j,b,i}} \tag{3}$$

where the $A_i \in \mathbb{G}$ and $\alpha_{j,b,i} \in \mathbb{Z}_p$ are constants, and the $X_i \in \mathbb{G}$ are variables. We say that a vector $\vec{x} = (x_1, \ldots, x_\nu) \in \mathbb{G}^\nu$ *satisfies* the equation, if Equation 3 holds when setting $X_i = x_i$. A set S of pairing product equations is *satisfiable*, if there exists a vector $\vec{x}$ that satisfies all equations $s \in S$ simultaneously. In the following, we will consider sets of satisfiable PPEs as languages for NIZK proof systems.

Disjunctions of Pairing Product Equations. Groth [30, Section 4.8] shows how to express the disjunction of several sets of PPEs through one set. Concretely, given n sets $S_1, \ldots, S_n$ of PPEs, he constructs a set $S := OR(S_1, \ldots, S_n)$ of PPEs such that

- every solution $\vec{x}$ that satisfies S allows to efficiently derive a solution $\vec{x}_i$ of at least one S_i,
- every solution $\vec{x}_i$ of some S_i allows to efficiently derive a solution $\vec{x}$ of S,
- if S_i has ν_i variables X_i and consists of equations of total length ℓ_i, then S has total length $2\ell + 1$ for $\ell = \sum_{i=1}^{n} \ell_i$, and $(\sum_{i=1}^{n} \nu_i) + n + \ell$ variables.

NIWI Proofs for a Set of Pairing Product Equations. Groth and Sahai [31] present an efficient non-interactive witness-indistinguishable proof system for arbitrary sets of PPEs. Their system features a CRS *crs* that can be chosen either to be *hiding* or to be *binding*. If *crs* is hiding, then the resulting proofs are perfectly witness-indistinguishable. If *crs* is binding, the resulting proofs enjoy perfect soundness, and become extractable: a special trapdoor to *crs* allows to extract a witness $\vec{x}$ from a valid proof. Hiding and binding CRSs are computationally indistinguishable under the DLIN assumption in the underlying group $\mathbb{G}$. When implemented over a DLIN-group (as will be the case in our setting), their system has the following efficiency properties (cf. Figure 2 in [31]):

- the CRS contains 6 $\mathbb{G}$-elements,
- each used variable X_i results in 3 $\mathbb{G}$-elements in the proof,
- each PPE incurs 9 $\mathbb{G}$-elements in the proof.

TSig-*Verification as a Set of Pairing Product Equations.* The verification algorithm of our weakly-secure DLIN-based signature scheme TSig from Section 3 can be expressed as a set of PPEs. Concretely, assume a verification key $vk = (g, h, k, U_1, \ldots, U_8, X_0, z_0)$ for TSig, and a message $M = (m_{d,1}, \ldots, m_{d,8}) \in \mathbb{G}^8$. Recall that a TSig-signature $\Sigma \in \mathbb{G}^{10d+2}$ determines OTSig-verification-keys vk_i, messages M_i, and signatures $\Sigma^{(i)}$ such that Σ is valid iff $\Sigma^{(i)}$ is a valid OTSig-signature of $M_i = (m_{i,j})_{j=1}^{8}$ under vk_{i-1} for all $i \in [d]$. Hence, verification amounts to checking a set $S_{vk,M}^{\mathsf{TSig}} = \{OR(S_{L,i}, S_{R,i})\}_{i \in [d]}$, where $S_{D,i}$ (for $i \in [d]$ and $D \in \{L, R\}$) is given by

$$S_{D,i} = \left\{ \left(\prod_{j=1}^{8} E(U_i, m_{i,j}) \right) \cdot E(G, s_i) \cdot E(H, t_i) = E(X_{D,i}, z_{D,i}) \right\}$$

of PPEs, where $G, H, U_1, \ldots, U_8 \in \mathbb{G}^3$ and the $m_{d,j} \in \mathbb{G}$ are constants, and the $m_{i,j} \in \mathbb{G}$ (for $i \in [d-1]$), $X_{L,i}, X_{R,i} \in \mathbb{G}^3$, and $z_{L,i}, z_{R,i}, s_i, t_i \in \mathbb{G}$ (for $i \in [d]$) are variables.

Tightly Secure One-Time Signatures. As a final preparation, we require a means to secure proofs from tampering. Typically, this is done via a one-time signature scheme (as, e.g., in [40]). For our purposes, however, we require *tightly secure* (but not necessarily structure-preserving) one-time signatures. To enable a tight reduction, we will consider signature schemes with an algorithm $\mathsf{TOTS.Param}$ that outputs common system parameters $pars_{\mathsf{tots}}$.

Definition 4. *The security experiment for strong* n-fold *one-time EUF-CMA security is identical to the strong one-time EUF-CMA experiment (see Section 2), except that the adversary* $\mathcal{A}$ *gets the scheme's public parameters and* n *verification keys* pk_i *(*$i \in [n]$*) as input.* $\mathcal{A}$ *may request (up to) one signature for each* pk_i*, and may finally output a forged signature under exactly one* pk_i*. We say that a signature scheme* TOTS *is strongly* n*-fold one-time* (ϵ, t, q)*-secure if there is no* $\mathcal{A}$ *that* (ϵ, t, q)*-breaks the strong EUF-CMA security of* TOTS*.*

We now construct a signature scheme TOTS whose n-fold one-time EUF-CMA security experiment reduces to the discrete logarithm problem in $\mathbb{G}$. The corresponding reduction loses only a factor of 2, independently of n.

$\mathsf{TOTS.Param}(\kappa)$**:** The common parameters $pars_{\mathsf{tots}}$ are a two generators g, h_0 and a collision-resistant hash function $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_p$.

$\mathsf{TOTS.Gen}(pars_{\mathsf{tots}}, pars_{\mathsf{tots}})$**:** Uniformly choose exponents $\omega_1, s_1 \in \mathbb{Z}_p$ and output $vk_{\mathsf{tots}} := (h_1, c_1) := (g^{\omega_1}, g^{s_1})$ and $sk_{\mathsf{tots}} := (\omega_1, s_1)$.

$\mathsf{TOTS.Sign}(pars_{\mathsf{tots}}, sk_{\mathsf{tots}}, m)$**:** Uniformly choose $r_0 \in \mathbb{Z}_p$ and compute $c_0 := g^{\mathsf{H}(m)} h_0^{r_0}$ and $r_1 = (s_1 - \mathsf{H}(c_0))/\omega_1 \bmod p$, such that $c_1 = g^{\mathsf{H}(c_0)} h_1^{r_1}$. Output $\sigma = (r_0, r_1)$.

$\mathsf{TOTS.Vfy}(pars_{\mathsf{tots}}, vk_{\mathsf{tots}}, m, \sigma)$**:** Parse $\sigma =: (r_0, r_1)$, and set $c_0 := g^{\mathsf{H}(m)} h_0^{r_0}$. If $c_1 = g^{\mathsf{H}(c_0)} h_1^{r_1}$, output 1, else 0.

Note that TOTS essentially consists of a two-fold application of Pedersen commitments [43], interpreted as one-time signatures.

Lemma 1. *Let* $n \in \mathbb{N}$*. Then scheme* TOTS *above is strongly* n*-fold one-time EUF-CMA secure assuming* H *is collision-resistant and the discrete logarithm assumption in* $\mathbb{G}$ *holds. Concretely,* $\epsilon_{\mathsf{n\text{-}cma}} \leq 2\epsilon_{\mathsf{dlog}} + \epsilon_{\mathsf{crhf}}$ *for the advantage* $\epsilon_{\mathsf{n\text{-}cma}}$ *of an arbitrary* n*-fold one-time EUF-CMA adversary* $\mathcal{A}$*, and the advantages* ϵ_{dlog} *of a corresponding DLOG-solver* $\mathcal{B}$ *and* ϵ_{crhf} *of a* H*-collision-finder* $\mathcal{C}$*.*

The proof is fairly standard, see [33] for a sketch.

4.2 Our Simulation-Sound NIZK Proof System

We are now ready to describe our proof system for a set S of PPEs. Intuitively, we will prove, using GS NIWI proofs, that *either* S is satisfiable, *or* that we know a TSig-signature for a "suitably unique" value (or both). The "suitably unique"

value will be a verification key for a strongly (and tightly) secure one-time signature scheme. Simulated proofs prove the "or" branch of the statement, using TSig's signing key. Simulation-soundness (and also soundness) follows from the existential unforgeability of TSig, and from the soundness of GS proofs. A bit more formally, consider the following non-interactive proof system:

NIZK.Gen($\mathbb{G}$) outputs a CRS $crs = (crs_{\mathsf{GS}}, vk, pars_{\mathsf{tots}})$, where (a) crs_{GS} is a binding CRS for DLIN-based GS proofs over $\mathbb{G}$, (b) vk is a verification key for TSig, (c) $pars_{\mathsf{tots}}$ are parameters for a strongly n-fold EUF-CMA secure one-time signature scheme TOTS, whose security reduction does not depend on n.

NIZK.Prove($crs, S, \vec{x}$) takes as input a CRS $crs = (crs_{\mathsf{GS}}, vk)$, a set S of PPEs, and a satisfying assignment $\vec{x} = (x_1, \ldots, x_\nu) \in \mathbb{Z}_p^\nu$. Then, NIZK.Prove samples a TOTS keypair $(vk_{\mathsf{tots}}, sk_{\mathsf{tots}})$ and outputs $\pi = (\pi_{\mathsf{GS}}, vk_{\mathsf{tots}}, \sigma_{\mathsf{tots}})$. Here, π_{GS} is a GS proof (using CRS crs_{GS}) for the set $OR(S, S^{\mathsf{TSig}}_{vk, vk_{\mathsf{tots}}})$ of PPEs (as described above), and σ_{tots} is a TOTS-signature under vk_{tots} of π_{GS}.

NIZK.Vfy(crs, S, π) takes a CRS $crs = (crs_{\mathsf{GS}}, vk)$, a set S of PPEs, and a proof π as above, verifies σ_{tots}, and then checks π as a GS proof for the set $OR(S, S^{\mathsf{TSig}}_{vk, vk_{\mathsf{tots}}})$ of PPEs.

Theorem 4. *The proof system* NIZK *just described is* $(\epsilon_{\mathsf{ZK}}, \epsilon_{\mathsf{snd}}, \epsilon_{\mathsf{simsnd}}, t, Q)$*-secure, where* $\epsilon_{\mathsf{ZK}} \leq 2|\epsilon_{\mathsf{GS}}|$ *and* $\epsilon_{\mathsf{snd}}, \epsilon_{\mathsf{simsnd}} \leq \epsilon_{\mathsf{tots}} + \epsilon_{\mathsf{tsig}}$ *for the advantages of suitable adversaries on the indistinguishability of hiding and binding GS CRSs, the strong* Q*-fold one-time EUF-CMA security of* TOTS*, and the weak EUF-CMA security of* TSig*. All constructed adversaries have roughly the same runtime as the zero-knowledge, soundness, resp. simulation-soundness experiments (with adversaries of runtime* t*).*

5 Tight IND-CCA Security in the Multi-user Setting

A public-key encryption scheme consists of four algorithms PKE = (PKE.Param, PKE.Gen, PKE.Enc, PKE.Dec). Parameter generation $\Pi \xleftarrow{\$} \mathsf{PKE.Param}(\kappa)$ takes as input a security parameter κ and outputs parameters Π. The key generation algorithm $(pk, sk) \xleftarrow{\$} \mathsf{PKE.Gen}(\Pi)$ generates, on input Π, a public encryption key pk and a secret decryption key sk. The probabilistic encryption algorithm $c \xleftarrow{\$} \mathsf{PKE.Enc}(pk, m)$ takes as input a public key pk and a message m, and outputs a ciphertext c. The deterministic decryption algorithm $\mathsf{PKE.Dec}(sk, c) \in \{m, \perp\}$ takes as input a secret key sk and a ciphertext c, and outputs a message m or an error symbol $\perp$. We require the usual correctness properties.

The following security experiment, played between a challenger and an adversary $\mathcal{A}$, is based on the multi-user security definition from [9]. The experiment is parametrized by two integers $\mu, q \in \mathbb{N}$.

1. The challenger runs $\Pi \xleftarrow{\$} \mathsf{PKE.Param}(\kappa)$ once and then $\mathsf{PKE.Gen}(\Pi)$ μ times to generate μ key pairs $(pk^{(i)}, sk^{(i)})$, $i \in [\mu]$. Then it tosses a coin $b \xleftarrow{\$} \{0, 1\}$, initializes a list $\mathsf{Clist} := \emptyset$ to the empty list, and defines a counter $j_i := 0$ for each $i \in [\mu]$.

2. The adversary receives the public keys $pk^{(1)}, \ldots, pk^{(\mu)}$ as input. It may query the challenger for two types of operations.
   Encryption queries. The adversary submits two messages m_0, m_1 and an index $i \in [\mu]$. If $j_i \geq q$ then the challenger returns $\perp$. Otherwise it encrypts m_b under public key $pk^{(i)}$ by computing $c = \mathsf{PKE.Enc}(pk^{(i)}, m_b)$. Then it appends (c, i) to Clist, updates counter j_i as $j_i := j_i + 1$, and returns c.
   Decryption queries. The adversary submits a ciphertext c and an index $i \in [\mu]$. If $(c, i) \in \mathsf{Clist}$ then the challenger returns $\perp$. Otherwise it returns whatever $\mathsf{PKE.Dec}(sk^{(i)}, c)$ returns.
3. Eventually the adversary $\mathcal{A}$ outputs a bit b'. We say that the adversary *wins* the game, if $b = b'$.

Definition 5. *Let $\mathcal{A}$ be an adversary that runs in time t and wins with probability $1/2+\epsilon$. Then $\mathcal{A}$ (ϵ, t)-*breaks the (μ, q)-IND-CCA security *of* PKE*. If $\mathcal{A}$ never asks any decryption query, then $\mathcal{A}$ (ϵ, t)-*breaks the (μ, q)-IND-CPA security *of* PKE*. For ATK $\in \{CPA, CCA\}$ we say that* PKE *is* (ϵ, t, μ, q)-IND-ATK secure*, if there exists no adversary that (ϵ, t)-breaks the (μ, q)-IND-ATK security of* PKE*.*

Note that for ATK $\in$ {CPA,CCA} the classical definitions of IND-ATK security [27, 44] are identical to $(1, 1)$-IND-ATK security in the above sense. Moreover, the generic reduction from [9] shows that an adversary $\mathcal{A}$ that (ϵ, t)-breaks the (μ, q)-IND-ATK security of public-key encryption scheme PKE implies an adversary $\mathcal{A}'$ that (ϵ', t')-breaks the $(1, 1)$-IND-ATK security of PKE with $t' \approx t$ and $\epsilon' \geq \epsilon/(q\mu)$. Thus, the generic reduction looses a factor of $q\mu$.

Generic Construction. The construction of the public-key encryption scheme $\mathsf{Enc_{CCA}}$ with tight security reduction follows the Naor-Yung paradigm [42, 45, 40]. It uses as building blocks an (IND-CPA secure) public-key encryption scheme $\mathsf{Enc_{CPA}} = (\mathsf{CPA.Param}, \mathsf{CPA.Gen}, \mathsf{CPA.Enc}, \mathsf{CPA.Dec})$ and a (simulation-sound) non-interactive zero-knowledge proof system $\mathsf{NIZK} = (\mathsf{NIZK.Gen}, \mathsf{NIZK.Prove}, \mathsf{NIZK.Vfy})$. We let scheme $\mathsf{Enc_{CCA}} = (\mathsf{CCA.Param}, \mathsf{CCA.Gen}, \mathsf{CCA.Enc}, \mathsf{CCA.Dec})$ be as follows.

$\mathsf{CCA.Param}(\kappa)$ generates a common reference string for the NIZK proof system $crs \xleftarrow{\$} \mathsf{NIZK.Gen}(\kappa)$ for the language $\mathcal{L} := \{(pk_0, pk_1, c_0, c_1)\}$ such that $(pk_0, pk_1, c_0, c_1) \in \mathcal{L}$ if and only if

$$c_0 = \mathsf{CPA.Enc}(pk_0, m) \wedge c_1 = \mathsf{CPA.Enc}(pk_1, m).$$

That is, we have $(pk_0, pk_1, vk_{\mathsf{ots}}, c_0, c_1, c_{\Pi}) \in \mathcal{L}$ iff c_0 and c_1 encrypt the same message m.

$\mathsf{CCA.Gen}(\Pi)$ generates two key pairs $(pk_0, sk_0), (pk_1, sk_1) \xleftarrow{\$} \mathsf{CPA.Gen}$ of the public-key encryption scheme. The resulting public key is $pk = (pk_0, pk_1, \Pi)$, the secret key is $sk = sk_0$.

$\mathsf{CCA.Enc}(pk, m)$ encrypts a message m by computing $c_0 = \mathsf{CPA.Enc}(pk_0, m)$, $c_1 = \mathsf{CPA.Enc}(pk_1, m)$, and a proof π that $(pk_0, pk_1, c_0, c_1) \in \mathcal{L}$, using the

encryption randomness of c_0 and c_1 as witness. The resulting ciphertext is $c = (c_0, c_1, \pi)$.

$\mathsf{CCA.Dec}(sk, c)$ decrypts a given ciphertext as follows. First it checks whether $(pk_0, pk_1, c_0, c_1) \in \mathcal{L}$ by verifying the proof π. If false, then it returns $\perp$. Otherwise it computes and returns $m = \mathsf{CPA.Dec}(sk_0, c_0)$.

It is a classical result [45] that the above encryption scheme is $(1,1)$-IND-CCA secure, if $\mathsf{Enc_{CPA}}$ is $(1,1)$-IND-CPA secure and NIZK is one-time simulation sound. In the sequel we generalize this to showing that the (μ, q)-IND-CCA security of $\mathsf{Enc_{CCA}}$ reduces *tightly* (i.e., independent of μ and q) to the (μ, q)-IND-CPA security of $\mathsf{Enc_{CPA}}$ and the μq-security of NIZK.

We remark that our NIZK proof system from Section 4 inherits the (witness)-extractability of Groth-Sahai proofs. Hence, one could think of treating the NIZK system NIZK as one instance of an IND-CPA secure PKE scheme (with the extraction trapdoor as decryption key). It would seem natural to expect that a variant of scheme $\mathsf{Enc_{CCA}}$ above with only one $\mathsf{Enc_{CPA}}$ instance might be IND-CCA secure. We do not know if this holds, however: concretely, to show witness-indistinguishability of our proof system NIZK, we will at some point need to switch NIZK into hiding mode. In this mode, no extraction trapdoor exists, and it is unclear how to go answer decryption queries for $\mathsf{Enc_{CCA}}$.

Theorem 5. *Let* $\mathsf{Enc_{CPA}}$ *be* $(\epsilon_{\mathsf{CPA}}, t_{\mathsf{CPA}}, \mu, q)$*-IND-CPA secure, and let* NIZK *be* $(\epsilon_{\mathsf{ZK}}, \epsilon_{\mathsf{snd}}, \epsilon_{\mathsf{simsnd}}, t_{\mathsf{NIZK}}, \mu q)$*-secure. Then* $\mathsf{Enc_{CCA}}$ *is* (ϵ, t, μ, q)*-IND-CCA secure, where* t_{NIZK} *and* t_{CPA} *are roughly the runtime of the IND-CCA experiment with an adversary of runtime* t*, and* $\epsilon \leq 2 \cdot (\epsilon_{\mathsf{CPA}} + \epsilon_{\mathsf{ZK}}) + \epsilon_{\mathsf{snd}} + \epsilon_{\mathsf{simsnd}}$.

In order to instantiate the CCA-secure scheme from the previous section, we finally need an IND-CPA secure encryption scheme with tight security reduction. A well-known construction can be found in [33].

References

[1] Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-Preserving Signatures and Commitments to Group Elements. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 209–236. Springer, Heidelberg (2010)

[2] Abe, M., Haralambiev, K., Ohkubo, M.: Signing on elements in bilinear groups for modular protocol design. Cryptology ePrint Archive, Report 2010/133 (2010), `http://eprint.iacr.org/`

[3] Abe, M., Groth, J., Haralambiev, K., Ohkubo, M.: Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 649–666. Springer, Heidelberg (2011)

[4] Abe, M., Chase, M., David, B., Kohlweiss, M., Nishimaki, R., Ohkubo, M.: Constant-size structure-preserving signatures: Generic constructions and simple assumptions. Cryptology ePrint Archive, Report 2012/285 (2012), `http://eprint.iacr.org/`

[5] Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993, pp. 62–73. ACM Press (November 1993)

[6] Bellare, M., Shoup, S.: Two-Tier Signatures, Strongly Unforgeable Signatures, and Fiat-Shamir Without Random Oracles. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 201–216. Springer, Heidelberg (2007)
[7] Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: 38th FOCS, pp. 394–403. IEEE Computer Society Press (October 1997)
[8] Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among Notions of Security for Public-Key Encryption Schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 26–45. Springer, Heidelberg (1998)
[9] Bellare, M., Boldyreva, A., Micali, S.: Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Heidelberg (2000)
[10] Bernstein, D.J.: Proving Tight Security for Rabin-Williams Signatures. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 70–87. Springer, Heidelberg (2008)
[11] Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
[12] Boneh, D., Mironov, I., Shoup, V.: A Secure Signature Scheme from Bilinear Maps. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 98–110. Springer, Heidelberg (2003)
[13] Camenisch, J., Chandran, N., Shoup, V.: A Public Key Encryption Scheme Secure against Key Dependent Chosen Plaintext and Adaptive Chosen Ciphertext Attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 351–368. Springer, Heidelberg (2009)
[14] Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. Journal of Cryptology 20(3), 265–294 (2007)
[15] Cathalo, J., Libert, B., Yung, M.: Group Encryption: Non-interactive Realization in the Standard Model. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 179–196. Springer, Heidelberg (2009)
[16] Chase, M., Kohlweiss, M.: A domain transformation for structure-preserving signatures on group elements. Cryptology ePrint Archive, Report 2011/342 (2011), http://eprint.iacr.org/
[17] Chevallier-Mames, B., Joye, M.: A Practical and Tightly Secure Signature Scheme Without Hash Function. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 339–356. Springer, Heidelberg (2007)
[18] Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
[19] Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
[20] Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM Journal on Computing 30(2), 391–437 (2000)
[21] ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory 31, 469–472 (1985)
[22] Fuchsbauer, G.: Automorphic Signatures and Applications. PhD thesis, ENS, Paris (2010)
[23] Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)

[24] Gennaro, R., Halevi, S., Rabin, T.: Secure Hash-and-Sign Signatures without the Random Oracle. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 123–139. Springer, Heidelberg (1999)
[25] Gentry, C.: Practical Identity-Based Encryption Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
[26] Gentry, C., Halevi, S.: Hierarchical Identity Based Encryption with Polynomially Many Levels. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 437–456. Springer, Heidelberg (2009)
[27] Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
[28] Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing 17(2), 281–308 (1988)
[29] Green, M., Hohenberger, S.: Practical Adaptive Oblivious Transfer from Simple Assumptions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 347–363. Springer, Heidelberg (2011)
[30] Groth, J.: Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)
[31] Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
[32] Hofheinz, D.: All-But-Many Lossy Trapdoor Functions. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 209–227. Springer, Heidelberg (2012)
[33] Hofheinz, D., Jager, T.: Tightly secure signatures and public-key encryption. Cryptology ePrint Archive (2012), http://eprint.iacr.org/
[34] Hofheinz, D., Kiltz, E.: Secure Hybrid Encryption from Weakened Key Encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (2007)
[35] Hofheinz, D., Kiltz, E.: Practical Chosen Ciphertext Secure Encryption from Factoring. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 313–332. Springer, Heidelberg (2009)
[36] Joye, M.: An Efficient On-Line/Off-Line Signature Scheme without Random Oracles. In: Franklin, M.K., Hui, L.C.K., Wong, D.S. (eds.) CANS 2008. LNCS, vol. 5339, pp. 98–107. Springer, Heidelberg (2008)
[37] Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: Jajodia, S., Atluri, V., Jaeger, T. (eds.) ACM CCS 2003, pp. 155–164. ACM Press (October 2003)
[38] Kurosawa, K., Desmedt, Y.: A New Paradigm of Hybrid Encryption Scheme. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Heidelberg (2004)
[39] Lewko, A., Waters, B.: New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
[40] Lindell, Y.: A Simpler Construction of CCA2-Secure Public-Key Encryption Under General Assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 241–254. Springer, Heidelberg (2003)
[41] Merkle, R.C.: A Certified Digital Signature. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 218–238. Springer, Heidelberg (1990)

[42] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd ACM STOC. ACM Press (May 1990)
[43] Pedersen, T.P.: Non-interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)
[44] Rackoff, C., Simon, D.R.: Non-interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
[45] Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th FOCS, pp. 543–553. IEEE Computer Society Press (October 1999)
[46] Schäge, S.: Tight Proofs for Signature Schemes without Random Oracles. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 189–206. Springer, Heidelberg (2011)
[47] Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)