More Efficient (Almost) Tightly Secure Structure-Preserving Signatures

Romain Gay[1,2(✉)], Dennis Hofheinz[3], Lisa Kohl[3], and Jiaxin Pan[3]

[1] Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, Paris, France
rgay@di.ens.fr
[2] INRIA, Paris, France
[3] Karlsruhe Institute of Technology, Karlsruhe, Germany
{Dennis.Hofheinz,Lisa.Kohl,Jiaxin.Pan}@kit.edu

Abstract. We provide a structure-preserving signature (SPS) scheme with an (almost) tight security reduction to a standard assumption. Compared to the state-of-the-art tightly secure SPS scheme of Abe et al. (CRYPTO 2017), our scheme has smaller signatures and public keys (of about 56%, resp. 40% of the size of signatures and public keys in Abe et al.'s scheme), and a lower security loss (of $\mathbf{O}(\log Q)$ instead of $\mathbf{O}(\lambda)$, where λ is the security parameter, and $Q = \mathsf{poly}(\lambda)$ is the number of adversarial signature queries).

While our scheme is still less compact than structure-preserving signature schemes *without* tight security reduction, it significantly lowers the price to pay for a tight security reduction. In fact, when accounting for a non-tight security reduction with larger key (i.e., group) sizes, the computational efficiency of our scheme becomes at least comparable to that of non-tightly secure SPS schemes.

Technically, we combine and refine recent existing works on tightly secure encryption and SPS schemes. Our technical novelties include a modular treatment (that develops an SPS scheme out of a basic message authentication code), and a refined hybrid argument that enables a lower security loss of $\mathbf{O}(\log Q)$ (instead of $\mathbf{O}(\lambda)$).

Keywords: Structure-preserving signatures · Tight security

R. Gay—Supported by ERC Project aSCEND (639554), and a Google PhD fellowship.

D. Hofheinz—Supported by ERC Project PREP-CRYPTO (724307), and by DFG grants HO 4534/4-1 and HO 4534/2-2.

L. Kohl—Supported by ERC Project PREP-CRYPTO (724307), and by DFG grant HO 4534/2-2.

J. Pan—Supported by DFG grant HO 4534/4-1.

J. B. Nielsen and V. Rijmen (Eds.): EUROCRYPT 2018, LNCS 10821, pp. 230–258, 2018.
https://doi.org/10.1007/978-3-319-78375-8_8

1 Introduction

Structure-Preserving Signatures (SPSs). Informally, a cryptographic scheme (such as an encryption or signature scheme) is called structure-preserving if its operation can be expressed using equations over a (usually pairing-friendly) cyclic group. A structure-preserving scheme has the advantage that we can reason about it with efficient zero-knowledge proof systems such as the Groth-Sahai non-interactive zero-knowledge (NIZK) system [31]. This compatibility is the key to constructing efficient anonymous credential systems (e.g., [10]), and can be extremely useful in voting schemes and mix-nets (e.g., [30]).

In this work, we are concerned with structure-preserving signature (SPS) schemes. Since popular tools such as "structure-breaking" collision-resistant hash functions cannot be used in a structure-preserving scheme, constructing an SPS scheme is a particularly challenging task. Still, there already exist a variety of SPS schemes in the literature [2,4–6,17–19,29,35,37,39,44] (see also Table 1 for details on some of them).

Tight Security for SPS Schemes. A little more specifically, in this work we are interested in *tightly secure* SPS schemes. Informally, a cryptographic scheme is tightly secure if it enjoys a tight security reduction, i.e., a security reduction that transforms any adversary $\mathcal{A}$ on the scheme into a problem-solver with about the same runtime and success probability as $\mathcal{A}$, *independently* of the number of uses of the scheme.[1] A tight security reduction gives security guarantees that do not degrade in the size of the setting in which the scheme is used.

Specifically, tight security reductions allow to give "universal" keylength recommendations that do not depend on the envisioned size of an application. This is useful when deploying an application for which the eventual number of uses cannot be reasonably bounded a priori. Moreover, this point is particularly vital for SPS schemes. Namely, an SPS scheme is usually combined with several other components that all use the same cyclic group. Thus, a keylength increase (which implies changing the group, and which might be necessary for a non-tightly secure scheme for which a secure keylength depends on the number of uses) affects several schemes, and is particularly costly.

In recent years, progress has been made in the construction of a variety of tightly[2] secure cryptographic schemes such as public-key encryption schemes [11,25,33–35,42,43], identity-based encryption schemes [8,14,20,21,27,36], and signature schemes [3,6,14,16,21,34,35,42]. However, somewhat surprisingly, only few SPS schemes with tight security reductions are known. Moreover, these

[1] We are only interested in reductions to well-established and plausible computational problems here. While the security of any scheme can be trivially (and tightly) reduced to the security of that same scheme, such a trivial reduction is of course not very useful.

[2] Most of the schemes in the literature are only "almost" tightly secure, meaning that their security reduction suffers from a small multiplicative loss (that however is independent of the number of uses of the scheme). In the following, we will not make this distinction, although we will of course be precise in the description and comparison of the reduction loss of our own scheme.

tightly secure SPS schemes [6,35] are significantly less efficient than either "ordinary" SPS or tightly secure signature schemes (see Table 1). One reason for this apparent difficulty to construct tightly secure SPS schemes is that tight security appears to require dedicated design techniques (such as a sophisticated hybrid argument over the bits of an IBE identity [21]), and most known such techniques cannot be expressed in a structure-preserving manner.

Table 1. Comparison of standard-model SPS schemes (in their most efficient variants). We list unilateral schemes (with messages over one group) and bilateral schemes (with messages over both source groups of a pairing) separately. The notation (x_1, x_2) denotes x_1 elements in $\mathbb{G}_1$ and x_2 elements in $\mathbb{G}_2$. $|M|$, $|\sigma|$, and $|pk|$ denote the size of messages, signatures, and public keys (measured in group elements). "Sec. loss" denotes the multiplicative factor that the security reduction to "Assumption" loses, where we omit dominated and additive factors. (Here, "generic" means that only a proof in the generic group model is known.) For the tree-based scheme HJ12, ℓ denotes the depth of the tree (which limits the number of signing queries to 2^ℓ). Q denotes the number of adversarial signing queries, and λ is the security parameter.

Scheme	$\lvert M\rvert$	$\lvert\sigma\rvert$	$\lvert pk\rvert$	Sec. loss	Assumption
HJ12 [35]	1	$10\ell + 6$	13	8	DLIN
ACDKNO16 [2]	$(n_1, 0)$	$(7, 4)$	$(5, n_1 + 12)$	Q	SXDH, XDLIN
LPY15 [44]	$(n_1, 0)$	$(10, 1)$	$(16, 2n_1 + 5)$	$\mathbf{O}(Q)$	SXDH, XDLINX
KPW15 [39]	$(n_1, 0)$	$(6, 1)$	$(0, n_1 + 6)$	$2Q^2$	SXDH
JR17 [37]	$(n_1, 0)$	$(5, 1)$	$(0, n_1 + 6)$	$Q \log Q$	SXDH
AHNOP17 [6]	$(n_1, 0)$	$(13, 12)$	$(18, n_1 + 11)$	80λ	SXDH
Ours (unilateral)	$(n_1, 0)$	$(8, 6)$	$(2, n_1 + 9)$	$6 \log Q$	SXDH
AGHO11 [5]	(n_1, n_2)	$(2, 1)$	$(n_1, n_2 + 2)$	—	Generic
ACDKNO16 [2]	(n_1, n_2)	$(8, 6)$	$(n_2 + 6, n_1 + 13)$	Q	SXDH, XDLIN
KPW15 [39]	(n_1, n_2)	$(7, 3)$	$(n_2 + 1, n_1 + 7)$	$2Q^2$	SXDH
AHNOP17 [6]	(n_1, n_2)	$(14, 14)$	$(n_2 + 19, n_1 + 12)$	80λ	SXDH
Ours (bilateral)	(n_1, n_2)	$(9, 8)$	$(n_2 + 4, n_1 + 9)$	$6 \log Q$	SXDH

1.1 Our Contribution

Overview. We present a tightly secure SPS scheme with significantly improved efficiency and tighter security reduction compared to the state-of-the-art tightly secure SPS scheme of Abe et al. [6]. Specifically, our signatures contain 14 group elements (compared to 25 group elements in [6]), and our security reduction loses a factor of only $\mathbf{O}(\log Q)$ (compared to $\mathbf{O}(\lambda)$), where λ denotes the security parameter, and $Q = \mathsf{poly}(\lambda)$ denotes the number of adversarial signature queries. When accounting for loose reductions through an appropriate keylength increase, the computational efficiency of our scheme even compares favorably to that of state-of-the-art non-tightly secure SPS schemes.

In the following, we will detail how we achieve our results, and in particular the progress we make upon previous techniques. We will also compare our work to existing SPS schemes (both tightly and non-tightly secure).

Central Idea: A Modular Treatment. A central idea in our work (that in particular contrasts our approach to the one of Abe et al.) is a *modular* construction. That is, similar to the approach to tight IBE security of Blazy, Kiltz, and Pan [14], the basis of our construction is a tightly secure message authentication code (MAC). This tightly secure MAC will then be converted into a signature scheme by using NIZK proofs, following (but suitably adapting) the generic MAC-to-signatures conversion of Bellare and Goldwasser [12].

Starting Point: A Tightly Secure MAC. Our tightly secure MAC will have to be structure-preserving, so the MAC used in [14] cannot be employed in our case. Instead, we derive our MAC from the recent tightly secure key encapsulation mechanism (KEM) of Gay, Hofheinz, and Kohl [26] (which in turn builds upon the Kurosawa-Desmedt PKE scheme [41]). To describe their scheme, we assume a group $\mathbb{G} = \langle g \rangle$ of prime order p, and we use the implicit notation $[x] := g^x$ from [24]. We also fix an integer k that determines the computational assumption to which we want to reduce.[3] Now in (a slight simplification of) the scheme of [26], a ciphertext C with corresponding KEM key K is of the form

$$C = ([\mathbf{t}], \pi), \qquad K = [(\mathbf{k}_0 + \mu \mathbf{k}_1)^\top \mathbf{t}] \quad (\text{for } \mu = H([\mathbf{t}])), \tag{1}$$

where H is a collision-resistant hash function, and $\mathbf{k}_0, \mathbf{k}_1, \mathbf{t} \in \mathbb{Z}_p^{2k}$ and π are defined as follows. First, $\mathbf{k}_0, \mathbf{k}_1 \in \mathbb{Z}_p^{2k}$ comprise the secret key. Next, $\mathbf{t} = \mathbf{A}_0 \mathbf{r}$ for a fixed matrix $\mathbf{A}_0$ (given as $[\mathbf{A}_0]$ in the public key) and a random vector $\mathbf{r} \in \mathbb{Z}_p^k$ chosen freshly for each encryption. Finally, π is a NIZK proof that proves that $\mathbf{t} \in \mathrm{span}(\mathbf{A}_0) \cup \mathrm{span}(\mathbf{A}_1)$ for another fixed matrix $\mathbf{A}_1$ (also given as $[\mathbf{A}_1]$ in the public key). The original Kurosawa-Desmedt scheme [41] is identical, except that π is omitted, and $k = 1$. Hence, the main benefit of π is that it enables a tight security reduction.[4]

We can view this KEM as a MAC scheme simply by declaring the MAC tag for a message M to be the values (C, K) from (1), only with $\mu := M$ (instead of $\mu = H([\mathbf{t}])$). The verification procedure of the resulting MAC will check π, and then check whether C really decrypts to K. (Hence, MAC verification still requires the secret key $\mathbf{k}_0, \mathbf{k}_1$.) Now a slight adaptation of a generic argument of Dodis et al. [22] reduces the security of this MAC tightly to the security of the underlying KEM scheme. Unfortunately, this resulting MAC is not structure-preserving yet (even if the used NIZK proof π is): the message $M = \mu$ is a scalar (from $\mathbb{Z}_p$).[5]

[3] For $k = 1$, we can reduce to DDH in $\mathbb{G}$, and for $k > 1$, we can reduce to the k-Linear assumption, and in fact even to the weaker Matrix-DDH assumption [24].

[4] Actually, the scheme of [26] uses an efficient designated-verifier NIZK proof π that is however not structure-preserving (and thus not useful for our case), and also induces an additional term in K. For our purposes, we can think of π as a (structure-preserving) Groth-Sahai proof.

[5] A structure-preserving scheme should have group elements (and not scalars) as messages, since Groth-Sahai proofs cannot (easily) be used to prove knowledge of scalars.

Abstracting Our Strategy into a Single "core lemma". We can distill the essence of the security proof of our MAC above into a single "core lemma". This core lemma forms the heart of our work, and shows how to randomize all tags of our MAC. While this randomization follows a previous paradigm called "adaptive partitioning" (used to prove the tight security of PKE [26,33] and SPS schemes [6]), our core lemma induces a much smaller reduction loss. The reason for this smaller reduction loss is that previous works on tightly secure schemes (including [6,26,33]) conduct their reduction along the individual bits of a certain hash value (or message to be signed). Since this hash value (or message) usually has $\mathbf{O}(\lambda)$ bits, this induces a hybrid argument of $\mathbf{O}(\lambda)$ steps, and thus a reduction loss of $\mathbf{O}(\lambda)$. In contrast, we conduct our security argument along the individual bits of the *index* of a signing query (i.e., a value from 1 to Q, where Q is the number of signing queries). This index exists only in the security proof, and can thus be considered as an "implicit" way to structure our reduction.[6]

From MACs to Signatures and Structure-Preserving Signatures. Fortunately, our core lemma can be used to prove not only our MAC scheme, but also a suitable signature and SPS scheme tightly secure. To construct a signature scheme, we can now use an case-tailored (and heavily optimized) version of the generic transformation of Bellare and Goldwasser [12]. In a nutshell, that transformation turns a MAC tag (that requires a secret key to verify) into a publicly verifiable signature by adding a NIZK proof to the tag that proves its validity, relative to a public commitment to the secret key. For our MAC, we only need to prove that the given key K really is of the form $K = [(\mathbf{k}_0 + \mu\mathbf{k}_1)^\top \mathbf{t}]$. This linear statement can be proven with a comparatively simple and efficient NIZK proof π'. For $k = 1$, an optimized Groth-Sahai-based implementation of π, and an implicit π' (that uses ideas from [38,40]), the resulting signature scheme will have signatures that contain 14 group elements.

To turn our scheme into an SPS scheme, we need to reconsider the equation $K = [(\mathbf{k}_0 + \mu\mathbf{k}_1)^\top \mathbf{t}]$ from (1). In our MAC (and also in the signature scheme above), we have set $\mu = M \in \mathbb{Z}_p$, which we cannot afford to do for an SPS scheme. Our solution consists in choosing a different equation that fulfills the following requirements:

(a) it is algebraic (in the sense that it integrates a message $M \in \mathbb{G}$), and
(b) it is compatible with our core lemma (so it can be randomized quickly).

For our scheme, we start from the equation

$$K = [\mathbf{k}_0^\top \mathbf{t} + \mathbf{k}^\top \begin{pmatrix} M \\ 1 \end{pmatrix}] \tag{2}$$

for uniform keys $\mathbf{k}_0, \mathbf{k}$. We note that a similar equation has already been used by Kiltz, Pan, and Wee [39] for constructing SPS schemes, although with a very

[6] A reduction loss of $\mathbf{O}(\log Q)$ has been achieved in the context of IBE schemes [20], but their techniques are different and rely on a composite-order group.

different and non-tight security proof. We can plug this equation into the MAC-to-signature transformation sketched above, to obtain an SPS scheme with only 14 group elements (for $k = 1$) per signature.

Our security proof will directly rely on our core lemma to first randomize the $\mathbf{k}_0^\top \mathbf{t}$ part of (2) in all signatures. After that, similar to [39], an information-theoretic argument (that only uses the pairwise independence of the second part of (2), when viewed as a function of M) shows security.

Our basic SPS scheme is unilateral, i.e., its messages are vectors over only one source group of a given pairing. To obtain a bilateral scheme that accepts "mixed" messages over both source groups of an asymmetric pairing, we can use a generic transformation of [39] that yields a bilateral scheme with signatures of 17 group elements (for $k = 1$).

1.2 Related Work and Efficiency Comparison

In this subsection, we compare our work to the closest existing work (namely, the tightly secure SPS scheme of Abe et al. [6]) and other, non-tightly secure SPS schemes.

Comparison to the Work of Abe et al. The state of the art in tightly secure SPS schemes (and in fact currently the only other efficient tightly secure SPS scheme) is the recent work of Abe et al. [6]. Technically, their scheme also uses a tightly secure PKE scheme (in that case [33]) as an inspiration. However, there are also a number of differences in our approaches which explain our improved efficiency and reduction.

Table 2. Comparison of the computational efficiency of state-of-the-art SPS schemes (in their most efficient, SXDH-based variants) with our SXDH-based schemes in the unilateral (UL) and bilateral (BL) version. With "PPEs" and "Pairings", we denote the number of those operations necessary during verification, where "batched" denotes optimized figures obtained by "batching" verification equations [13]. The "$|M|$" and "Sec. loss" columns have the same meaning as in Table 1. The column "$|\mathbb{G}_1|$" denotes the (bit)size of elements from the first source group in a large but realistic scenario (under some simplifying assumptions), see the discussion in Sect. 1.2. "$|\sigma|$ (bits)" denotes the resulting overall signature size, where we assume that the bitsize of $\mathbb{G}_2$ elements is twice the bitsize of $\mathbb{G}_1$-elements.

Scheme	$\lvert M\rvert$	PPEs	Pairings (plain)	Pairings (batched)	Sec. loss	$\lvert\mathbb{G}_1\rvert$ (bits)	$\lvert\sigma\rvert$ (bits)
KPW [39]	$(n_1, 0)$	3	$n_1 + 11$	$n_1 + 10$	$2Q^2$	322	2576
JR [37]	$(n_1, 0)$	2	$n_1 + 8$	$n_1 + 6$	$Q \log Q$	270	1890
AHNOP [6]	$(n_1, 0)$	15	$n_1 + 57$	$n_1 + 16$	80λ	226	8362
Ours (UL)	$(n_1, 0)$	6	$n_1 + 29$	$n_1 + 11$	$6 \log Q$	216	4320
KPW [39]	(n_1, n_2)	4	n_1+n_2+15	n_1+n_2+14	$2Q^2$	322	4186
AHNOP [6]	(n_1, n_2)	16	n_1+n_2+61	n_1+n_2+18	80λ	226	9492
Ours (BL)	(n_1, n_2)	7	n_1+n_2+33	n_1+n_2+15	$6 \log Q$	216	5400

First, Abe et al.'s scheme involves more (and more complex) NIZK proofs, since they rather closely follow the PKE scheme from [33]. This leads to larger proofs and thus larger signatures. Instead, our starting point is the much simpler scheme of [26] (which only features one comparatively simple NIZK proof in its ciphertext).

Second, while the construction of Abe et al. is rather monolithic, our construction can be explained as a modification of a simple MAC scheme. Our approach thus allows for a more modular exposition, and in particular we can outsource the core of the reduction into a core lemma (as explained above) that can be applied to MAC, signature, and SPS scheme.

Third, like previous tightly secure schemes (and in particular the PKE schemes of [26,33]), Abe et al. conduct their security reduction along the individual bits of a certain hash value (or message to be signed). As explained above, our reduction is more economic, and uses a hybrid argument over an "implicit" counter value.

Efficiency Comparison. We give a comparison to other SPS schemes in Table 1. This table shows that our scheme is still significantly less efficient *in terms of signature size* than existing, non-tightly secure SPS schemes. However, when considering *computational efficiency*, and when accounting for a larger security loss in the reduction with larger groups, things look differently.

The currently most efficient non-tightly secure SPS schemes are due to Jutla and Roy [37] and Kiltz, Pan, and Wee [39]. Table 2 compares the computational complexity of their verification operation with the tightly secure SPSs of Abe et al. and our schemes. Now consider a large scenario with $Q = 2^{30}$ signing queries and a target security parameter of $\lambda = 100$. Assume further that we use groups that only allow generic attacks (that require time about the square root of the group size). This means that we should run a scheme in a group of size at least $2^{2(\lambda+\log L)}$, where L denotes the multiplicative loss of the respective security reduction. Table 2 shows the resulting group sizes in column "$|\mathbb{G}_1|$" (in bits, such that $|\mathbb{G}_1| = 200$ denotes a group of size 2^{200}).

Now very roughly, the computational complexity of pairings can be assumed to be cubic in the (bit)size of the group [7,9,23,28]. Hence, in the unilateral setting, and assuming an optimized verification implementation (that uses "batching" [13]) the computational efficiency of the verification in our scheme is roughly on par with that in the (non-tightly secure) state-of-the-art scheme of Jutla and Roy [37], even for small messages. For larger messages, our scheme becomes preferable. In the bilateral setting, our scheme is clearly the most efficient known scheme.

Roadmap

We fix some notation and recall some preliminaries in Sect. 2. In Sect. 3, we present our basic MAC and prove it secure (using the mentioned core lemma). In Sects. 4 and 5, we present our signature and SPS schemes. Due to lack of space, for some proofs (including the more technical parts of the proof of the core lemma, and a full proof for the signature scheme) we refer to the full version.

2 Preliminaries

In this section we provide the preliminaries which our paper builds upon. First, we want to give an overview of notation used throughout all sections.

2.1 Notation

By $\lambda \in \mathbb{N}$ we denote the security parameter. We always employ $\mathsf{negl}\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ to denote a negligible function, that is for all polynomials $p \in \mathbb{N}[X]$ there exists an $n_0 \in \mathbb{N}$ such that $\mathsf{negl}(n) < 1/p(n)$ for all $n \geq n_0$. For any set $\mathcal{S}$, by $s \leftarrow_R \mathcal{S}$ we set s to be a uniformly at random sampled element from $\mathcal{S}$. For any distribution $\mathcal{D}$ by $d \leftarrow \mathcal{D}$ we denote the process of sampling an element d according to the distribution $\mathcal{D}$. For any probabilistic algorithm $\mathcal{B}$ by $\mathrm{out} \leftarrow \mathcal{B}(\mathrm{in})$ by out we denote the output of $\mathcal{B}$ on input in. For a deterministic algorithm we sometimes use the notation $\mathrm{out} := \mathcal{B}(\mathrm{in})$ instead. By p we denote a prime throughout the paper. For any element $m \in \mathbb{Z}_p$, we denote by $m_i \in \{0,1\}$ the i-th bit of m's bit representation and by $m_{|i} \in \{0,1\}^i$ the bit string comprising the first i bits of m's bit representation.

It is left to introduce some notation regarding matrices. To this end let $k, \ell \in \mathbb{N}$ such that $\ell > k$. For any matrix $\mathbf{A} \in \mathbb{Z}_p^{\ell \times k}$, we write

$$\mathrm{span}(\mathbf{A}) := \{\mathbf{A}\mathbf{r} \mid \mathbf{r} \in \mathbb{Z}_p^k\} \subset \mathbb{Z}_p^\ell,$$

to denote the *span* of $\mathbf{A}$.

For a full rank matrix $\mathbf{A} \in \mathbb{Z}_p^{\ell \times k}$ we denote by $\mathbf{A}^\perp$ a matrix in $\mathbb{Z}_p^{\ell \times (\ell - k)}$ with $\mathbf{A}^\top \mathbf{A}^\perp = \mathbf{0}$ and rank $\ell - k$. We denote the set of all matrices with these properties as

$$\mathsf{orth}(\mathbf{A}) := \{\mathbf{A}^\perp \in \mathbb{Z}_p^{\ell \times (\ell-k)} \mid \mathbf{A}^\top \mathbf{A}^\perp = \mathbf{0} \text{ and } \mathbf{A}^\perp \text{ has rank } \ell - k\}.$$

For vectors $\mathbf{v} \in \mathbb{Z}_p^{k+n}$ ($n \in \mathbb{N}$), by $\overline{\mathbf{v}} \in \mathbb{Z}_p^k$ we denote the vector consisting of the upper k entries of $\mathbf{v}$ and accordingly by $\underline{\mathbf{v}} \in \mathbb{Z}_p^n$ we denote the vector consisting of the remaining n entries of $\mathbf{v}$.

Similarly, for a matrix $\mathbf{A} \in \mathbb{Z}_p^{2k \times k}$, by $\overline{\mathbf{A}} \in \mathbb{Z}_p^{k \times k}$ we denote the upper square matrix and by $\underline{\mathbf{A}} \in \mathbb{Z}_p^{k \times k}$ the lower one.

2.2 Pairing Groups and Matrix Diffie-Hellman Assumptions

Let GGen be a probabilistic polynomial time (PPT) algorithm that on input 1^λ returns a description $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, G_T, p, P_1, P_2, e)$ of asymmetric pairing groups where $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ are cyclic group of order p for a 2λ-bit prime p, P_1 and P_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable (non-degenerate) bilinear map. Define $P_T := e(P_1, P_2)$, which is a generator of $\mathbb{G}_T$. We use implicit representation of group elements. For $i \in \{1, 2, T\}$ and $a \in \mathbb{Z}_p$, we define $[a]_i = aP_i \in \mathbb{G}_i$ as the implicit representation of a in $\mathbb{G}_i$. Given $[a]_1$, $[a]_2$, one can efficiently compute $[ab]_T$ using the pairing e.

For two matrices $\mathbf{A}$, $\mathbf{B}$ with matching dimensions, we define $e([\mathbf{A}]_1, [\mathbf{B}]_2) := [\mathbf{AB}]_T \in \mathbb{G}_T$.

We recall the definitions of the Matrix Decision Diffie-Hellman (MDDH) assumption from [24].

Definition 1 (Matrix distribution). *Let $k, \ell \in \mathbb{N}$, with $\ell > k$ and p be a 2λ-bit prime. We call a PPT algorithm $\mathcal{D}_{\ell,k}$ a* matrix distribution *if it outputs matrices in $\mathbb{Z}_p^{\ell \times k}$ of full rank k.*

Note that instantiating $\mathcal{D}_{2,1}$ with a PPT algorithm outputting matrices $\begin{pmatrix} 1 \\ a \end{pmatrix}$ for $a \leftarrow_R \mathbb{Z}_p$, $\mathcal{D}_{2,1}$-MDDH relative to $\mathbb{G}_1$ corresponds to the DDH assumption in $\mathbb{G}_1$. Thus, for $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, G_T, p, P_1, P_2, e)$, assuming $\mathcal{D}_{2,1}$-MDDH relative to $\mathbb{G}_1$ and relative to $\mathbb{G}_2$, corresponds to the SXDH assumption.

In the following we only consider matrix distributions $\mathcal{D}_{\ell,k}$, where for all $\mathbf{A} \leftarrow_R \mathcal{D}_{\ell,k}$ the first k rows of $\mathbf{A}$ form an invertible matrix. We also require that in case $\ell = 2k$ for any two matrices $\mathbf{A}_0, \mathbf{A}_1 \leftarrow_R \mathcal{D}_{2k,k}$ the matrix $(\mathbf{A}_0 \mid \mathbf{A}_1)$ has full rank with overwhelming probability. In the following we will denote this probability by $1 - \Delta_{\mathcal{D}_{2k,k}}$. Note that if $(\mathbf{A}_0 \mid \mathbf{A}_1)$ has full rank, then for any $\mathbf{A}_0^\perp \in \mathsf{orth}(\mathbf{A}_0)$, $\mathbf{A}_1^\perp \in \mathsf{orth}(\mathbf{A}_1)$ the matrix $(\mathbf{A}_0^\perp \mid \mathbf{A}_1^\perp) \in \mathbb{Z}_p^{2k \times 2k}$ has full rank as well, as otherwise there would exists a non-zero vector $\mathbf{v} \in \mathbb{Z}_p^{2k} \backslash \{\mathbf{0}\}$ with $(\mathbf{A}_0 \mid \mathbf{A}_1)^\top \mathbf{v} = \mathbf{0}$. Further, by similar reasoning $(\mathbf{A}_0^\perp)^\top \mathbf{A}_1 \in \mathbb{Z}_p^{k \times k}$ has full rank.

The $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman problem in $\mathbb{G}_i$, for $i \in \{1, 2, T\}$, is to distinguish the between tuples of the form $([\mathbf{A}]_i, [\mathbf{Aw}]_i)$ and $([\mathbf{A}]_i, [\mathbf{u}]_i)$, for a randomly chosen $\mathbf{A} \leftarrow_R \mathcal{D}_{\ell,k}$, $\mathbf{w} \leftarrow_R \mathbb{Z}_p^k$ and $\mathbf{u} \leftarrow_R \mathbb{Z}_p^\ell$.

Definition 2 ($\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman $\mathcal{D}_{\ell,k}$-MDDH). *Let $\mathcal{D}_{\ell,k}$ be a matrix distribution. We say that the* $\mathcal{D}_{\ell,k}$-Matrix Diffie-Hellman *($\mathcal{D}_{\ell,k}$-MDDH) assumption holds relative to a prime order group $\mathbb{G}_i$ for $i \in \{1, 2, T\}$, if for all PPT adversaries $\mathcal{A}$,*

$$\begin{aligned}\mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_i,\mathcal{D}_{\ell,k},\mathcal{A}}(\lambda) := &|\Pr[\mathcal{A}(\mathcal{PG}, [\mathbf{A}]_i, [\mathbf{Aw}]_i) = 1] \\ &- \Pr[\mathcal{A}(\mathcal{PG}, [\mathbf{A}]_i, [\mathbf{u}]_i) = 1]| \leq \mathsf{negl}(\lambda),\end{aligned}$$

where the probabilities are taken over $\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, P_1, P_2) \leftarrow \mathsf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow_R \mathcal{D}_{\ell,k}, \mathbf{w} \leftarrow_R \mathbb{Z}_p^k, \mathbf{u} \leftarrow_R \mathbb{Z}_p^\ell$.

For $Q \in \mathbb{N}$, $\mathbf{W} \leftarrow_R \mathbb{Z}_p^{k \times Q}$ and $\mathbf{U} \leftarrow_R \mathbb{Z}_p^{\ell \times Q}$, we consider the Q-fold $\mathcal{D}_{\ell,k}$-MDDH assumption, which states that distinguishing tuples of the form $([\mathbf{A}]_i, [\mathbf{AW}]_i)$ from $([\mathbf{A}]_i, [\mathbf{U}]_i)$ is hard. That is, a challenge for the Q-fold $\mathcal{D}_{\ell,k}$-MDDH assumption consists of Q independent challenges of the $\mathcal{D}_{\ell,k}$-MDDH assumption (with the same $\mathbf{A}$ but different randomness $\mathbf{w}$). In [24] it is shown that the two problems are equivalent, where the reduction loses at most a factor $\ell - k$.

Lemma 1 (Random self-reducibility of $\mathcal{D}_{\ell,k}$-MDDH, [24]). *Let $\ell, k, Q \in \mathbb{N}$ with $\ell > k$ and $Q > \ell - k$ and $i \in \{1, 2, T\}$. For any PPT adversary $\mathcal{A}$,*

there exists an adversary $\mathcal{B}$ *such that* $T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ *with* $\mathsf{poly}(\lambda)$ *independent of* $T(\mathcal{A})$*, and*

$$\mathrm{Adv}^{Q\text{-mddh}}_{\mathcal{PG},\mathbb{G}_i,\mathcal{D}_{\ell,k},\mathcal{A}}(\lambda) \leq (\ell - k) \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_i,\mathcal{D}_{\ell,k},\mathcal{B}}(\lambda) + \tfrac{1}{p-1}.$$

Here

$$\begin{aligned}\mathrm{Adv}^{Q\text{-mddh}}_{\mathcal{PG},\mathbb{G}_i,\mathcal{D}_{\ell,k},\mathcal{A}}(\lambda) := |&\Pr[\mathcal{A}(\mathcal{PG}, [\mathbf{A}]_i, [\mathbf{AW}]_i) = 1] \\ &- \Pr[\mathcal{A}(\mathcal{PG}, [\mathbf{A}]_i, [\mathbf{U}]_i) = 1]|\,,\end{aligned}$$

where the probability is over $\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, P_1, P_2) \leftarrow \mathsf{GGen}(1^\lambda)$*,* $\mathbf{A} \leftarrow_R \mathcal{D}_{\ell,k}$*,* $\mathbf{W} \leftarrow_R \mathbb{Z}_p^{k\times Q}$ *and* $\mathbf{U} \leftarrow_R \mathbb{Z}_p^{\ell\times Q}$*.*

For $k \in \mathbb{N}$ we define $\mathcal{D}_k := \mathcal{D}_{k+1,k}$.

The Kernel-Diffie-Hellman assumption $\mathcal{D}_k$-KMDH [45] is a natural *computational analogue* of the $\mathcal{D}_k$-MDDH Assumption.

Definition 3 ($\mathcal{D}_k$-Kernel Diffie-Hellman assumption $\mathcal{D}_k$-KMDH). *Let* $\mathcal{D}_k$ *be a matrix distribution. We say that the* $\mathcal{D}_k$*-Kernel Diffie-Hellman (*$\mathcal{D}_k$*-KMDH) assumption holds relative to a prime order group* $\mathbb{G}_i$ *for* $i \in \{1, 2\}$ *if for all PPT adversaries* $\mathcal{A}$*,*

$$\begin{aligned}\mathrm{Adv}^{\mathrm{kmdh}}_{\mathcal{PG},\mathbb{G}_i,\mathcal{D}_{\ell,k},\mathcal{A}}(\lambda) &:= \Pr[\mathbf{c}^\top \mathbf{A} = \mathbf{0} \wedge \mathbf{c} \neq \mathbf{0} \mid [\mathbf{c}]_{3-i} \leftarrow_R \mathcal{A}(\mathcal{PG}, [\mathbf{A}]_i)] \\ &\leq \mathsf{negl}(\lambda),\end{aligned}$$

where the probabilities are taken over $\mathcal{PG} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, P_1, P_2) \leftarrow \mathsf{GGen}(1^\lambda)$*, and* $\mathbf{A} \leftarrow_R \mathcal{D}_k$*.*

Note that we can use a non-zero vector in the kernel of $\mathbf{A}$ to test membership in the column space of $\mathbf{A}$. This means that the $\mathcal{D}_k$-KMDH assumption is a relaxation of the $\mathcal{D}_k$-MDDH assumption, as captured in the following lemma from [45].

Lemma 2 ([45]). *For any matrix distribution* $\mathcal{D}_k$*,* $\mathcal{D}_k$*-MDDH* $\Rightarrow$ $\mathcal{D}_k$*-KMDH.*

2.3 Signature Schems and Message Authentication Codes

Definition 4 (MAC). *A* message authentication code (MAC) *is a tuple of PPT algorithms* $\mathsf{MAC} := (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ *such that:*

$\mathsf{Gen}(1^\lambda)$*: on input of the security parameter, generates public parameters pp and a secret key sk.*

$\mathsf{Tag}(pp, sk, m)$*: on input of public parameters pp, the secret key sk and a message* $m \in \mathcal{M}$*, returns a tag* tag*.*

$\mathsf{Ver}(pp, sk, m, \mathsf{tag})$*: verifies the tag* tag *for the message m, outputting a bit* $b = 1$ *if* tag *is valid respective to m, and* 0 *otherwise.*

We say MAC *is* **perfectly correct**, *if for all* $\lambda \in \mathbb{N}$, *all* $m \in \mathcal{M}$ *and all* $(pp, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ *we have*

$$\mathsf{Ver}(pp, sk, m, \mathsf{Tag}(pp, sk, m)) = 1.$$

Definition 5 (UF-CMA security). *Let* $\mathsf{MAC} := (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ *be a MAC. For any adversary* $\mathcal{A}$, *we define the following experiment:*

$\underline{\mathrm{Exp}^{\text{uf-cma}}_{\mathsf{MAC},\mathcal{A}}(\lambda)}$:	$\underline{\textsc{TagO}(m)}$:
$(pp, sk) \leftarrow \mathsf{Gen}(1^\lambda)$	$\mathcal{Q}_{\mathsf{tag}} := \mathcal{Q}_{\mathsf{tag}} \cup \{m\}$
$\mathcal{Q}_{\mathsf{tag}} := \emptyset$	$\mathsf{tag} \leftarrow \mathsf{Tag}(pp, sk, m)$
$(m^\star, \mathsf{tag}^\star) \leftarrow \mathcal{A}^{\textsc{TagO}(\cdot)}(pp)$	***return*** tag
if $m^\star \notin \mathcal{Q}_{\mathsf{tag}}$ ***and*** $\textsc{VerO}(m^\star, \mathsf{tag}^\star) = 1$	
return 1	$\underline{\textsc{VerO}(m, \mathsf{tag})}$:
else return 0	$b \leftarrow \mathsf{Ver}(pp, sk, m, \mathsf{tag})$
	return b

The adversary is restricted to one call to VerO. *We say that a MAC scheme* MAC *is* UF-CMA secure, *if for all PPT adversaries* $\mathcal{A}$,

$$\mathrm{Adv}^{\text{uf-cma}}_{\mathsf{MAC},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{\text{uf-cma}}_{\mathsf{MAC},\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda).$$

Note that in our notion of UF-CMA security, the adversary gets only one forgery attempt. This is due to the fact that we employ the MAC primarily as a building block for our signature. Our notion suffices for this purpose, as an adversary can check the validity of a signature itself.

Definition 6 (Signature). *A* signature scheme *is a tuple of PPT algorithms* $\mathsf{SIG} := (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *such that:*

$\mathsf{Gen}(1^\lambda)$*: on input of the security parameter, generates a pair* (pk, sk) *of keys.*
$\mathsf{Sign}(pk, sk, m)$*: on input of the public key* pk*, the secret key* sk *and a message* $m \in \mathcal{M}$*, returns a signature* σ*.*
$\mathsf{Ver}(pk, m, \sigma)$*: verifies the signature* σ *for the message* m*, outputting a bit* $b = 1$ *if* σ *is valid respective to* m*, and* 0 *otherwise.*

We say that SIG *is* **perfectly correct**, *if for all* $\lambda \in \mathbb{N}$, *all* $m \in \mathcal{M}$ *and all* $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$,

$$\mathsf{Ver}(pk, m, \mathsf{Sign}(pk, sk, m)) = 1.$$

In bilinear pairing groups, we say a signature scheme SIG *is* structure-preserving *if its public keys, signing messages, signatures contain only group elements and verification proceeds via only a set of pairing product equations.*

Definition 7 (UF-CMA security). *For a signature scheme* $\mathsf{SIG} := (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *and any adversary* $\mathcal{A}$, *we define the following experiment:*

$\underline{\mathrm{Exp}^{\text{uf-cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda)}$:	$\underline{\textsc{Signo}(m)}$:
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$	$\mathcal{Q}_{\mathsf{sign}} := \mathcal{Q}_{\mathsf{sign}} \cup \{m\}$
$\mathcal{Q}_{\mathsf{sign}} := \emptyset$	$\sigma \leftarrow \mathsf{Sign}(pk, sk, m)$
$(m^\star, \sigma^\star) \leftarrow \mathcal{A}^{\textsc{Signo}(\cdot)}(pk)$	***return*** σ
if $m^\star \notin \mathcal{Q}_{\mathsf{sign}}$ ***and*** $\mathsf{Ver}(pk, m^\star, \sigma^\star) = 1$	
return 1	
else return 0	

We say that a signature scheme SIG *is* UF-CMA*, if for all PPT adversaries* $\mathcal{A}$*,*

$$\mathrm{Adv}^{\text{uf-cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{\text{uf-cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda).$$

2.4 Non-interactive Zero-Knowledge Proof (NIZK)

The notion of a non-interactive zero-knowledge proof was introduced in [15]. In the following we present the definition from [32]. Non-interactive zero-knowledge proofs will serve as a crucial building block for our constructions.

Definition 8 (Non-interactive zero-knowledge proof [32]). *We consider a family of languages* $\mathcal{L} = \{\mathcal{L}_{pars}\}$ *with efficiently computable witness relation* $\mathcal{R}_{\mathcal{L}}$*. A non-interactive zero-knowledge proof for* $\mathcal{L}$ *is a tuple of PPT algorithms* $\mathsf{PS} := (\mathsf{PGen}, \mathsf{PTGen}, \mathsf{PPrv}, \mathsf{PVer}, \mathsf{PSim})$ *such that:*

$\mathsf{PGen}(1^\lambda, pars)$ *generates a common reference string crs.*
$\mathsf{PTGen}(1^\lambda, pars)$ *generates a common reference string crs and additionally a trapdoor td.*
$\mathsf{PPrv}(crs, x, w)$ *given a word* $x \in \mathcal{L}$ *and a witness w with* $\mathcal{R}_{\mathcal{L}}(x, w) = 1$*, outputs a proof* $\Pi \in \mathcal{P}$*.*
$\mathsf{PVer}(crs, x, \Pi)$ *on input crs,* $x \in \mathcal{X}$ *and* Π *outputs a verdict* $b \in \{0, 1\}$*.*
$\mathsf{PSim}(crs, td, x)$ *given a crs with corresponding trapdoor td and a word* $x \in \mathcal{X}$*, outputs a proof* Π*.*

Further we require the following properties to hold.

Completeness: *For all possible public parameters pars, for all words* $x \in \mathcal{L}$*, and all witnesses w such that* $\mathcal{R}_{\mathcal{L}}(x, w) = 1$*, we have*

$$\Pr[\mathsf{PVer}(crs, x, \Pi) = 1] = 1,$$

where the probability is taken over $(crs, psk) \leftarrow \mathsf{PGen}\ (1^\lambda, pars)$ *and* $\Pi \leftarrow \mathsf{PPrv}(crs, x, w)$*.*

Composable zero-knowledge*: *For all PPT adversaries* $\mathcal{A}$ *we have that*

$$\begin{aligned}\mathrm{Adv}^{\text{keygen}}_{\mathsf{PS},\mathcal{A}}(\lambda) := \big|&\Pr[\mathcal{A}(1^\lambda, crs) = 1 \mid crs \leftarrow \mathsf{PGen}(1^\lambda, pars)] \\ &- \Pr[\mathcal{A}(1^\lambda, crs) = 1 \mid (crs, td) \leftarrow \mathsf{PTGen}(1^\lambda, pars)]\big|\end{aligned}$$

is negligible in λ*.*

Further for all public parameters pars, all pairs (crs, td) *in the range of* $\mathsf{PTGen}(1^\lambda)$*, all words* $x \in \mathcal{L}$*, and all witnesses* w *with* $\mathcal{R}_\mathcal{L}(x, w) = 1$*, we have that the outputs of*

$$\mathsf{PPrv}(crs, x, w) \text{ and } \mathsf{PSim}(crs, td, x)$$

are statistically indistinguishable.

Perfect soundness: *For all crs in the range of* $\mathsf{PGen}(1^\lambda, pars)$*, for all words* $x \notin \mathcal{L}$ *and all proofs* Π *it holds* $\mathsf{PVer}(crs, x, \Pi) = 0$*.*

Remark. We will employ a weaker notion of composable zero-knowledge in the following. Namely:

Composable zero-knowledge: For a PPT adversary $\mathcal{A}$, we define

$$\mathrm{Adv}^{\mathsf{zk}}_{\mathsf{PS},\mathcal{A}}(\lambda) := \left| \Pr\left[b' = b \;\middle|\; \begin{array}{l} crs_0 \leftarrow_R \mathsf{PGen}(1^\lambda, pars); \\ (crs_1, td) \leftarrow_R \mathsf{PTGen}(1^\lambda, pars); \\ b \leftarrow_R \{0,1\}; \\ b' \leftarrow_R \mathcal{A}^{\textsc{Prove}(\cdot,\cdot)}(1^\lambda, crs_b) \end{array} \right] - \tfrac{1}{2} \right|.$$

$\underline{\mathsf{PGen}(1^\lambda, pars)}$:
$\mathbf{D} \leftarrow_R \mathcal{D}_k$, $\mathbf{z} \leftarrow_R \mathbb{Z}_p^{k+1} \backslash \mathrm{span}(\mathbf{D})$
//recall $\mathcal{D}_k := \mathcal{D}_{k+1,k}$
$crs := (pars, [\mathbf{D}]_2, [\mathbf{z}]_2)$
return crs

$\underline{\mathsf{PPrv}(crs, [\mathbf{x}]_1, \mathbf{r})}$:
let $b \in \{0,1\}$ **s.t.** $[\mathbf{x}]_1 = [\mathbf{A}_b]_1 \cdot \mathbf{r}$
$\mathbf{v} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{z}_{1-b}]_2 := [\mathbf{D}]_2 \cdot \mathbf{v}$
// $([\mathbf{D}]_2, [\mathbf{z}_{1-b}]_2)$ trapdoor crs
$[\mathbf{z}_b]_2 := [\mathbf{z}]_2 - [\mathbf{z}_{1-b}]_2$
// crs guaranteeing soundness
$\mathbf{S}_0, \mathbf{S}_1 \leftarrow_R \mathbb{Z}_p^{k \times k}$
$[\mathbf{C}_b]_2 := \mathbf{S}_b \cdot [\mathbf{D}]_2^\top + \mathbf{r} \cdot [\mathbf{z}_b]_2^\top$
//commitment to $\mathbf{r}$ with rand. $\mathbf{S}_b$
$[\Pi_b]_1 := [\mathbf{A}_b]_1 \cdot \mathbf{S}_b$
//proof for $\mathbf{x} = \mathbf{A}_b\mathbf{r}$
$[\mathbf{C}_{1-b}]_2 := \mathbf{S}_{1-b} \cdot [\mathbf{D}]_2^\top$
//commitment to $\mathbf{0}$ with rand. $\mathbf{S}_{1-b}$
$[\Pi_{1-b}]_1 := [\mathbf{A}_{1-b}]_1 \cdot \mathbf{S}_{1-b} - [\mathbf{x}]_1 \cdot \mathbf{v}^\top$
//trapdoor proof for $\mathbf{x} = \mathbf{A}_{1-b}\mathbf{r}$
return $([\mathbf{z}_0]_2, ([\mathbf{C}_i]_2, [\Pi_i]_1)_{i \in \{0,1\}})$

$\underline{\mathsf{PTGen}(1^\lambda, pars)}$:
$\mathbf{D} \leftarrow_R \mathcal{D}_k$, $\mathbf{u} \leftarrow_R \mathbb{Z}_p^k$
$\mathbf{z} := \mathbf{D} \cdot \mathbf{u}$
$crs := (pars, [\mathbf{D}]_2, [\mathbf{z}]_2)$, $td := \mathbf{u}$
return (crs, td)

$\underline{\mathsf{PVer}(crs, [\mathbf{x}]_1, ([\mathbf{z}_0]_2, ([\mathbf{C}_i]_2, [\Pi_i]_1)_{i \in \{0,1\}}))}$:
$[\mathbf{z}_1]_2 := [\mathbf{z}]_2 - [\mathbf{z}_0]_2$
if for all $i \in \{0,1\}$ **it holds**
$e([\mathbf{A}_i]_1, [\mathbf{C}_i]_2)$
$= e([\Pi_i]_1, [\mathbf{D}]_2^\top) + e([\mathbf{x}]_1, [\mathbf{z}_i]_2^\top)$
//check $\mathbf{C}_i \cdot \mathbf{A}_i \stackrel{?}{=} \Pi_i \cdot \mathbf{D}^\top + \mathbf{x} \cdot \mathbf{z}_i^\top$
return 1
else return 0

$\underline{\mathsf{PSim}(crs, td, [\mathbf{x}]_1)}$:
parse $td =: \mathbf{u}$
$\mathbf{v} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{z}_0]_2 := [\mathbf{D}]_2 \cdot \mathbf{v}$
$[\mathbf{z}_1]_2 := [\mathbf{z}]_2 - [\mathbf{z}_0]_2$
$\mathbf{S}_0, \mathbf{S}_1 \leftarrow_R \mathbb{Z}_p^{k \times k}$
$[\mathbf{C}_0]_2 := \mathbf{S}_0 \cdot [\mathbf{D}]_2^\top$
$[\Pi_0]_1 := [\mathbf{A}_0]_1 \cdot \mathbf{S}_0 - [\mathbf{x}]_1 \cdot \mathbf{v}^\top$
$[\mathbf{C}_1]_2 := \mathbf{S}_1 \cdot [\mathbf{D}]_2^\top$
$[\Pi_1]_1 := [\mathbf{A}_1]_1 \cdot \mathbf{S}_1 - [\mathbf{x}]_1 \cdot (\mathbf{u} - \mathbf{v})^\top$
return $([\mathbf{z}_0]_2, ([\mathbf{C}_i]_2, [\Pi_i]_1)_{i \in \{0,1\}})$

Fig. 1. NIZK argument for $\mathcal{L}^\vee_{\mathbf{A}_0,\mathbf{A}_1}$ [31,46].

Here $\textsc{Prove}(x, w)$ returns $\perp$ if $\mathcal{R}_{\mathcal{L}}(x, w) = 0$ or Π_b if $\mathcal{R}_{\mathcal{L}}(x, w) = 1$, where $\Pi_0 \leftarrow_R \mathsf{PPrv}(crs_0, x, w)$ and $\Pi_1 \leftarrow_R \mathsf{PSim}(crs_1, td, x)$. We say that PS satisfies composable zero-knowledge if $\mathrm{Adv}^{\mathrm{zk}}_{\mathsf{PS},\mathcal{A}}(\lambda)$ is negligible in λ for all PPT $\mathcal{A}$.

Note that the original definition of composable zero-knowledge tightly implies our definition of composable zero-knowledge. We choose to work with the latter in order to simplify the presentation of our proofs. Note that for working with this definition in the tightness setting, it is crucial that $\mathrm{Adv}^{\mathrm{zk}}_{\mathsf{PS},\mathcal{A}}(\lambda)$ is independent of the number of queries to the oracle $\textsc{Prove}$.

2.5 NIZK for Our OR-language

In this section we recall an instantiation of a NIZK for an OR-language implicitly given in [31,46]. This NIZK will be a crucial part of all our constructions, allowing to employ the randomization techniques from [6,26,33] to obtain a tight security reduction.

Public Parameters. Let $\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$. Let $k \in \mathbb{N}$. Let $\mathbf{A}_0, \mathbf{A}_1 \leftarrow_R \mathcal{D}_{2k,k}$. We define the public parameters to comprise

$$pars := (\mathcal{PG}, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1).$$

We consider $k \in \mathbb{N}$ to be chosen ahead of time, fixed and implicitly known to all algorithms.

OR-Proof ([31,46]). In Fig. 1 we present a non-interactive zero-knowledge proof for the OR-language

$$\mathcal{L}^{\vee}_{\mathbf{A}_0,\mathbf{A}_1} := \{[\mathbf{x}]_1 \in \mathbb{Z}_p^{2k} \mid \exists \mathbf{r} \in \mathbb{Z}_p^k \colon [\mathbf{x}]_1 = [\mathbf{A}_0]_1 \cdot \mathbf{r} \vee [\mathbf{x}]_1 = [\mathbf{A}_1]_1 \cdot \mathbf{r}\}.$$

Note that this OR-proof is implicitly given in [31,46]. We recall the proof in the full version.

Lemma 3. *If the $\mathcal{D}_k$-MDDH assumption holds in the group $\mathbb{G}_2$, then the proof system $\mathsf{PS} := (\mathsf{PGen}, \mathsf{PTGen}, \mathsf{PPrv}, \mathsf{PVer}, \mathsf{PSim})$ as defined in Fig. 1 is a non-interactive zero-knowledge proof for $\mathcal{L}^{\vee}_{\mathbf{A}_0,\mathbf{A}_1}$. More precisely, for every adversary $\mathcal{A}$ attacking the composable zero-knowledge property of PS, we obtain an adversary $\mathcal{B}$ with $T(\mathcal{B}) \approx T(\mathcal{A}) + Q_{\mathsf{prove}} \cdot \mathsf{poly}(\lambda)$ and*

$$\mathrm{Adv}^{\mathrm{zk}}_{\mathsf{PS},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_2,\mathcal{D}_k,\mathcal{B}}(\lambda).$$

3 Tightly Secure Message Authentication Code Scheme

Let $k \in \mathbb{N}$ and let $\mathsf{PS} := (\mathsf{PGen}, \mathsf{PTGen}, \mathsf{PPrv}, \mathsf{PSim})$ a non-interactive zero-knowledge proof for $\mathcal{L}^{\vee}_{\mathbf{A}_0,\mathbf{A}_1}$ as defined in Sect. 2.5. In Fig. 2 we provide a MAC $\mathsf{MAC} := (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ whose security can be tightly reduced to $\mathcal{D}_{2k,k}$-MDDH and the security of the underlying proof system PS.

$\underline{\mathsf{Gen}(1^\lambda)}$:
$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$
$\mathbf{A}_0, \mathbf{A}_1 \leftarrow \mathcal{D}_{2k,k}$
$pars := (\mathcal{PG}, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1)$
$crs \leftarrow \mathsf{PGen}(1^\lambda, pars)$
$\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$
$pp := (\mathcal{PG}, [\mathbf{A}_0]_1, crs)$
$sk := (\mathbf{k}_0, \mathbf{k}_1)$
return (pp, sk)

$\underline{\mathsf{Tag}(pp, sk, \mu \in \mathbb{Z}_p)}$:
parse $pp =: (\mathcal{PG}, [\mathbf{A}_0]_1, crs)$
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{t}]_1 := [\mathbf{A}_0]_1 \mathbf{r}$
$\Pi \leftarrow \mathsf{PPrv}(crs, [\mathbf{t}]_1, \mathbf{r})$
$[u]_1 := (\mathbf{k}_0 + \mu \mathbf{k}_1)^\top [\mathbf{t}]_1$
$\mathsf{tag} := ([\mathbf{t}]_1, \Pi, [u]_1)$
return tag

$\underline{\mathsf{Ver}(pp, sk, \mu \in \mathbb{Z}_p, \mathsf{tag})}$:
parse $\mathsf{tag} =: ([\mathbf{t}]_1, \Pi, [u]_1)$
$b \leftarrow \mathsf{PVer}(crs, [\mathbf{t}]_1, \Pi)$
if $b = 1$ **and** $[u]_1 \neq [0]_1$
and $[u]_1 = (\mathbf{k}_0 + \mu \mathbf{k}_1)^\top [\mathbf{t}]_1$
return 1
else return 0

Fig. 2. Tightly secure MAC $\mathsf{MAC} := (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ from the $\mathcal{D}_{2k,k}$-MDDH assumption.

Instead of directly proving UF-CMA security of our MAC, we will first provide our so-called core lemma, which captures the essential randomization technique from [6,26,33]. We can employ this lemma to prove the security of our MAC and (structure-preserving) signature schemes. Essentially, the core lemma shows that the term $[\mathbf{k}_0^\top \mathbf{t}]_1$ is pseudorandom. We give the corresponding formal experiment in Fig. 3.

$\underline{\mathrm{Exp}^{\mathrm{core}}_{\beta,\mathcal{A}}(\lambda), \text{ for } \beta \in \{0,1\}}$:
$\mathsf{ctr} := 0$
$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$
$\mathbf{A}_0, \mathbf{A}_1 \leftarrow_R \mathcal{D}_{2k,k}$
$pars := (\mathcal{PG}, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1)$
$crs \leftarrow \mathsf{PGen}(1^\lambda, pars)$
$\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$
$pp := (\mathcal{PG}, [\mathbf{A}_0]_1, crs)$
$\mathsf{tag} \leftarrow \mathcal{A}^{\mathrm{TagO}()}(pp)$
return $\mathrm{VerO}(\mathsf{tag})$

$\underline{\mathrm{TagO}()}$:
$\mathsf{ctr} := \mathsf{ctr} + 1$
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{t}]_1 := [\mathbf{A}_0]_1 \mathbf{r}$
$\Pi \leftarrow \mathsf{PPrv}(crs, [\mathbf{t}]_1, \mathbf{r})$
$[u']_1 := (\mathbf{k}_0 \boxed{+\beta \cdot \mathbf{F}(\mathsf{ctr})})^\top [\mathbf{t}]_1$
$\mathsf{tag} := ([\mathbf{t}]_1, \Pi, [u']_1)$
return tag

$\underline{\mathrm{VerO}(\mathsf{tag})}$:
parse $\mathsf{tag} = ([\mathbf{t}]_1, \Pi, [u']_1)$
$b \leftarrow \mathsf{PVer}(crs, [\mathbf{t}]_1, \Pi)$
if $b = 1$ **and** $\exists \mathsf{ctr}' \leq \mathsf{ctr}$:
$[u']_1 = (\mathbf{k}_0 \boxed{+\beta \cdot \mathbf{F}(\mathsf{ctr}')})^\top [\mathbf{t}]_1$
return 1
else return 0

Fig. 3. Experiment for the core lemma. Here, $\mathbf{F} : \mathbb{Z}_p \to \mathbb{Z}_p^{2k}$ is a random function computed on the fly. We highlight the difference between $\mathrm{Exp}^{\mathrm{core}}_{0,\mathcal{A}}$ and $\mathrm{Exp}^{\mathrm{core}}_{1,\mathcal{A}}$ in gray.

Lemma 4 (Core lemma). *If the $\mathcal{D}_{2k,k}$-MDDH assumption holds in $\mathbb{G}_1$ and the tuple of algorithms $\mathsf{PS} := (\mathsf{PGen}, \mathsf{PTGen}, \mathsf{PPrv}, \mathsf{PVer})$ is a non-interactive zero-knowledge proof system for $\mathcal{L}^{\vee}_{\mathbf{A}_0,\mathbf{A}_1}$, then going from experiment $\mathrm{Exp}^{\mathrm{core}}_{0,\mathcal{A}}(\lambda)$ to $\mathrm{Exp}^{\mathrm{core}}_{1,\mathcal{A}}(\lambda)$ can (up to negligible terms) only increase the winning chances of an adversary. More precisely, for every adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}$, $\mathcal{B}'$ with running time $T(\mathcal{B}) \approx T(\mathcal{B}') \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ such that*

$$\mathrm{Adv}^{\mathrm{core}}_{0,\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathrm{core}}_{1,\mathcal{A}}(\lambda) + \Delta^{\mathrm{core}}_{\mathcal{A}}(\lambda),$$

where

$$\begin{aligned}\Delta^{\mathrm{core}}_{\mathcal{A}}(\lambda) :=& (4k\lceil \log Q \rceil + 2) \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_1,\mathcal{D}_{2k,k},\mathcal{B}}(\lambda) \\ &+ (2\lceil \log Q \rceil + 2) \cdot \mathrm{Adv}^{\mathrm{ZK}}_{\mathsf{PS},\mathcal{B}'}(\lambda) \\ &+ \lceil \log Q \rceil \cdot \Delta_{\mathcal{D}_{2k,k}} + \tfrac{4\lceil \log Q \rceil + 2}{p-1} + \tfrac{\lceil \log Q \rceil \cdot Q}{p}.\end{aligned}$$

Recall that by definition of the distribution $\mathcal{D}_{2k,k}$ (Sect. 2.2), the term $\Delta_{\mathcal{D}_{2k,k}}$ is statistically small.

Proof Outline. Since the proof of Lemma 4 is rather complex, we first outline our strategy. Intuitively, our goal is to randomize the term u' used by oracles TagO and VerO (i.e., to change this term from $\mathbf{k}_0^\top \mathbf{t}$ to $(\mathbf{k}_0 + \mathbf{F}(\mathsf{ctr}))^\top \mathbf{t}$ for a truly random function $\mathbf{F}$). In this, it will also be helpful to change the distribution of $\mathbf{t} \in \mathbb{Z}_p^{2k}$ in tags handed out by TagO as needed. (Intuitively, changing $\mathbf{t}$ can be justified with the $\mathcal{D}_{2k,k}$-MDDH assumption, but we can only rely on the soundness of PS if $\mathbf{t} \in \mathrm{span}(\mathbf{A}_0) \cup \mathrm{span}(\mathbf{A}_1)$. In other words, we may assume that $\mathbf{t} \in \mathrm{span}(\mathbf{A}_0) \cup \mathrm{span}(\mathbf{A}_1)$ for any of $\mathcal{A}$'s VerO queries, but only if the same holds for all $\mathbf{t}$ chosen by TagO.)

We will change u' using a hybrid argument, where in the i-th hybrid we set $u' = (\mathbf{k}_0^\top + \mathbf{F}_i(\mathsf{ctr}_{|i}))^\top \mathbf{t}$ for a random function $\mathbf{F}_i$ on i-bit prefixes, and the i-bit prefix $\mathsf{ctr}_{|i}$ of ctr. (That is, we introduce more and more dependencies on the bits of ctr.) To move from hybrid i to hybrid $i+1$, we proceed again along a series of hybrids (outsourced into the full version), and perform the following modifications:

Partitioning. First, we choose $\mathbf{t} \in \mathrm{span}(\mathbf{A}_{\mathsf{ctr}_{i+1}})$ in VerO, where ctr_{i+1} is the $(i+1)$-th bit of ctr. As noted above, this change can be justified with the $\mathcal{D}_{2k,k}$-MDDH assumption, and we may still assume $\mathbf{t} \in \mathrm{span}(\mathbf{A}_0) \cup \mathrm{span}(\mathbf{A}_1)$ in every TagO query from $\mathcal{A}$.

Decoupling. At this point, the values u' computed in TagO and VerO are either of the form $u' = (\mathbf{k}_0^\top + \mathbf{F}_i(\mathsf{ctr}_{|i}))^\top \mathbf{A}_0 \mathbf{r}$ or $u' = (\mathbf{k}_0^\top + \mathbf{F}_i(\mathsf{ctr}_{|i}))^\top \mathbf{A}_1 \mathbf{r}$ (depending on $\mathbf{t}$). Since $\mathbf{F}_i : \{0,1\}^i \to \mathbb{Z}_p^{2k}$ is truly random, and the matrix $\mathbf{A}_0 || \mathbf{A}_1 \in \mathbb{Z}_p^{2k \times 2k}$ has linearly independent columns (with overwhelming probability), the two possible subterms $\mathbf{F}_i(\mathsf{ctr}_{|i})^\top \mathbf{A}_0$ and $\mathbf{F}_i(\mathsf{ctr}_{|i})^\top \mathbf{A}_1$ are independent. Thus, switching to $u' = (\mathbf{k}_0^\top + \mathbf{F}_{i+1}(\mathsf{ctr}_{|i+1}))^\top \mathbf{t}$ does not change $\mathcal{A}$'s view at all.

After these modifications (and resetting $\mathbf{t}$), we have arrived at the $(i+1)$-th hybrid, which completes the proof. However, this outline neglects a number of details, including a proper reasoning of PS proofs, and a careful discussion of the decoupling step. In particular, an additional complication arises in this step from the fact that an adversary may choose $\mathbf{t} \in \mathrm{span}(A_b)$ for an arbitrary bit b not related to any specific ctr. This difficulty is the reason for the somewhat surprising "$\exists \mathsf{ctr}' \leq \mathsf{ctr}$" clause in VerO.

Proof (of Lemma 4). We proceed via a series of hybrid games $\mathsf{G}_0, \ldots, \mathsf{G}_{3.\lceil \log Q \rceil}$, described in Fig. 4, and we denote by ε_i the advantage of $\mathcal{A}$ to win G_i, that is $\Pr[\mathsf{G}_i(\mathcal{A}, 1^\lambda) = 1]$, where the probability is taken over the random coins of G_i and $\mathcal{A}$.

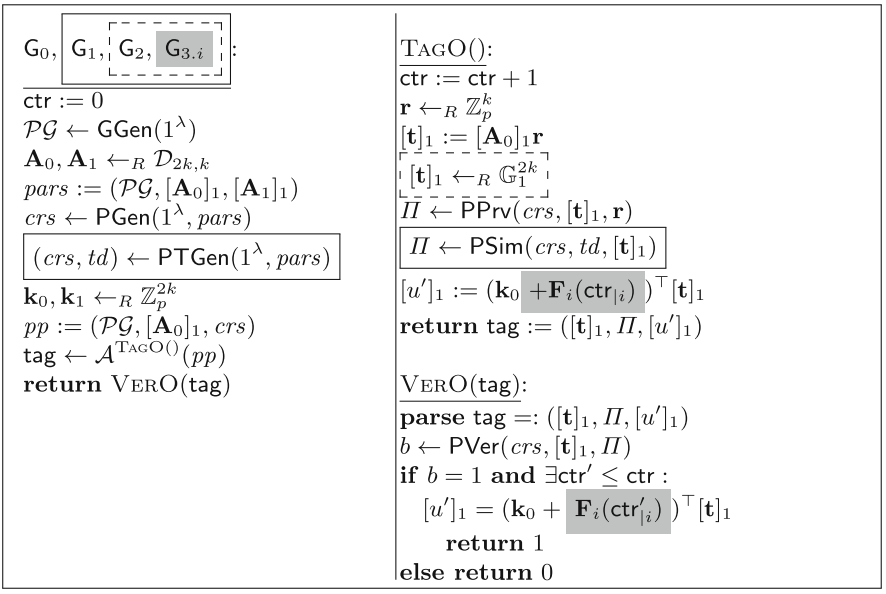

Fig. 4. Games $\mathsf{G}_0, \mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_{3.i}$ for $i \in \{0, \ldots, \lceil \log Q \rceil - 1\}$, for the proof of the core lemma (Lemma 4). $\mathbf{F}_i : \{0,1\}^i \to \mathbb{Z}_p^{2k}$ denotes a random function, and $\mathsf{ctr}_{|i}$ denotes the i-bit prefix of the counter ctr written in binary. In each procedure, the components inside a solid (dotted, gray) frame are only present in the games marked by a solid (dotted, gray) frame.

G_0: We have $\mathsf{G}_0 = \mathrm{Exp}^{\mathrm{core}}_{0,\mathcal{A}}(\lambda)$ and thus by definition:

$$\varepsilon_0 = \mathrm{Adv}^{\mathrm{core}}_{0,\mathcal{A}}(\lambda).$$

$\mathsf{G}_0 \rightsquigarrow \mathsf{G}_1$: Game G_1 is as G_0, except that *crs* is generated by PTGen and the proofs computed by TagO are generated using PSim instead of PPrv. This change is justified by the zero-knowledge of PS. Namely, let $\mathcal{A}$ be an adversary distinguishing between G_0 and G_1. Then we can construct an adversary

$\mathcal{B}$ on the composable zero-knowledge property of PS as follows. The adversary $\mathcal{B}$ follows G_0, except he uses the *crs* obtained by its own experiment instead of calling PGen. $\mathcal{B}$ answers tag queries following the tag oracle, but instead of computing Π itself it asks its own oracle PROVE. Now $\mathcal{B}$ simulates G_0 in case it was given a real *crs* and it simulates G_1 in case it was given a *crs* generated by PTGen. $\mathcal{B}$ is thus such that $T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ and

$$|\varepsilon_0 - \varepsilon_1| \leq \mathrm{Adv}^{\mathrm{ZK}}_{\mathsf{PS},\mathcal{B}}(\lambda).$$

$\mathsf{G}_1 \rightsquigarrow \mathsf{G}_2$: We can switch $[\mathbf{t}]_1$ to random over $\mathbb{G}_1$ by applying the $\mathcal{D}_{2k,k}$ assumption. More precisely, let $\mathcal{A}$ be an adversary distinguishing between G_1 and G_2 and let $\mathcal{B}$ be an adversary given a Q-fold $\mathcal{D}_{2k,k}$-MDDH challenge $(\mathcal{PG}, [\mathbf{A}_0]_1, [\mathbf{z}_1]_1, \ldots, [\mathbf{z}_Q]_1)$ as input. Now $\mathcal{B}$ sets up the game for $\mathcal{A}$ similar to G_1, but instead choosing $\mathbf{A}_0 \leftarrow_R \mathcal{D}_{2k,k}$, it uses its challenge matrix $[\mathbf{A}_0]_1$ as part of the public parameters *pars*. Further, to answer tag queries $\mathcal{B}$ sets $[\mathbf{t}_i]_1 := [\mathbf{z}_i]_1$ and computes the rest accordingly. This is possible as the proof Π is simulated from game G_1 on. In case $\mathcal{B}$ was given a real $\mathcal{D}_{2k,k}$-challenge, it simulates G_1 and otherwise G_2. Lemma 1 yields the existence of an adversary $\mathcal{B}_1$ with $T(\mathcal{B}_1) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ and

$$|\varepsilon_1 - \varepsilon_2| \leq k \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_1,\mathcal{D}_{2k,k},\mathcal{B}_1}(\lambda) + \tfrac{1}{p-1}.$$

$\mathsf{G}_2 \rightsquigarrow \mathsf{G}_{3.0}$: As for all $\mathsf{ctr} \in \mathbb{N}$ we have $\mathbf{F}_0(\mathsf{ctr}_{|0}) = \mathbf{F}_0(\epsilon)$ and $\mathbf{k}_0$ is distributed identically to $\mathbf{k}_0 + \mathbf{F}_0(\epsilon)$ for $\mathbf{k}_0 \leftarrow_R \mathbb{Z}_p^{2k}$ we have

$$\varepsilon_2 = \varepsilon_{3.0}.$$

$\mathsf{G}_{3.i} \rightsquigarrow \mathsf{G}_{3.(i+1)}$: For the proof of this transition we refer to the full version. We obtain: For every adversary $\mathcal{A}$ there exist adversaries $\mathcal{B}_i$, $\mathcal{B}'_i$ such that $T(\mathcal{B}_i) \approx T(\mathcal{B}'_i) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$, and

$$\begin{aligned}\varepsilon_{3.i} \leq & \varepsilon_{3.(i+1)} + 4k \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_1,\mathcal{D}_{2k,k},\mathcal{B}_i}(\lambda) + 2\mathrm{Adv}^{\mathrm{ZK}}_{\mathsf{PS},\mathcal{B}'_i}(\lambda) \\ & + \Delta_{\mathcal{D}_{2k,k}} + \tfrac{4}{p-1} + \tfrac{Q}{p}.\end{aligned}$$

$\mathsf{G}_{3.\lceil \log Q \rceil} \rightsquigarrow \mathbf{Exp}^{\mathbf{core}}_{\mathbf{1},\mathcal{A}}(\boldsymbol{\lambda})$: It is left to reverse the changes introduced in the transitions from game G_0 to game G_2 to end up at the experiment $\mathrm{Exp}^{\mathrm{core}}_{1,\mathcal{A}}(1^\lambda)$.

In order to do so we introduce an intermediary game G_4, where we set $[\mathbf{t}] := [\mathbf{A}_0]_1\mathbf{r}$ for $\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$. This corresponds to reversing transition $\mathsf{G}_1 \rightsquigarrow \mathsf{G}_2$. By the same reasoning for every adversary $\mathcal{A}$ we thus obtain an adversary $\mathcal{B}_{3.\lceil \log Q \rceil}$ with $T(\mathcal{B}_{3.\lceil \log Q \rceil}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ such that

$$|\varepsilon_{3.\lceil \log Q \rceil} - \varepsilon_4| \leq k \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_1,\mathcal{D}_{2k,k},\mathcal{B}_{3.\lceil \log Q \rceil}}(\lambda) + \tfrac{1}{p-1}.$$

As $[\mathbf{t}]_1$ is now chosen from $\mathrm{span}([\mathbf{A}_0]_1)$ again, we can switch back to honest generation of the common reference string *crs* and proofs Π. As in transition

$\mathsf{G}_0 \rightsquigarrow \mathsf{G}_1$ for an adversary $\mathcal{A}$ we obtain an adversary $\mathcal{B}_4$ with $T(\mathcal{B}_4) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ and

$$|\varepsilon_4 - \mathrm{Adv}^{\mathrm{core}}_{1,\mathcal{A}}(\lambda)| \leq \mathrm{Adv}^{\mathrm{ZK}}_{\mathsf{PS},\mathcal{B}_4}(\lambda).$$

Theorem 1 (UF-CMA security of MAC). *If the $\mathcal{D}_{2k,k}$-MDDH assumptions holds in $\mathbb{G}_1$, and the tuple $\mathsf{PS} := (\mathsf{PGen}, \mathsf{PTGen}, \mathsf{PPrv}, \mathsf{PVer})$ is a non-interactive zero-knowledge proof system for $\mathcal{L}^{\vee}_{\mathbf{A}_0,\mathbf{A}_1}$, then the MAC $\mathsf{MAC} := (\mathsf{Gen}, \mathsf{Tag}, \mathsf{Ver})$ provided in Fig. 2 is $\mathsf{UF\text{-}CMA}$ secure. Namely, for any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ with running time $T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$, where Q is the number of queries to $\textsc{TagO}$, poly is independent of Q, and*

$$\mathrm{Adv}^{\text{uf-cma}}_{\mathsf{MAC},\mathcal{A}}(\lambda) \leq \Delta^{\mathrm{core}}_{\mathcal{B}}(\lambda) + \tfrac{Q}{p}.$$

Proof. We employ an intermediary game G_0 to prove $\mathsf{UF\text{-}CMA}$ security of the MAC. By ε_0 we denote the advantage of $\mathcal{A}$ to win game G_0, that is $\Pr[\mathsf{G}_0(\mathcal{A}, 1^\lambda) = 1]$, where the probability is taken over the random coins of G_0 and $\mathcal{A}$.

$\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{A}}(\lambda) \rightsquigarrow \mathbf{G_0}$: Let $\mathcal{A}$ be an adversary distinguishing between $\mathrm{Exp}^{\text{uf-cma}}_{\mathcal{A}}(\lambda)$ and G_0. Then we construct an adversary $\mathcal{B}$ with $T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ allowing to break the core lemma (Lemma 4) as follows. On input pp from $\mathrm{Exp}^{\mathrm{core}}_{\beta}(1^\lambda, \mathcal{B})$ the adversary $\mathcal{B}$ forwards pp to $\mathcal{A}$. Then, $\mathcal{B}$ samples $\mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$. Afterwards, on a tag query μ from $\mathcal{A}$, $\mathcal{B}$ queries its own $\textsc{TagO}$ oracle (which takes no input), receives $([\boldsymbol{t}]_1, \Pi, [u']_1)$, computes $[u]_1 := [u']_1 + \mu \mathbf{k}_1^\top [\mathbf{t}]_1$, and answers with $([\mathbf{t}]_1, \Pi, [u]_1)$. Finally, given the forgery $\big(\mu^\star, \mathsf{tag}^\star := ([\mathbf{t}]_1, \Pi, [u^\star]_1)\big)$ from $\mathcal{A}$, if $\mu^\star \notin \mathcal{Q}_{\mathsf{tag}}$ and $[u^\star]_1 \neq [0]_1$, then the adversary $\mathcal{B}$ sends $\mathsf{tag}' := ([\mathbf{t}]_1, \Pi, [u^\star]_1 + \mu \mathbf{k}_1^\top [\mathbf{t}]_1)$ to its experiment (otherwise an invalid tuple). Then we have $\mathrm{Adv}^{\text{uf-cma}}_{\mathsf{MAC},\mathcal{A}}(\lambda) = \mathrm{Adv}^{\mathrm{core}}_{0,\mathcal{B}}(\lambda)$ and $\varepsilon_0 = \mathrm{Adv}^{\mathrm{core}}_{1,\mathcal{B}}(\lambda)$. The core lemma (Lemma 4) yields

$$\mathrm{Adv}^{\mathrm{core}}_{0,\mathcal{B}}(\lambda) \leq \mathrm{Adv}^{\mathrm{core}}_{1,\mathcal{B}}(\lambda) + \Delta^{\mathrm{core}}_{\mathcal{B}}(\lambda)$$

and thus altogether we obtain

$$\mathrm{Adv}^{\text{uf-cma}}_{\mathsf{MAC},\mathcal{A}}(\lambda) \leq \varepsilon_0 + \Delta^{\mathrm{core}}_{\mathcal{B}}(\lambda).$$

Game $\mathbf{G_0}$: We now prove that any adversary $\mathcal{A}$ has only negligible chances to win game G_0 using the randomness of $\mathbf{F}$ together with the pairwise independence of $\mu \mapsto \mathbf{k}_0 + \mu \mathbf{k}_1$.

Let $\big(\mu^\star, \mathsf{tag}^\star\big)$ be the forgery of $\mathcal{A}$. we can replace $\mathbf{k}_1$ by $\mathbf{k}_1 - \mathbf{v}$ for $\mathbf{v} \leftarrow_R \mathbb{Z}_p^{2k}$, as both are distributed identically. Next, for all $j \leq Q$ we can replace $\mathbf{F}(j)$ by $\mathbf{F}(j) + \mu^{(j)} \cdot \mathbf{v}$ for the same reason. This way, $\textsc{TagO}(\mu^{(j)})$ computes

$$\begin{aligned}[u^{(j)}]_1 :&= [(\mathbf{k}_0 + \mu^{(j)}\mathbf{k}_1 \boxed{-\mu^{(j)}\mathbf{v}} + \mathbf{F}(j) \boxed{+\mu^{(j)}\mathbf{v}})^\top \mathbf{t}^{(j)}]_1 \\ &= [(\mathbf{k}_0 + \mu^{(j)}\mathbf{k}_1 + \mathbf{F}(j)^\top \mathbf{t}^{(j)}]_1,\end{aligned}$$

and $\textsc{VerO}\big([\mu^\star]_2, \mathsf{tag}^\star := ([\mathbf{t}]_1, \Pi, [u])\big)$ checks if there exists a counter $i \in \mathcal{Q}_{\mathsf{tag}}$ such that:

$$\begin{aligned}[u]_1 &= [(\mathbf{k}_0 + \mu^\star \mathbf{k}_1 \boxed{-\mu^\star \mathbf{v}} + \mathbf{F}(i) \boxed{+\mu^{(i)}\mathbf{v}})^\top \mathbf{t}]_1 \\ &= [(\mathbf{k}_0 + \mu^\star \mathbf{k}_1 + \mathbf{F}(i)^\top \mathbf{t}^\star]_1 \boxed{+[(\mu^{(i)} - \mu^\star)\mathbf{v}^\top \mathbf{t}]_1}.\end{aligned}$$

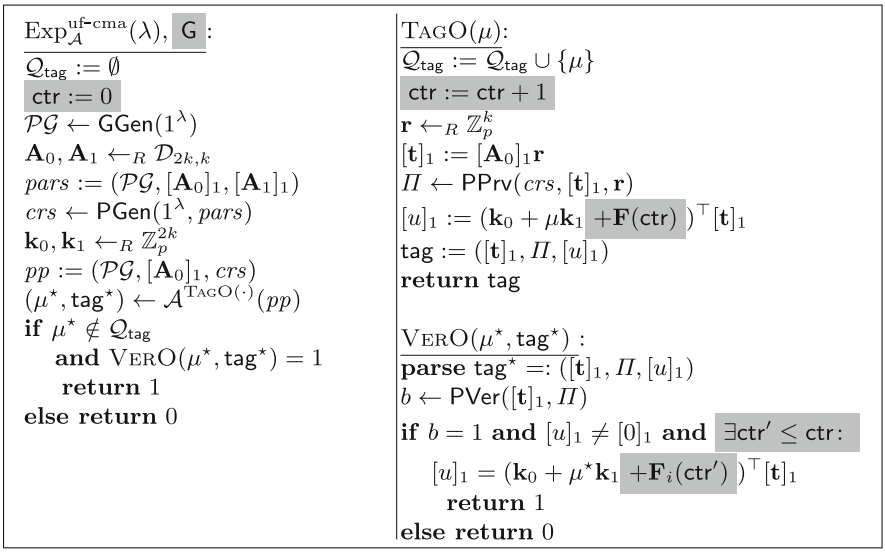

Fig. 5. The UF-CMA security experiment and game G for the UF-CMA proof of MAC in Fig. 2. $\mathbf{F} : \{0,1\}^{\lceil \log Q \rceil} \to \mathbb{Z}_p^{2k}$ denotes a random function, applied on ctr written in binary. In each procedure, the components inside a gray frame are only present in the games marked by a gray frame.

For the forgery to be successful, it must hold $\mu^\star \notin \mathcal{Q}_{\mathsf{tag}}$ and $[u] \neq 0$ (and thus $[\mathbf{t}]_1 \neq [\mathbf{0}]_1$). Therefore, each value computed by VERO is (marginally) uniformly random over $\mathbb{G}_1$.

As the verification oracle checks for all counters $i \leq Q$, applying the union bound yields

$$\varepsilon_0 \leq \frac{Q}{p}.$$

4 Tightly Secure Signature Scheme

In this section, we present a signature scheme SIG for signing messages from $\mathbb{Z}_p$, described in Fig. 6, whose UF-CMA security can be tightly reduced to the $\mathcal{D}_{2k,k}$-MDDH and $\mathcal{D}_k$-MDDH assumptions.

SIG builds upon the tightly secure MAC from Sect. 3, and functions as a stepping stone to explain the main ideas of the upcoming structure-preserving signature in Sect. 5. Recall that our MAC outputs $\mathsf{tag} = ([\mathbf{t}]_1, \Pi, [u]_1)$, where Π is a (publicly verifiable) NIZK proof of the statement $\mathbf{t} \in \mathrm{span}(\mathbf{A}_0) \cup \mathrm{span}(\mathbf{A}_1)$, and $u = (\mathbf{k}_0 + \mu \mathbf{k}_1)^\top \mathbf{t}$ has an affine structure. Hence, alternatively, we can also view our MAC as an affine MAC [14] with $\mathbf{t} \in \mathrm{span}(\mathbf{A}_0) \cup \mathrm{span}(\mathbf{A}_1)$ and a NIZK proof for that. Similar to [14], we use (tuned) Groth-Sahai proofs to make $[u]_1$ publicly verifiable. Similar ideas have been used to construct efficient quasi-adaptive NIZK for linear subspace [38,40], structure-preserving signatures [39],

$\underline{\mathsf{Gen}(1^\lambda)}$:
$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$
$\mathbf{A}_0, \mathbf{A}_1 \leftarrow \mathcal{D}_{2k,k}$
$pars := (\mathcal{PG}, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1)$
$crs \leftarrow \mathsf{PGen}(1^\lambda, pars)$
$\mathbf{A} \leftarrow_R \mathcal{D}_k$
$\mathbf{K}_0, \mathbf{K}_1 \leftarrow_R \mathbb{Z}_p^{2k\times(k+1)}$
$pk := (\mathcal{PG}, [\mathbf{A}_0]_1, crs,$
$\quad [\mathbf{A}]_2, [\mathbf{K}_0\mathbf{A}]_2, [\mathbf{K}_1\mathbf{A}]_2)$
$sk := (\mathbf{K}_0, \mathbf{K}_1)$
return (pk, sk)

$\underline{\mathsf{Sign}(pk, sk, \mu \in \mathbb{Z}_p)}$:
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{t}]_1 := [\mathbf{A}_0]_1\mathbf{r}$
$\Pi \leftarrow \mathsf{PPrv}(crs, [\mathbf{t}]_1, \mathbf{r})$
$[\mathbf{u}]_1 := (\mathbf{K}_0 + \mu\mathbf{K}_1)^\top [\mathbf{t}]_1$
$\sigma := ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1)$
return σ

$\underline{\mathsf{Ver}(pk, \mu \in \mathbb{Z}_p, \sigma)}$:
parse tag $=: ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1)$
$b \leftarrow \mathsf{PVer}(crs, [\mathbf{t}]_1, \Pi)$
if $b = 1$ **and** $[\mathbf{u}]_1 \neq [\mathbf{0}]_1$ **and** $e([\mathbf{u}]_1^\top, [\mathbf{A}]_2)$
$\quad = e([\mathbf{t}]_1^\top, [\mathbf{K}_0\mathbf{A}]_2 + \mu[\mathbf{K}_1\mathbf{A}]_2)$
$\quad$ **return** 1
else return 0

Fig. 6. Tightly UF-CMA secure signature scheme SIG.

and identity-based encryption schemes [14]. In the following theorem we state the state the security of SIG. For a proof we refer to the full version.

Theorem 2 (Security of SIG). *If* $\mathsf{PS} := (\mathsf{PGen}, \mathsf{PPrv}, \mathsf{PVer}, \mathsf{PSim})$ *is a non-interactive zero-knowledge proof system for* $\mathcal{L}^\vee_{\mathbf{A}_0,\mathbf{A}_1}$*, then the signature scheme* SIG *described in Fig. 6 is* UF-CMA *secure under the* $\mathcal{D}_{2k,k}$*-MDDH and* $\mathcal{D}_k$*-MDDH assumptions. Namely, for any adversary* $\mathcal{A}$*, there exist adversaries* $\mathcal{B}, \mathcal{B}'$ *with running time* $T(\mathcal{B}) \approx T(\mathcal{B}') \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$*, where* Q *is the number of queries to* SIGNO*,* poly *is independent of* Q*, and*

$$\mathrm{Adv}^{\mathrm{uf-cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathrm{uf-cma}}_{\mathsf{MAC},\mathcal{B}}(\lambda) + \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_2,\mathcal{D}_k,\mathcal{B}'}(\lambda).$$

5 Tightly Secure Structure-Preserving Signature Scheme

In this section we present a structure-preserving signature scheme SPS, described in Fig. 7, whose security can be tightly reduced to the $\mathcal{D}_{2k,k}$-MDDH and $\mathcal{D}_k$-MDDH assumptions. It builds upon the tightly secure signature presented in Sect. 4 by using a similar idea of [39]. Precisely, we view μ as a label and the main difference between both schemes is that in the proof we do not need to guess which μ the adversary may reuse for its forgery, and thus our security proof is tight.

Theorem 3 (Security of SPS). *If* $\mathsf{PS} := (\mathsf{PGen}, \mathsf{PTGen}, \mathsf{PVer}, \mathsf{PSim})$ *is a non-interactive zero-knowledge proof system for* $\mathcal{L}^\vee_{\mathbf{A}_0,\mathbf{A}_1}$*, the signature scheme* SPS *described in Fig. 7 is* UF-CMA *secure under the* $\mathcal{D}_{2k,k}$*-MDDH and* $\mathcal{D}_k$*-MDDH assumptions. Namely, for any adversary* $\mathcal{A}$*, there exist adversaries* $\mathcal{B}, \mathcal{B}'$ *with running time* $T(\mathcal{B}) \approx T(\mathcal{B}') \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$*, where* Q *is the number of queries to* SIGNO*,* poly *is independent of* Q*, and*

$$\mathrm{Adv}^{\mathrm{uf-cma}}_{\mathsf{SPS},\mathcal{A}}(\lambda) \leq \Delta^{\mathrm{core}}_{\mathcal{B}}(\lambda) + \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_2,\mathcal{D}_k,\mathcal{B}'}(\lambda) + \frac{Q}{p^k} + \frac{Q}{p}.$$

$\underline{\mathsf{Gen}(1^\lambda)}$:
$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$
$\mathbf{A}_0, \mathbf{A}_1 \leftarrow_R \mathcal{D}_{2k,k}$
$pars := (\mathcal{PG}, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1)$
$crs \leftarrow \mathsf{PGen}(pars, 1^\lambda)$
$\mathbf{A} \leftarrow_R \mathcal{D}_k$
$\mathbf{K}_0 \leftarrow_R \mathbb{Z}_p^{2k\times(k+1)}$
$\mathbf{K} \leftarrow_R \mathbb{Z}_p^{(n+1)\times(k+1)}$
$pk := (\mathcal{PG}, [\mathbf{A}_0]_1, crs, [\mathbf{A}]_2, [\mathbf{K}_0\mathbf{A}]_2, [\mathbf{K}\mathbf{A}]_2)$
$sk := (\mathbf{K}_0, \mathbf{K})$
return (pk, sk)

$\underline{\mathsf{Sign}(pk, sk, [\mathbf{m}]_1 \in \mathbb{G}_1^n)}$:
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$ $[\mathbf{t}]_1 := [\mathbf{A}_0]_1\mathbf{r}$
$\Pi \leftarrow \mathsf{PPrv}(crs, [\mathbf{t}]_1, \mathbf{r})$
$[\mathbf{u}]_1 := \mathbf{K}_0^\top [\mathbf{t}]_1 + \mathbf{K}^\top \begin{bmatrix}\mathbf{m}\\1\end{bmatrix}_1$
return $\sigma := ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1)$

$\underline{\mathsf{Ver}(pk, \sigma, [\mathbf{m}]_1)}$:
parse $\sigma := ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1)$
$b \leftarrow \mathsf{PVer}(pk, [\mathbf{t}]_1, \Pi)$
if $b = 1$ **and** $e([\mathbf{u}]_1^\top, [\mathbf{A}]_2) = e([\mathbf{t}]_1^\top, [\mathbf{K}_0\mathbf{A}]_2) + e(\begin{bmatrix}\mathbf{m}\\1\end{bmatrix}_1^\top, [\mathbf{K}\mathbf{A}]_2)$
return 1
else return 0

Fig. 7. Tightly UF-CMA secure structure-preserving signature scheme SPS with message space $\mathbb{G}_1^n$.

When using PS *from Sect. 2.5, we obtain*

$$\begin{aligned}\mathrm{Adv}^{\text{uf-cma}}_{\mathsf{SPS},\mathcal{A}}(\lambda) \leq & (4k\lceil\log Q\rceil + 2)\cdot \mathrm{Adv}^{\text{mddh}}_{\mathcal{PG},\mathbb{G}_1,\mathcal{D}_{2k,k},\mathcal{B}}(\lambda) \\ & + (2\lceil\log Q\rceil + 3)\cdot \mathrm{Adv}^{\text{mddh}}_{\mathcal{PG},\mathbb{G}_2,\mathcal{D}_k,\mathcal{B}'}(\lambda) + \lceil\log Q\rceil \cdot \Delta_{\mathcal{D}_{2k,k}} \\ & + \tfrac{4\lceil\log Q\rceil+2}{p-1} + \tfrac{(Q+1)\lceil\log Q\rceil+Q}{p} + \tfrac{Q}{p^k}.\end{aligned}$$

Strategy. In a nutshell, we will embed a "shadow MAC" in our signature scheme, and then invoke the core lemma to randomize the MAC tags computed during signing queries and the final verification of $\mathcal{A}$'s forgery. A little more specifically, we will embed a term $\mathbf{k}_0^\top\mathbf{t}$ into the $\mathbf{A}$-orthogonal space of each $\mathbf{u}$ computed by SignO and VerO. (Intuitively, changes to this $\mathbf{A}$-orthogonal space do not influence the verification key, and simply correspond to changing from one signing key to another signing key that is compatible with the same verification key.) Using our core lemma, we can randomize this term $\mathbf{k}_0^\top\mathbf{t}$ to $(\mathbf{k}_0 + \mathbf{F}(\mathsf{ctr}))^\top\mathbf{t}$ for a random function $\mathbf{F}$ and a signature counter ctr. Intuitively, this means that we use a freshly randomized signing key for each signature query. After these changes, an adversary only has a statistically small chance in producing a valid forgery.

Proof (of Theorem 3). We proceed via a series of hybrid games G_0 to G_2, described in Fig. 8. By ε_i we denote the advantage of $\mathcal{A}$ to win G_i.

$\mathbf{Exp}^{\text{uf-cma}}_{\mathsf{SPS},\mathcal{A}}(\lambda) \rightsquigarrow \mathbf{G_0}$: Here we change the verification oracle as described in Fig. 8.

Note that a pair $(\mu^\star, \sigma^\star)$ that passes VerO in G_0 always passes the VerO check in $\mathrm{Exp}^{\text{uf-cma}}_{\mathsf{SPS},\mathcal{A}}(\lambda)$. Thus, to bound $|\mathrm{Adv}^{\text{uf-cma}}_{\mathsf{SPS},\mathcal{A}}(\lambda) - \varepsilon_0|$, it suffices to bound the probability that $\mathcal{A}$ produces a tuple $(\mu^\star, \sigma^\star)$ that passes VerO in $\mathrm{Exp}^{\text{uf-cma}}_{\mathsf{SPS},\mathcal{A}}(\lambda)$,

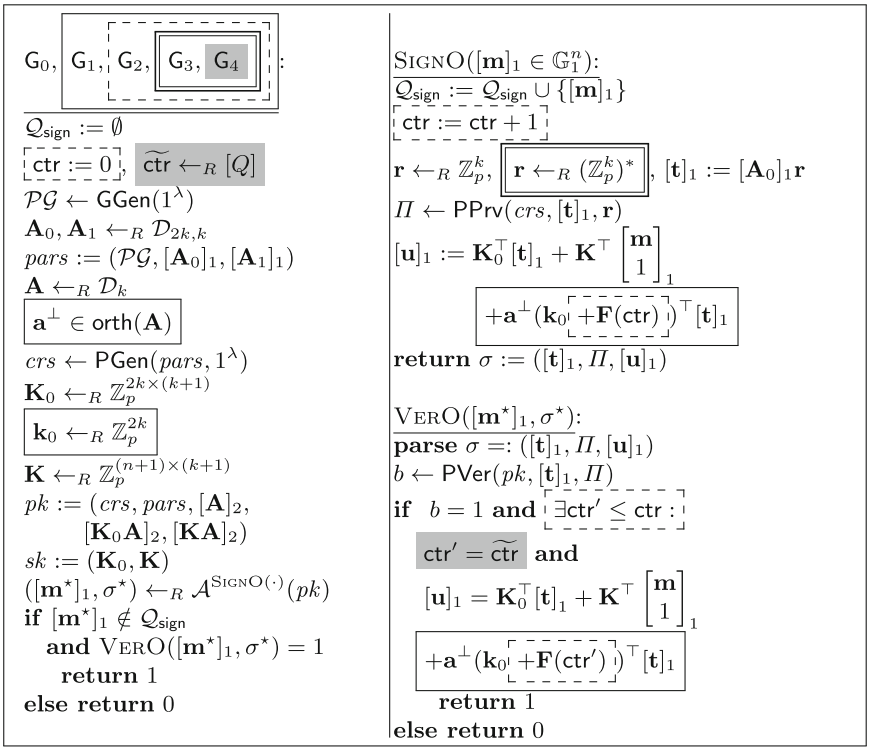

Fig. 8. Games G_0 to G_2 for proving Theorem 3. Here, $\mathbf{F} : \mathbb{Z}_p \to \mathbb{Z}_p^{2k}$ is a random function. In each procedure, the components inside a solid (dotted, double, gray) frame are only present in the games marked by a solid (dotted, double, gray) frame.

but not in G_0. For the signature $\sigma^\star =: ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1)$ we can write the verification equation in $\mathrm{Exp}^{\text{uf-cma}}_{\mathsf{SPS},\mathcal{A}}(\lambda)$ as

$$e([\mathbf{u}]_1^\top, [\mathbf{A}]_2) = e([\mathbf{t}]_1^\top, [\mathbf{K}_0\mathbf{A}]_2) + e(\begin{bmatrix}\mathbf{m}\\1\end{bmatrix}_1^\top, [\mathbf{K}\mathbf{A}]_2)$$
$$\Leftrightarrow e([\mathbf{u}]_1 - [\mathbf{t}]_1^\top\mathbf{K}_0 - \begin{bmatrix}\mathbf{m}\\1\end{bmatrix}_1^\top\mathbf{K}, [\mathbf{A}]_2) = \mathbf{0}$$

Observe that for any $(\mu^\star, ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1))$ that passes the verification equation in the experiment $\mathrm{Exp}^{\text{uf-cma}}_{\mathsf{SPS},\mathcal{A}}(\lambda)$, but not the one in G_0, the value

$$[\mathbf{u}]_1 - [\mathbf{t}]_1^\top\mathbf{K}_0 - \begin{bmatrix}\mathbf{m}\\1\end{bmatrix}_1^\top\mathbf{K}$$

is a non-zero vector in the kernel of $\mathbf{A}$. Thus, from $\mathcal{A}$ we can construct an adversary $\mathcal{B}$ against the $\mathcal{D}_k$-KMDH assumption. Finally, Lemma 2 yields an adversary

$\mathcal{B}'$ with $T(\mathcal{B}') \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ such that

$$|\mathrm{Adv}^{\text{uf-cma}}_{\mathsf{SPS},\mathcal{A}}(\lambda) - \varepsilon_0| \leq \mathrm{Adv}^{\text{mddh}}_{\mathcal{PG},\mathbb{G}_2,\mathcal{D}_k,\mathcal{B}}(\lambda).$$

$\mathsf{G_0} \rightsquigarrow \mathsf{G_1}$: We can replace $\mathbf{K}_0$ by $\mathbf{K}_0 + \mathbf{k}_0(\mathbf{a}^\perp)^\top$ for $\mathbf{a}^\perp \in \mathsf{orth}(\mathbf{A})$ and $\mathbf{k}_i \leftarrow_R \mathbb{Z}_p^{2k}$, as both are distributed identically. Note that this change does not show up in the public key pk. Looking ahead, this change will allow us to use the computational core lemma (Lemma 4). This yields

$$\varepsilon_0 = \varepsilon_1.$$

$\mathsf{G_1} \rightsquigarrow \mathsf{G_2}$: Let $\mathcal{A}$ be an adversary playing either G_1 or G_2. We build an adversary $\mathcal{B}$ such that $T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ and

$$\Pr[\mathrm{Exp}^{\text{core}}_{0,\mathcal{B}}(1^\lambda) = 1] = \varepsilon_1 \quad \text{and} \quad \Pr[\mathrm{Exp}^{\text{core}}_{1,\mathcal{B}}(1^\lambda) = 1] = \varepsilon_2.$$

This implies, by the core lemma (Lemma 4), that

$$\varepsilon_1 \leq \varepsilon_2 + \Delta^{\text{core}}_{\mathcal{B}}(\lambda).$$

We now describe $\mathcal{B}$ against $\mathrm{Exp}^{\text{core}}_{\beta,\mathcal{B}}(1^\lambda)$ for β equal to either 0 or 1. First, $\mathcal{B}$ receives $pp := (\mathcal{PG}, [\mathbf{A}_0]_1, crs)$ from $\mathrm{Exp}^{\text{core}}_{\beta,\mathcal{B}}(1^\lambda)$, then, $\mathcal{B}$ samples $\mathbf{A} \leftarrow_R \mathcal{D}_k$, $\mathbf{a}^\perp \in \mathsf{orth}(\mathbf{A})$, $\mathbf{K}_0 \leftarrow_R \mathbb{Z}_p^{2k\times(k+1)}$, $\mathbf{K} \leftarrow_R \mathbb{Z}_p^{(n+1)\times(k+1)}$ and forwards $pk := (\mathcal{PG}, [\mathbf{A}_0]_1, crs, [\mathbf{A}]_2, [\mathbf{K}_0\mathbf{A}]_2, [\mathbf{K}\mathbf{A}]_2)$ to $\mathcal{A}$.

To simulate $\textsc{Sign}\mathrm{O}([\mathbf{m}]_1)$, $\mathcal{B}$ uses its oracle $\textsc{Tag}\mathrm{O}$, which takes no input, and gives back $([\mathbf{t}]_1, \Pi, [u]_1)$. Then, $\mathcal{B}$ computes $[\mathbf{u}]_1 := \mathbf{K}_0^\top[\mathbf{t}]_1 + \mathbf{a}^\perp[u]_1 + \mathbf{K}^\top \begin{bmatrix}\mathbf{m}\\ 1\end{bmatrix}_1$, and returns $\sigma := ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1)$ to $\mathcal{A}$.

Finally, given the forgery $([\mathbf{m}^\star]_1, \sigma^\star)$ with corresponding signature $\sigma^\star := ([\mathbf{t}^\star]_1, \Pi^\star, [\mathbf{u}^\star]_1)$, $\mathcal{B}$ first checks if $[\mathbf{m}^\star]_1 \notin \mathcal{Q}_{\mathsf{sign}}$ and $[\mathbf{u}^\star]_1 \neq [\mathbf{0}]_1$. If it is not the case, then $\mathcal{B}$ returns 0 to $\mathcal{A}$. If it is the case, with the knowledge of $\mathbf{a}^\perp \in \mathbb{Z}_p$, $\mathcal{B}$ efficiently checks whether there exists $[u^\star]_1 \in \mathbb{G}_1$ such that $[\mathbf{u}^\star]_1 - \mathbf{K}_0^\top[\mathbf{t}^\star]_1 - \mathbf{K}^\top \begin{bmatrix}\mathbf{m}^\star\\ 1\end{bmatrix}_1 = [u^\star]_1\mathbf{a}^\perp$. If it is not the case, $\mathcal{B}$ returns 0 to $\mathcal{A}$. If it is the case, $\mathcal{B}$ computes $[u^\star]_1$ (it can do so efficiently given $\mathbf{a}^\perp$), sets $\mathsf{tag} := ([\mathbf{t}^\star]_1, \Pi^\star, [u^\star]_1)$, calls its verification oracle $\textsc{Ver}\mathrm{O}(\mathsf{tag})$, and forwards the answer to $\mathcal{A}$.

$\mathsf{G_2} \rightsquigarrow \mathsf{G_3}$: In game G_2 the vectors $\mathbf{r}$ sampled by $\textsc{Sign}\mathrm{O}$ are uniformly random over $\mathbb{Z}_p^k$, while they are uniformly random over $(\mathbb{Z}_p^k)^* = \mathbb{Z}_p^k \backslash \{0\}$ in G_3. Since this is the only difference between the games, the difference of advantage is bounded by the statistical distance between the two distributions of $\mathbf{r}$. A union bound over the number of queries yields

$$\varepsilon_2 - \varepsilon_3 \leq \frac{Q}{p^k}.$$

$\mathsf{G_3} \rightsquigarrow \mathsf{G_4}$: These games are the same except for the extra condition $\widetilde{\mathsf{ctr}} = \mathsf{ctr}'$ in G_4, which happens with probability $\frac{1}{Q}$ over the choice of $\widetilde{\mathsf{ctr}} \leftarrow_R [Q]$. Since the adversary view is independent of $\widetilde{\mathsf{ctr}}$, we have

$$\varepsilon_4 = \frac{\varepsilon_3}{Q}.$$

Game $\mathbf{G_4}$: We prove that $\varepsilon_4 \leq \frac{1}{p}$.

First, we can replace $\mathbf{K}$ by $\mathbf{K} + \mathbf{v}(\mathbf{a}^\perp)^\top$ for $\mathbf{v} \leftarrow_R \mathbb{Z}_p^{n+1}$, and $\{\mathbf{F}(i) : i \in [Q], i \neq \widetilde{\mathsf{ctr}}\}$ by $\{\mathbf{F}(i) + \mathbf{w}_i : i \in [Q], i \neq \widetilde{\mathsf{ctr}}\}$ for $\mathbf{w}_i \leftarrow_R \mathbb{Z}_p^{2k}$. Note that this does not change the distribution of the game.

Thus, for the i-th signing query with $i \neq \widetilde{\mathsf{ctr}}$ the value $\mathbf{u}$ is computed by $\textsc{SignO}([\mathbf{m}_i]_1)$ as

$$[\mathbf{u}]_1 = \mathbf{K}_0^\top [\mathbf{t}]_1 + (\mathbf{K}^\top + \mathbf{a}^\perp \mathbf{v}^\top) \begin{bmatrix} \mathbf{m}_i \\ 1 \end{bmatrix}_1 + \mathbf{a}^\perp (\mathbf{k}_0 + \mathbf{F}(i) + \mathbf{w}_i)^\top [\mathbf{t}]_1,$$

with $[\mathbf{t}]_1 := [\mathbf{A}_0]_1 \mathbf{r}$, $\mathbf{r} \leftarrow_R (\mathbb{Z}_p^k)^*$. This is identically distributed to

$$[\mathbf{u}]_1 = \mathbf{K}_0^\top [\mathbf{t}]_1 + \mathbf{K}^\top \begin{bmatrix} \mathbf{m}_i \\ 1 \end{bmatrix}_1 + \gamma_i \cdot \mathbf{a}^\perp, \text{ with } \gamma_i \leftarrow_R \mathbb{Z}_p.$$

For the $\widetilde{\mathsf{ctr}}$'th signing query, we have

$$[\mathbf{u}]_1 = \mathbf{K}_0^\top [\mathbf{t}]_1 + (\mathbf{K}^\top + \mathbf{a}^\perp \mathbf{v}^\top) \begin{bmatrix} \mathbf{m}_{\widetilde{\mathsf{ctr}}} \\ 1 \end{bmatrix}_1 + \mathbf{a}^\perp (\mathbf{k}_0 + \mathbf{F}(\widetilde{\mathsf{ctr}}))^\top [\mathbf{t}]_1.$$

Assuming $\mathcal{A}$ succeeds in producing a valid forgery, $\textsc{VerO}$ computes

$$[\mathbf{u}^\star]_1 = \mathbf{K}_0^\top [\mathbf{t}^\star]_1 + (\mathbf{K}^\top + \mathbf{a}^\perp \mathbf{v}^\top) \begin{bmatrix} \mathbf{m}^\star \\ 1 \end{bmatrix}_1 + \mathbf{a}^\perp (\mathbf{k}_0 + \mathbf{F}(\widetilde{\mathsf{ctr}}))^\top [\mathbf{t}]_1.$$

Since $\mathbf{m}^\star \neq \mathbf{m}_{\widetilde{\mathsf{ctr}}}$ by definition of the security game, we can use the pairwise independence of $\mathbf{m} \mapsto \mathbf{v}^\top \begin{bmatrix} \mathbf{m} \\ 1 \end{bmatrix}_1$ to argue that $\mathbf{v}^\top \begin{bmatrix} \mathbf{m}^\star \\ 1 \end{bmatrix}_1$ and $\mathbf{v}^\top \begin{bmatrix} \mathbf{m}_{\widetilde{\mathsf{ctr}}} \\ 1 \end{bmatrix}_1$ are two independent values, uniformly random over $\mathbb{G}_1$. Thus, the verification equation is satisfied with probability at most $\frac{1}{p}$, that is

$$\varepsilon_4 \leq \tfrac{1}{p}.$$

Bilateral Structure-Preserving Signature Scheme. Our structure-preserving signature scheme, SPS, defined in Fig. 7 can sign only messages from $\mathbb{G}_1^n$. By applying the generic transformation from [39, Sect. 6], we can transform our SPS to sign messages from $\mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2}$ using their two-tier SPS, which is a generalization of [1]. The transformation is tightness-preserving by Theorem 6 of [39] and costs additional k elements from $\mathbb{G}_1$ and $k+1$ elements from $\mathbb{G}_2$ in the signature. For the SXDH assumption ($k = 1$), our bilateral SPS scheme requires additional 1 element from $\mathbb{G}_1$ and 2 elements from $\mathbb{G}_2$ in the signature.

References

1. Abe, M., Chase, M., David, B., Kohlweiss, M., Nishimaki, R., Ohkubo, M.: Constant-size structure-preserving signatures: generic constructions and simple assumptions. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 4–24. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34961-4_3
2. Abe, M., Chase, M., David, B., Kohlweiss, M., Nishimaki, R., Ohkubo, M.: Constant-size structure-preserving signatures: generic constructions and simple assumptions. J. Cryptol. **29**(4), 833–878 (2016). https://doi.org/10.1007/s00145-015-9211-7
3. Abe, M., David, B., Kohlweiss, M., Nishimaki, R., Ohkubo, M.: Tagged one-time signatures: tight security and optimal tag size. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 312–331. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_20
4. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. J. Cryptol. **29**(2), 363–421 (2016). https://doi.org/10.1007/s00145-014-9196-7
5. 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). https://doi.org/10.1007/978-3-642-22792-9_37
6. Abe, M., Hofheinz, D., Nishimaki, R., Ohkubo, M., Pan, J.: Compact structure-preserving signatures with almost tight security. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10402, pp. 548–580. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_19
7. Acar, T., Lauter, K., Naehrig, M., Shumow, D.: Affine pairings on ARM. In: Abdalla, M., Lange, T. (eds.) Pairing 2012. LNCS, vol. 7708, pp. 203–209. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36334-4_13
8. Attrapadung, N., Hanaoka, G., Yamada, S.: A framework for identity-based encryption with almost tight security. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 521–549. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_22
9. Barreto, P.S.L.M., Costello, C., Misoczki, R., Naehrig, M., Pereira, G.C.C.F., Zanon, G.: Subgroup security in pairing-based cryptography. In: Lauter, K., Rodríguez-Henríquez, F. (eds.) LATINCRYPT 2015. LNCS, vol. 9230, pp. 245–265. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22174-8_14
10. Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: P-signatures and non-interactive anonymous credentials. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 356–374. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78524-8_20
11. 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). https://doi.org/10.1007/3-540-45539-6_18
12. Bellare, M., Goldwasser, S.: New paradigms for digital signatures and message authentication based on non-interactive zero knowledge proofs. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 194–211. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_19

13. Blazy, O., Fuchsbauer, G., Izabachène, M., Jambert, A., Sibert, H., Vergnaud, D.: Batch Groth–Sahai. In: Zhou, J., Yung, M. (eds.) ACNS 2010. LNCS, vol. 6123, pp. 218–235. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13708-2_14
14. Blazy, O., Kiltz, E., Pan, J.: (Hierarchical) identity-based encryption from affine message authentication. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 408–425. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_23
15. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: 20th ACM STOC, pp. 103–112. ACM Press, May 1988
16. 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). https://doi.org/10.1007/3-540-36563-X_7
17. Camenisch, J., Dubovitskaya, M., Haralambiev, K.: Efficient structure-preserving signature scheme from standard assumptions. In: Visconti, I., De Prisco, R. (eds.) SCN 2012. LNCS, vol. 7485, pp. 76–94. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32928-9_5
18. 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). https://doi.org/10.1007/978-3-642-10366-7_11
19. Chase, M., Kohlweiss, M.: A new hash-and-sign approach and structure-preserving signatures from DLIN. In: Visconti, I., De Prisco, R. (eds.) SCN 2012. LNCS, vol. 7485, pp. 131–148. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32928-9_8
20. Chen, J., Gong, J., Weng, J.: Tightly secure IBE under constant-size master public key. In: Fehr, S. (ed.) PKC 2017. LNCS, vol. 10174, pp. 207–231. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54365-8_9
21. Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 435–460. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_25
22. Dodis, Y., Kiltz, E., Pietrzak, K., Wichs, D.: Message authentication, revisited. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 355–374. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_22
23. Enge, A., Milan, J.: Implementing cryptographic pairings at standard security levels. In: Chakraborty, R.S., Matyas, V., Schaumont, P. (eds.) SPACE 2014. LNCS, vol. 8804, pp. 28–46. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12060-7_3
24. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 129–147. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_8
25. Gay, R., Hofheinz, D., Kiltz, E., Wee, H.: Tightly CCA-secure encryption without pairings. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 1–27. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_1
26. Gay, R., Hofheinz, D., Kohl, L.: Kurosawa-Desmedt meets tight security. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10403, pp. 133–160. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_5

27. Gong, J., Chen, J., Dong, X., Cao, Z., Tang, S.: Extended nested dual system groups, revisited. In: Cheng, C.-M., Chung, K.-M., Persiano, G., Yang, B.-Y. (eds.) PKC 2016. LNCS, vol. 9614, pp. 133–163. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49384-7_6
28. Grewal, G., Azarderakhsh, R., Longa, P., Hu, S., Jao, D.: Efficient implementation of bilinear pairings on ARM processors. In: Knudsen, L.R., Wu, H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 149–165. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35999-6_11
29. 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). https://doi.org/10.1007/11935230_29
30. Groth, J., Lu, S.: A non-interactive shuffle with pairing based verifiability. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 51–67. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76900-2_4
31. Groth, J., Ostrovsky, R., Sahai, A.: New techniques for noninteractive zero-knowledge. J. ACM **59**(3), 1–35 (2012). https://doi.org/10.1145/2220357.2220358. ISSN: 0004-5411. http://doi.acm.org/10.1145/2220357.2220358
32. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_24
33. Hofheinz, D.: Adaptive partitioning. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10212, pp. 489–518. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56617-7_17
34. Hofheinz, D.: Algebraic partitioning: fully compact and (almost) tightly secure cryptography. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9562, pp. 251–281. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49096-9_11
35. Hofheinz, D., Jager, T.: Tightly secure signatures and public-key encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 590–607. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_35
36. Hofheinz, D., Koch, J., Striecks, C.: Identity-based encryption with (almost) tight security in the multi-instance, multi-ciphertext setting. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 799–822. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_36
37. Jutla, C.S., Roy, A.: Improved structure preserving signatures under standard bilinear assumptions. In: Fehr, S. (ed.) PKC 2017. LNCS, vol. 10175, pp. 183–209. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54388-7_7
38. Jutla, C.S., Roy, A.: Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8617, pp. 295–312. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44381-1_17
39. Kiltz, E., Pan, J., Wee, H.: Structure-preserving signatures from standard assumptions, revisited. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 275–295. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_14
40. Kiltz, E., Wee, H.: Quasi-adaptive NIZK for linear subspaces revisited. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 101–128. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_4
41. 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). https://doi.org/10.1007/978-3-540-28628-8_26

42. Libert, B., Joye, M., Yung, M., Peters, T.: Concise multi-challenge CCA-secure encryption and signatures with almost tight security. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8874, pp. 1–21. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45608-8_1
43. Libert, B., Peters, T., Joye, M., Yung, M.: Compactly hiding linear spans. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 681–707. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_28
44. Libert, B., Peters, T., Yung, M.: Short group signatures via structure-preserving signatures: standard model security from simple assumptions. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 296–316. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_15
45. Morillo, P., Ràfols, C., Villar, J.L.: The kernel matrix Diffie-Hellman assumption. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10031, pp. 729–758. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_27
46. Ràfols, C.: Stretching Groth-Sahai: NIZK proofs of partial satisfiability. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 247–276. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46497-7_10