More Efficient (Almost) Tightly Secure Structure-Preserving Signatures

Romain Gay[1] *, Dennis Hofheinz[2] †, Lisa Kohl[2] ‡, and Jiaxin Pan[2] §

[1] Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, Paris, France, and INRIA
rgay@di.ens.fr

[2] 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.

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 [29, 18, 5, 19, 17, 35, 45, 40, 4, 2, 38, 6] (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

[*] Supported by ERC Project aSCEND (639554), and a Google PhD fellowship.
[†] Supported by ERC Project PREP-CRYPTO (724307), and by DFG grants HO 4534/4-1 and HO 4534/2-2.
[‡] Supported by ERC Project PREP-CRYPTO (724307), and by DFG grant HO 4534/2-2.
[§] Supported by DFG grant HO 4534/4-1.

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.[3] 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[4] secure cryptographic schemes such as public-key encryption schemes [11, 35, 43, 44, 34, 25, 33], identity-based encryption schemes [21, 14, 36, 8, 27, 20], and signature schemes [16, 35, 3, 21, 14, 43, 34, 6]. However, somewhat surprisingly, only few SPS schemes with tight security reductions are known. Moreover, these tightly secure SPS schemes [35, 6] 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.

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 [45]	$(n_1, 0)$	$(10, 1)$	$(16, 2n_1 + 5)$	$\mathbf{O}(Q)$	SXDH, XDLINX
KPW15 [40]	$(n_1, 0)$	$(6, 1)$	$(0, n_1 + 6)$	$2Q^2$	SXDH
JR17 [38]	$(n_1, 0)$	$(5, 1)$	$(0, n_1 + 6)$	$Q \log Q$	SXDH
AHNOP17 [6]	$(n_1, 0)$	$(13, 12)$	$(18, n_1 + 11)$	80λ	SXDH
JOR18 [37]	$(n_1, 0)$	$(11, 6)$	$(7, n_1 + 16)$	116λ	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 [40]	(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
JOR18 [37]	(n_1, n_2)	$(12, 8)$	$(n_2 + 8, n_1 + 17)$	116λ	SXDH
Ours (bilateral)	(n_1, n_2)	$(9, 8)$	$(n_2 + 4, n_1 + 9)$	$6 \log Q$	SXDH

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.

[3] 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.

[4] 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.

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 [42]). 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.[5] 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 $\mathbf{t} \in \operatorname{span}(\mathbf{A}_0) \cup \operatorname{span}(\mathbf{A}_1)$ for another fixed matrix $\mathbf{A}_1$ (also given as $[\mathbf{A}_1]$ in the public key). The original Kurosawa-Desmedt scheme [42] is identical, except that π is omitted, and $k = 1$. Hence, the main benefit of π is that it enables a tight security reduction.[6]

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

[5] 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].

[6] 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.

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 $\mathbb{Z}_p$).[7]

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 [33, 26] 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 [33, 26, 6]) 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.[8]

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 [39, 41]), 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 [40] for constructing SPS schemes, altough with a very 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 [40], 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.

[7] 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.

[8] 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.

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 [40] that yields a bilateral scheme with signatures of 17 group elements (for $k = 1$).

Scheme	$\lvert M\rvert$	**PPEs**	**Pairings**	**Pairings**	**Sec.**	$\lvert\mathbb{G}_1\rvert$	$\lvert\sigma\rvert$
			(plain)	(batched)	**loss**	(bits)	(bits)
KPW [40]	$(n_1, 0)$	3	$n_1 + 11$	$n_1 + 10$	$2Q^2$	322	2576
JR [38]	$(n_1, 0)$	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 (uniliteral)	$(n_1, 0)$	6	$n_1 + 29$	$n_1 + 11$	$6 \log Q$	216	4320
KPW [40]	(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 (biliteral)	(n_1, n_2)	7	$n_1 + n_2 + 33$	$n_1 + n_2 + 15$	$6 \log Q$	216	5400

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 and bilateral 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 "$\lvert M\rvert$" and "Sec. loss" columns have the same meaning as in Table 1. The column "$\lvert\mathbb{G}_1\rvert$" 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 Section 1.2. "$\lvert\sigma\rvert$ (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.

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.

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 [33, 26]), 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 [38] and Kiltz, Pan, and Wee [40]. 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, 28, 23, 9]. 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 [38], even for small messages. For larger messages, our scheme becomes preferable. In the bilateral setting, our scheme is clearly the most efficient known scheme.

Independent work. We briefly note that the recent work of Jutla, Ohkubo, and Roy [37] improved the efficiency of [6] by using more efficient NIZK proof systems. Their method to achieve tight security is via an encrypted partition bit in the signature, which is similar to the work of [6]. In contrast to this, we transform a simple tightly secure MAC to an efficient SPS. Amongst other things, we achieve shorter signatures and public keys and smaller security loss (cf. Table 1).

Roadmap

We fix some notation and recall some preliminaries in Section 2. In Section 3, we present our basic MAC and prove it secure (using the mentioned core lemma). In Section 4 and Section 5, we present our signature and SPS schemes.

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 out $\leftarrow \mathcal{B}(\text{in})$ by out we denote the output of $\mathcal{B}$ on input in. For a deterministic algorithm we sometimes use the notation out $:= \mathcal{B}(\text{in})$ 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

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

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 $1 - \varDelta_{\mathcal{D}_{2k,k}}$. 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 $\mathcal{A}$,*

$$\mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG}, \mathbb{G}_i, \mathcal{D}_{\ell,k}, \mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(\mathcal{PG}, [\mathbf{A}]_i, [\mathbf{Aw}]_i) = 1]$$

$$- \Pr[\mathcal{A}(\mathcal{PG}, [\mathbf{A}]_i, [\mathbf{u}]_i) = 1]| \leq \mathsf{negl}(\lambda),$$

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 ℓ, 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

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

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 [46] 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 $\mathcal{A}$,*

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

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 [46].

Lemma 2 ([46]). *For any matrix distribution $\mathcal{D}_k$, $\mathcal{D}_k$-MDDH $\Rightarrow$ $\mathcal{D}_k$-KMDH.*

2.3 Signature schemes 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:*

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

$\textsc{TagO}(m)$:
$\mathcal{Q}_{\mathsf{tag}} := \mathcal{Q}_{\mathsf{tag}} \cup \{m\}$
$\mathsf{tag} \leftarrow \mathsf{Tag}(pp, sk, m)$
return $\mathsf{tag}$

$\textsc{VerO}(m, \mathsf{tag})$:
$b \leftarrow \mathsf{Ver}(pp, sk, m, \mathsf{tag})$
return $b$
```

The adversary is restricted to one call to $\textsc{VerO}$*. We say that a MAC scheme* MAC *is* $\mathsf{UF\text{-}CMA}$ 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 $\mathsf{UF\text{-}CMA}$ 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:*

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

```
$\textsc{SignO}(m)$:
$\mathcal{Q}_{\mathsf{sign}} := \mathcal{Q}_{\mathsf{sign}} \cup \{m\}$
$\sigma \leftarrow \mathsf{Sign}(pk, sk, m)$
$\textbf{\textit{return}}\ \sigma$
```

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,* $x \in \mathcal{L}$ *and* Π *outputs a verdict* $b \in \{0, 1\}$*.*
$\mathsf{PSim}(crs, td, x)$ *given a crs with corresponding trapdoor td and a word* $x \in \mathcal{L}$*, outputs a proof* Π*.*

Further we require the following properties to hold.

Completeness: *For all possible public parameters pars, all* $\lambda \in \mathbb{N}$*, 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 $crs \leftarrow \mathsf{PGen}\ (1^\lambda, pars)$ *and* $\Pi \leftarrow \mathsf{PPrv}(crs, x, w)$*.*

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

$$\mathrm{Adv}^{\text{keygen}}_{\mathsf{PS},\mathcal{A}}(\lambda) := \Big| \Pr[\mathcal{A}(1^\lambda, crs) = 1 \mid crs \leftarrow \mathsf{PGen}(1^\lambda, pars)] - \Pr[\mathcal{A}(1^\lambda, crs) = 1 \mid (crs, td) \leftarrow \mathsf{PTGen}(1^\lambda, pars)] \Big| \leq \mathsf{negl}(\lambda).$$

Further, for all $x \in \mathcal{L}$ *with witness* w *such that* $\mathcal{R}_{\mathcal{L}}(x, w) = 1$*, the following are identically distributed:*

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

where $(crs, td) \leftarrow_R \mathsf{PTGen}(1^\lambda)$*.*

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* $\mathsf{PVer}(crs, x, \Pi) = 0$*.*

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

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

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

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

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

Fig. 1: NIZK argument for $\mathcal{L}^{\vee}_{\mathbf{A}_0,\mathbf{A}_1}$ ([31, 47]).

2.5 NIZK for our OR-language

In this section we recall an instantiation of a NIZK for an OR-language first given in [31] and later generalized in [47] for more general languages. 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

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

We consider $k \in \mathbb{N}$ to be chosen ahead of time, fixed and implicitly known to all algorithms (recall that in practice, $k = 1$ for SXDH, $k = 2$ for DLIN).

OR-proof ([31, 47]). In Figure 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, 47]. For the sake of completeness we recall the proof here.

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 Figure 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*

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

Proof. **Completeness:** Let $j \in \{0,1\}$ such that $[\mathbf{x}]_1 = [\mathbf{A}_j]_1 \cdot \mathbf{r}$. Let $([\mathbf{z}_0]_2, ([\mathbf{C}_i]_2, [\mathit{\Pi}_i]_1)_{i\in\{0,1\}})$ be returned by PPrv on input *crs*, $[\mathbf{x}]_1$ and $\mathbf{r}$.

$$\begin{aligned} e([\mathbf{A}_j]_1, [\mathbf{C}_j]_2) &= e([\mathbf{A}_j]_1, \mathbf{S}_j \cdot [\mathbf{D}]_2^\top + \mathbf{r} \cdot [\mathbf{z}_j]_2^\top) = [\mathbf{A}_j \cdot \mathbf{S}_j \cdot \mathbf{D}^\top]_T + [\mathbf{A}_j \cdot \mathbf{r} \cdot \mathbf{z}_j^\top]_T \\ &= [\mathit{\Pi}_j \cdot \mathbf{D}^\top]_T + [\mathbf{x} \cdot \mathbf{z}_j^\top]_T = e([\mathit{\Pi}_j]_1, [\mathbf{D}]_2^\top) + e([\mathbf{x}]_1, [\mathbf{z}_j]_2^\top) \end{aligned}$$

and further

$$\begin{aligned} e([\mathbf{A}_{1-j}]_1, [\mathbf{C}_{1-j}]_2) &= e([\mathbf{A}_{1-j}]_1, \mathbf{S}_{1-j} \cdot [\mathbf{D}]_2^\top) = [\mathbf{A}_{1-j} \cdot \mathbf{S}_{1-j} \cdot \mathbf{D}^\top]_T \\ &= [(\mathbf{A}_{1-j} \cdot \mathbf{S}_{1-j} - \mathbf{x} \cdot \mathbf{v}^\top + \mathbf{x} \cdot \mathbf{v}^\top) \cdot \mathbf{D}^\top]_T = [\mathit{\Pi}_{1-j} \cdot \mathbf{D}^\top]_T + [\mathbf{x} \cdot \mathbf{z}_{1-j}^\top]_T \\ &= e([\mathit{\Pi}_{1-j}]_1, [\mathbf{D}]_2^\top) + e([\mathbf{x}]_1, [\mathbf{z}_{1-j}]_2^\top). \end{aligned}$$

Composable zero-knowledge: Let $\mathcal{A}$ be a PPT adversary, attacking the zero-knowledge property. We build a PPT adversary $\mathcal{B}$ such that $T(\mathcal{B}) \approx T(\mathcal{A})$ and

$$\mathrm{Adv}^{\mathrm{zk}}_{\mathsf{PS},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_2,\mathcal{D}_k,\mathcal{B}}(\lambda) + \tfrac{1}{p}.$$

Upon receiving its MDDH challenge $(\mathcal{PG}, [\mathbf{D}]_2, [\mathbf{z}]_2)$, $\mathcal{B}$ samples $\mathbf{A}_0, \mathbf{A}_1 \leftarrow_R \mathcal{D}_{2k,k}$ and forwards the common reference string $\mathit{crs} := ((\mathcal{PG}, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1), [\mathbf{D}]_2, [\mathbf{z}]_2)$ to $\mathcal{A}$. When $\mathcal{B}$ receives a real MDDH tuple, that is, when there exists $\mathbf{u} \in \mathbb{Z}_p^k$ such that $[\mathbf{z}]_2 := [\mathbf{Du}]_2$, $\mathcal{B}$ simulates a *crs* as output by $\mathsf{PTGen}(1^\lambda, \mathit{pars})$. The other case is when $\mathcal{B}$ receives $[\mathbf{z}]_2 \leftarrow_R \mathbb{G}_2^{k+1}$. In that case, using the fact that the uniformly random distribution over $\mathbb{Z}_p^{k+1}$ and the uniformly random distribution over $\mathbb{Z}_p^{k+1} \setminus \mathrm{span}(\mathbf{D})$ are $1/p$-statistically close distributions, since $\mathbf{D}$ is of rank k, we can conclude that $\mathcal{B}$ simulates the *crs* as output by $\mathsf{PGen}(1^\lambda, \mathit{pars})$, within a $1/p$ statistical distance. Overall, we get: $\mathrm{Adv}^{\mathrm{zk}}_{\mathsf{PS},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_2,\mathcal{D}_k,\mathcal{B}}(\lambda) + \frac{1}{p}$.

Now, we proceed to prove that for all $[\mathbf{x}]_1 \in \mathcal{L}^{\vee}_{\mathbf{A}_0,\mathbf{A}_1}$ with witness $\mathbf{r} \in \mathbb{Z}_p^k$, $\{\mathsf{PPrv}(\mathit{crs}, [\mathbf{x}]_1, \mathbf{r}), (\mathit{crs}, \mathit{td}) \leftarrow \mathsf{PTGen}(1^\lambda, \mathit{pars})\}$ is identically distributed to $\{\mathsf{PSim}(\mathit{crs}, \mathit{td}, [\mathbf{x}]_1), (\mathit{crs}, \mathit{td}) \leftarrow \mathsf{PTGen}(1^\lambda, \mathit{pars})\}$, which concludes the proof.

First, note that PPrv and PSim compute the vectors $[\mathbf{z}_0]_2$ and $[\mathbf{z}_1]_2$ in the exact same way, i.e. for all $j \in \{0,1\}$, $\mathbf{z}_j := \mathbf{D}\mathbf{v}_j$ where $\mathbf{v}_0, \mathbf{v}_1$ are uniformly random over $\mathbb{Z}_p^k$ subject to $\mathbf{v}_0 + \mathbf{v}_1 = \mathbf{u}$ (recall $\mathbf{z} := \mathbf{Du}$). Second, on input $[\mathbf{x}]_1 := [\mathbf{A}_j\mathbf{r}]_1$, for some $j \in \{0,1\}$, $\mathsf{PPrv}(\mathit{crs}, [\mathbf{x}]_1, \mathbf{r})$ computes $[\mathbf{C}_{1-j}]_2$ and $[\mathit{\Pi}_{1-j}]_1$ exactly as PSim, that is: $[\mathbf{C}_{1-j}]_2 = \mathbf{S}_{1-j}[\mathbf{D}^\top]_2$ and $[\mathit{\Pi}_{1-j}]_1 = [\mathbf{A}_{1-j}]_1\mathbf{S}_{1-j} - [\mathbf{x}]_1 \cdot \mathbf{v}_{1-j}^\top$. The algorithm $\mathsf{PPrv}(\mathit{crs}, [\mathbf{x}]_1, \mathbf{r})$ additionally computes $[\mathbf{C}_j]_2 = \mathbf{S}_j[\mathbf{D}^\top]_2 + \mathbf{r} \cdot [\mathbf{z}_j^\top]_2$ and $[\mathit{\Pi}_j]_1 = [\mathbf{A}_j]_1\mathbf{S}_j$, with $\mathbf{S}_j \leftarrow_R \mathbb{Z}_p^{k\times k}$. Since the following are identically distributed:

$$\mathbf{S}_j \ \text{ and } \ \mathbf{S}_j - \mathbf{r} \cdot \mathbf{v}_j^\top,$$

for $\mathbf{S}_j \leftarrow_R \mathbb{Z}_p^{k\times k}$, we can re-write the commitment and proof computed by $\mathsf{PPrv}(crs, [\mathbf{x}]_1, \mathbf{r})$ as $[\mathbf{C}_j]_2 = \mathbf{S}_j[\mathbf{D}^\top]_2 - \mathbf{r}\cdot\mathbf{v}_j^\top[\mathbf{D}^\top]_2 + \mathbf{r}\cdot[\mathbf{z}_j^\top]_2 = [\mathbf{S}_j\mathbf{D}^\top]_2$ and $[\Pi_j]_1 = [\mathbf{A}_j]_1\mathbf{S}_j - [\mathbf{A}_j\mathbf{r}\cdot\mathbf{v}_j^\top\mathbf{D}^\top]_2 = [\mathbf{A}_j\mathbf{S}_j]_1 - [\mathbf{x}\cdot\mathbf{z}_j^\top]_2$, which is exactly as the output of PSim.

Perfect soundness: Since $\mathbf{z} = \mathbf{z}_0 + \mathbf{z}_1 \notin \mathrm{span}(\mathbf{D})$, there is a $j \in \{0,1\}$ such that $\mathbf{z}_j \notin \mathrm{span}(\mathbf{D})$. This implies that there exists a $\mathbf{d}^\perp \in \mathbb{Z}_p^{k+1}$ such that $\mathbf{D}^\top\mathbf{d}^\perp = \mathbf{0}$, and $\mathbf{z}_j^\top\mathbf{d}^\perp = 1$. Furthermore, as the row vectors of $\mathbf{D}$ together with $\mathbf{z}_j$ form a basis of $\mathbb{Z}_p^{k+1}$, we can write $[\mathbf{C}_j]_2 := [\mathbf{S}_j\cdot\mathbf{D}^\top + \mathbf{r}\cdot\mathbf{z}_j^\top]_2$ for some $\mathbf{S}_j \in \mathbb{Z}_p^{k\times k}, \mathbf{r} \in \mathbb{Z}_p^k$. Multiplying the verification equation by $\mathbf{d}^\perp$ thus yields $[\mathbf{A}_j\mathbf{r}]_1 = [\mathbf{x}]_1$, which proves a successful forgery outside $\mathcal{L}^\vee_{\mathbf{A}_0,\mathbf{A}_1}$ impossible.

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

```
$\mathsf{Tag}(pp, sk, \mu \in \mathbb{Z}_p)$:
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 $\mathsf{tag}$

$\mathsf{Ver}(pp, sk, \mu \in \mathbb{Z}_p, \mathsf{tag})$ :
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.

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 Section 2.5. In Figure 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.

Instead of directly proving $\mathsf{UF\text{-}CMA}$ 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 Figure 3.

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*

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

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

$\textsc{VerO}(\mathsf{tag})$ :
parse $\mathsf{tag} = ([\mathbf{t}]_1, \mathit{\Pi}, [u']_1)$
$b \leftarrow \mathsf{PVer}(crs, [\mathbf{t}]_1, \mathit{\Pi})$
if $b = 1$ and $\exists \mathsf{ctr}' \le \mathsf{ctr}$ :
  $[u']_1 = (\mathbf{k}_0 + \beta \cdot \mathbf{F}(\mathsf{ctr}'))^\top [\mathbf{t}]_1$
    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.

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*

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

where

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

Recall that by definition of the distribution $\mathcal{D}_{2k,k}$ *(Section 2.2), the term* $\mathit{\Delta}_{\mathcal{D}_{2k,k}}$ *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 proof of Lemma 5),and perform the following modifications:

- **Partitioning.** First, we choose $\mathbf{t} \in \operatorname{span}(\mathbf{A}_{\mathsf{ctr}_{i+1}})$ 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 $\mathbf{t} \in \operatorname{span}(\mathbf{A}_0) \cup \operatorname{span}(\mathbf{A}_1)$ 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 $\mathbf{t} \in \operatorname{span}(A_b)$ 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 Figure 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 $\mathcal{A}$.

```
$\mathsf{G}_0$, $\boxed{\mathsf{G}_1, \boxed{\mathsf{G}_2, \colorbox{lightgray}{$\mathsf{G}_{3.i}$}}}$:
$\mathsf{ctr} := 0$
$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$
$\mathbf{A}_0, \mathbf{A}_1 \leftarrow_R \mathcal{D}_{2k,k}$
$pars := (\mathcal{PG}, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1)$
$crs \leftarrow \mathsf{PGen}(1^\lambda, pars)$
$\boxed{(crs, td) \leftarrow \mathsf{PTGen}(1^\lambda, pars)}$
$\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$
$pp := (\mathcal{PG}, [\mathbf{A}_0]_1, crs)$
$\mathsf{tag} \leftarrow \mathcal{A}^{\text{TAGO}()}(pp)$
return VERO($\mathsf{tag}$)

TAGO():
$\mathsf{ctr} := \mathsf{ctr} + 1$
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{t}]_1 := [\mathbf{A}_0]_1 \mathbf{r}$
$\boxed{[\mathbf{t}]_1 \leftarrow_R \mathbb{G}_1^{2k}}$
$\Pi \leftarrow \mathsf{PPrv}(crs, [\mathbf{t}]_1, \mathbf{r})$
$\boxed{\Pi \leftarrow \mathsf{PSim}(crs, td, [\mathbf{t}]_1)}$
$[u']_1 := (\mathbf{k}_0 \colorbox{lightgray}{$+\mathbf{F}_i(\mathsf{ctr}_{|i})$})^\top [\mathbf{t}]_1$
return $\mathsf{tag} := ([\mathbf{t}]_1, \Pi, [u']_1)$

VERO(tag):
parse $\mathsf{tag} =: ([\mathbf{t}]_1, \Pi, [u']_1)$
$b \leftarrow \mathsf{PVer}(crs, [\mathbf{t}]_1, \Pi)$
if $b = 1$ and $\exists \mathsf{ctr}' \leq \mathsf{ctr}$ :
    $[u']_1 = (\mathbf{k}_0 + \colorbox{lightgray}{$\mathbf{F}_i(\mathsf{ctr}'_{|i})$})^\top [\mathbf{t}]_1$
        return 1
else return 0
```

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.

$\mathsf{G_0}$**:** 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 instead of PGen. Because the output of PSim and PPrv are identically distributed on a $crs \leftarrow_R \mathsf{PTGen}$ (see Definition 8), we can argue that the *crs* distribution is the only difference in these two games. This difference is justified by the zero-knowledge of PS. Namely, we build an adversary $\mathcal{B}$ on the composable zero-knowledge property of PS as follows. The adversary $\mathcal{B}$ obtains *crs* from its own experiment instead of calling PGen, samples $\mathbf{A}_0 \leftarrow_R \mathcal{D}_{2k,k}$, and forwards $pars := (\mathcal{PG}, [\mathbf{A}_0]_1, crs)$ to $\mathcal{A}$. Then $\mathcal{B}$ samples $\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$, thanks to which it can answer TagO and VerO queries. Note that $\mathcal{B}$ simulates G_0 in case it was given a *crs* generated by PGen, whereas it simulates G_1 in case it was given a *crs* generated by PTGen. Thus, $\mathcal{B}$ is 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 $\mathit{\Pi}$ 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

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

$\mathsf{G}_{\mathbf{3}.\boldsymbol{i}} \rightsquigarrow \mathsf{G}_{\mathbf{3}.(\boldsymbol{i}+\mathbf{1})}$**:** For the proof of this transition we refer to Lemma 5, which states that for every adversary $\mathcal{A}$ there exist adversaries $\mathcal{B}_i$, $\mathcal{B}'_i$ with running time $T(\mathcal{B}_i) \approx T(\mathcal{B}'_i) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$, and

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

$\mathsf{G}_{\mathbf{3}.\lceil\log Q\rceil} \rightsquigarrow \mathbf{Exp}^{\mathbf{core}}_{\mathbf{1},\boldsymbol{\mathcal{A}}}(\boldsymbol{\lambda})$**:** 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 $\operatorname{span}([\mathbf{A}_0]_1)$ again, we can switch back to honest generation of the common reference string *crs*. 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

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

Lemma 5 **($\mathsf{G}_{3.i} \rightsquigarrow \mathsf{G}_{3.(i+1)}$).** *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 for all $i \in \{0, \ldots, \lceil \log Q \rceil - 1\}$ between $\mathsf{G}_{3.i}$ and $\mathsf{G}_{3.(i+1)}$ as defined in Figure 7 the winning chances of an adversary can only increase (up to negligible terms). More precisely, for every adversary $\mathcal{A}$ there exist adversaries $\mathcal{B}_i$, $\mathcal{B}'_i$ with running times $T(\mathcal{B}_i) \approx T(\mathcal{B}'_i) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$, and*

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

Proof. We proceed via a series of hybrid games $\mathsf{H}_{i.j}$ for $i \in \{0, \ldots, \lceil \log Q \rceil - 1\}$, $j \in \{1, \ldots, 8\}$, described in Figure 5, and we denote by $\widehat{\varepsilon}_{i.j}$ the advantage of $\mathcal{A}$ to win $\mathsf{H}_{i.j}$. We give an overview of the transitions in Figure 6.

$\mathsf{G}_{3.i} \rightsquigarrow \mathsf{H}_{i.1}$: We switch $[\mathbf{t}]_1$ from chosen uniformly at random by TAGO to $[\mathbf{A}_{\mathsf{ctr}_{i+1}}\mathbf{r}]_1$ for $\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$, where ctr_{i+1} is the $i+1$'st bit of the binary representation of ctr, using the $\mathcal{D}_{2k,k}$-MDDH assumption twice. More precisely, we introduce an intermediary game $\mathsf{H}_{i.0}$, where we choose $[\mathbf{t}_i]_1$ as

$$[\mathbf{t}_i]_1 = \begin{cases} [\mathbf{A}_0\mathbf{r}_i] \text{ for } \mathbf{r}_i \leftarrow_R \mathbb{Z}_p^k & \text{if } \mathsf{ctr}_{i+1} = 0 \\ [\mathbf{u}_i]_1 \text{ for } \mathbf{u}_i \leftarrow_R \mathbb{Z}_p^{2k} & \text{else} \end{cases}.$$

```
$\mathsf{H}_{i.1}$ $\boxed{\mathsf{H}_{i.2}, \mathsf{H}_{i.3}, \mathsf{H}_{i.4}-\mathsf{H}_{i.6}, \mathsf{H}_{i.7}}$, $\mathsf{H}_{i.8}$:
$\mathsf{ctr} := 0$
$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$
$\mathbf{A}_0, \mathbf{A}_1 \leftarrow_R \mathcal{D}_{2k,k}$
$(crs, td) \leftarrow \mathsf{PTGen}(1^\lambda, pars)$
$\boxed{crs \leftarrow \mathsf{PGen}(1^\lambda, pars)}$
$\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$
$pp := (\mathcal{PG}, [\mathbf{A}_0]_1, crs)$
$\mathsf{tag} \leftarrow \mathcal{A}^{\textsc{TagO}()}(pp)$
return $\textsc{VerO}(\mathsf{tag})$
```

```
$\textsc{TagO}()$:
$\mathsf{ctr} := \mathsf{ctr} + 1$
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{t}]_1 := [\mathbf{A}_{\mathsf{ctr}_{i+1}}]_1 \mathbf{r}$
$\mathit{\Pi} \leftarrow \mathsf{PSim}(crs, td, [\mathbf{t}]_1)$
$\boxed{\mathit{\Pi} \leftarrow \mathsf{PPrv}(crs, [\mathbf{t}]_1, \mathbf{r})}$
$[u']_1 := [(\mathbf{k}_0 + \mathbf{F}_i(\mathsf{ctr}_{|i}))^\top \mathbf{t}]_1$
$[u']_1 := [(\mathbf{k}_0 + \mathbf{F}_{i+1}(\mathsf{ctr}_{|i+1}))^\top \mathbf{t}]_1$
$\mathsf{tag} := ([\mathbf{t}]_1, \mathit{\Pi}, [u']_1)$
return $\mathsf{tag}$
```

```
$\textsc{VerO}(\mathsf{tag})$ :
parse $\mathsf{tag} =: ([\mathbf{t}]_1, \mathit{\Pi}, [u']_1)$
$b \leftarrow \mathsf{PVer}(crs, [\mathbf{t}]_1, \mathit{\Pi})$
$\mathcal{S} := \{\mathbf{F}_i(\mathsf{ctr}'_{|i}) : \mathsf{ctr}' \leq \mathsf{ctr}\}$
Game $\mathsf{H}_{i.4}$:
$\mathcal{S} := \{ \mathbf{F}_{i+1}(\mathsf{ctr}'_{|i}|d_{[\mathbf{t}]}) : \mathsf{ctr}' \leq \mathsf{ctr}\}$
Game $\mathsf{H}_{i.5}$:
$\mathcal{S} := \{\mathbf{F}_{i+1}(\mathsf{ctr}'_{|i}|b) : \mathsf{ctr}' \leq \mathsf{ctr}, b \in \{0,1\} \}$
Game $\mathsf{H}_{i.6} - \mathsf{H}_{i.8}$:
$\mathcal{S} := \{\mathbf{F}_{i+1}(\mathsf{ctr}'_{|i+1}) : \mathsf{ctr}' \leq \mathsf{ctr}\}$
if $[\mathbf{t}]_1 \in \mathrm{span}([\mathbf{A}_0]) \cup \mathrm{span}([\mathbf{A}_1])$
  and $b = 1$ and $\exists \mathbf{w} \in \mathcal{S} : [u']_1 = (\mathbf{k}_0 + \mathbf{w})^\top [\mathbf{t}]_1$
    return 1
else return 0
```

Fig. 5: Games $\mathsf{H}_{i.j}$ for $i \in \{0, \ldots, \lceil \log Q \rceil - 1\}$, $j \in \{1, \ldots, 8\}$, for the proof of Lemma 5. Here, $\mathbf{F}_i : \{0,1\}^i \rightarrow \mathbb{Z}_p^{2k}$ denotes a random function, $\mathsf{ctr}_{|i}$ denotes the i-bit string that is a prefix of ctr written in binary, and ctr_i is the i'th bit of ctr written in binary. We have $d_{[\mathbf{t}]} = 0$ if $\mathbf{t} \in \mathrm{span}(\mathbf{A}_0)$, and $d_{[\mathbf{t}]} = 1$ if $\mathbf{t} \in \mathrm{span}(\mathbf{A}_1)$. In each procedure, the components inside a solid (dotted, gray) frame are only present in the games marked by a solid (dotted, gray) frame. For the intermediate transitions from game $\mathsf{H}_{i.4}$ to game $\mathsf{H}_{i.6}$ we use dark gray highlighting to emphasize the respective differences.

$\#$	$crs \leftarrow$	$[\mathbf{t}]_1$ in TagO	$\mathit{\Pi} \leftarrow$ in TagO	$[u']_1 = (\mathbf{k}_0 + \cdot)^\top [\mathbf{t}]_1$ in TagO	$\mathcal{S} := \{\cdot : \mathsf{ctr}' \in \mathcal{Q}_{\mathsf{tag}}\}$ in VerO	check on $[\mathbf{t}]_1$ in VerO	**game knows**	**remark**
$\mathsf{G}_{3.i}$	PTGen	$\leftarrow_R \mathbb{G}_1^{2k}$	PSim	$\mathbf{F}_i(\mathsf{ctr}_{\vert i})$	$\mathbf{F}_i(\mathsf{ctr}'_{\vert i})$	-	-	Game $\mathsf{G}_{3.i}$
$\mathsf{H}_{i.1}$	PTGen	$= [\mathbf{A}_{\mathsf{ctr}_{i+1}}]_1\mathbf{r}$	PSim	$\mathbf{F}_i(\mathsf{ctr}_{\vert i})$	$\mathbf{F}_i(\mathsf{ctr}'_{\vert i})$	-	-	$\mathcal{D}_{2k,k}$-MDDH
$\mathsf{H}_{i.2}$	PGen	$= [\mathbf{A}_{\mathsf{ctr}_{i+1}}]_1\mathbf{r}$	PPrv	$\mathbf{F}_i(\mathsf{ctr}_{\vert i})$	$\mathbf{F}_i(\mathsf{ctr}'_{\vert i})$	-	-	ZK of PS
$\mathsf{H}_{i.3}$	PGen	$= [\mathbf{A}_{\mathsf{ctr}_{i+1}}]_1\mathbf{r}$	PPrv	$\mathbf{F}_i(\mathsf{ctr}_{\vert i})$	$\mathbf{F}_i(\mathsf{ctr}'_{\vert i})$	$[\mathbf{t}]_1 \overset{?}{\in} \mathcal{L}^{\vee}_{\mathbf{A}_0,\mathbf{A}_1}$	$\mathbf{A}_0,\ \mathbf{A}_1$	SND of PS
$\mathsf{H}_{i.4}$	PGen	$= [\mathbf{A}_{\mathsf{ctr}_{i+1}}]_1\mathbf{r}$	PPrv	$\mathbf{F}_{i+1}(\mathsf{ctr}_{\vert i+1})$	$\mathbf{F}_{i+1}(\mathsf{ctr}'_{\vert i} \vert d_{[\mathbf{t}]})$	$[\mathbf{t}]_1 \overset{?}{\in} \mathcal{L}^{\vee}_{\mathbf{A}_0,\mathbf{A}_1}$	$\mathbf{A}_0,\ \mathbf{A}_1$	statistical
$\mathsf{H}_{i.5}$	PGen	$= [\mathbf{A}_{\mathsf{ctr}_{i+1}}]_1\mathbf{r}$	PPrv	$\mathbf{F}_{i+1}(\mathsf{ctr}_{\vert i+1})$	$\mathbf{F}_{i+1}(\mathsf{ctr}'_{\vert i} \vert b), b \in \{0,1\}$	$[\mathbf{t}]_1 \overset{?}{\in} \mathcal{L}^{\vee}_{\mathbf{A}_0,\mathbf{A}_1}$	$\mathbf{A}_0,\ \mathbf{A}_1$	incr. chances
$\mathsf{H}_{i.6}$	PGen	$= [\mathbf{A}_{\mathsf{ctr}_{i+1}}]_1\mathbf{r}$	PPrv	$\mathbf{F}_{i+1}(\mathsf{ctr}_{\vert i+1})$	$\mathbf{F}_{i+1}(\mathsf{ctr}'_{\vert i+1})$	$[\mathbf{t}]_1 \overset{?}{\in} \mathcal{L}^{\vee}_{\mathbf{A}_0,\mathbf{A}_1}$	$\mathbf{A}_0,\ \mathbf{A}_1$	statistical
$\mathsf{H}_{i.7}$	PGen	$= [\mathbf{A}_{\mathsf{ctr}_{i+1}}]_1\mathbf{r}$	PPrv	$\mathbf{F}_{i+1}(\mathsf{ctr}_{\vert i+1})$	$\mathbf{F}_{i+1}(\mathsf{ctr}'_{\vert i+1})$	-	-	SND of PS
$\mathsf{H}_{i.8}$	PTGen	$= [\mathbf{A}_{\mathsf{ctr}_{i+1}}]_1\mathbf{r}$	PSim	$\mathbf{F}_{i+1}(\mathsf{ctr}_{\vert i+1})$	$\mathbf{F}_{i+1}(\mathsf{ctr}'_{\vert i+1})$	-	-	ZK of PS
$\mathsf{G}_{3.(i+1)}$	PTGen	$\leftarrow_R \mathbb{G}_1^{2k}$	PSim	$\mathbf{F}_{i+1}(\mathsf{ctr}_{\vert i+1})$	$\mathbf{F}_{i+1}(\mathsf{ctr}'_{\vert i+1})$	-	-	$\mathcal{D}_{2k,k}$-MDDH

Fig. 6: Overview of the transitions in the proof of Lemma 5. We highlight the respective changes between the games in gray. In the third column, $\mathbf{r}$ is chosen at random from $\mathbb{Z}_p^k$ and ctr_{i+1} denotes the $i+1$'st bit of the bit representation of $\mathsf{ctr} \in \mathbb{Z}_p$. In the fifth and sixth column, the dot $\cdot$ represents the gap filled by the respective entries in the table. Further, $\mathbf{F}_i \colon \{0,1\}^i \to \mathbb{Z}_p^{2k}$, $\mathbf{F}_{i+1} \colon \{0,1\}^i \to \mathbb{Z}_p^{2k}$ are random functions and $\mathsf{ctr}_{\vert i}$ and $\mathsf{ctr}_{\vert i+1}$ denote the bit strings consisting of the first i respectively the first $i+1$ bits of the bit representation of $\mathsf{ctr} \in \mathbb{N}$. Further, we have $d_{[\mathbf{t}]} = 0$ if $\mathbf{t} \in \operatorname{span}(\mathbf{A}_0)$, and $d_{[\mathbf{t}]} = 1$ if $\mathbf{t} \in \operatorname{span}(\mathbf{A}_1)$. For the seventh column, recall that $\mathcal{L}^{\vee}_{\mathbf{A}_0,\mathbf{A}_1} := \operatorname{span}([\mathbf{A}_0]_1) \cup \operatorname{span}([\mathbf{A}_1]_1)$. In the remark we give the justification for the respective transition from the previous game to the current game.

Let $\mathcal{A}$ be an adversary distinguishing between $\mathsf{G}_{3.i}$ and $\mathsf{H}_{i.0}$ and let $\mathcal{B}$ be an adversary receiving a Q-fold MDDH-challenge $(\mathcal{PG}, [\mathbf{A}_0]_1, [\mathbf{z}_1]_1, \ldots, [\mathbf{z}_Q]_1)$ as input. Then $\mathcal{B}$ sets up the game for $\mathcal{A}$ similar to game $\mathsf{G}_{3.i}$, where he embeds $[\mathbf{A}_0]_1$ into the public parameters *pars*. Further, whenever obtaining a simulation query ctr with $\mathsf{ctr}_{|i+1} = 0$, $\mathcal{B}$ sets $[\mathbf{t}_i] := [\mathbf{z}_i]_1$ and otherwise follows $\mathsf{G}_{3.i}$. Similar, we can reduce the transition from game $\mathsf{H}_{i.0}$ to $\mathsf{H}_{i.1}$ to the $\mathcal{D}_{2k,k}$-MDDH assumption. Applying Lemma 1 yields an adversary $\mathcal{B}_{i.0}$ with $T(\mathcal{B}_{i.0}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ such that:

$$|\varepsilon_{3.i} - \widehat{\varepsilon}_{i.1}| \leq 2k \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_1,\mathcal{D}_{2k,k},\mathcal{B}_{i.0}}(\lambda) + \tfrac{2}{p-1}.$$

$\mathsf{H}_{i.1} \rightsquigarrow \mathsf{H}_{i.2}$**:** In this step we reverse the transition from game G_0 to G_1 in Theorem 1. Namely, we generate *crs* using PGen instead of PTGen, and we use the fact that proofs generated by PSim or PPrv are identically distributed when $crs \leftarrow_R \mathsf{PTGen}(1^\lambda, pars)$. Note that it is possible to use the algorithm PPrv, as from game $\mathsf{H}_{i.1}$ on, we choose all $[\mathbf{t}]_1$ in tag queries from $\mathcal{L}$ with corresponding witness and can thus honestly generate proofs. Therefore, by the same reasoning as for $\mathsf{G}_0 \rightsquigarrow \mathsf{G}_1$ there exists an adversary $\mathcal{B}_{i.1}$ such that $T(\mathcal{B}_{i.1}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ and

$$|\widehat{\varepsilon}_{i.1} - \widehat{\varepsilon}_{i.2}| \leq \mathrm{Adv}^{\mathrm{ZK}}_{\mathsf{PS},\mathcal{B}_{i.1}}(\lambda).$$

$\mathsf{H}_{i.2} \rightsquigarrow \mathsf{H}_{i.3}$**:** From game $\mathsf{H}_{i.3}$ on we introduce an additionally check in the verification oracle. Namely, $\textsc{VerO}$ checks that $[\mathbf{t}]_1 \in \mathrm{span}([\mathbf{A}_0]_1) \cup \mathrm{span}([\mathbf{A}_1]_1)$. As the *crs* is generated by PGen, we can employ the perfect soundness of PS to obtain

$$\widehat{\varepsilon}_{i.2} = \widehat{\varepsilon}_{i.3}.$$

$\mathsf{H}_{i.3} \rightsquigarrow \mathsf{H}_{i.4}$**:** Let $\mathbf{A}_0^\perp \in \mathsf{orth}(\mathbf{A}_0)$ and $\mathbf{A}_1^\perp \in \mathsf{orth}(\mathbf{A}_1)$. We introduce an intermediary game $\mathsf{H}_{i.3.1}$, where we replace the random function $\mathbf{F}_i \colon \{0,1\}^i \to \mathbb{Z}_p^{2k}$ by

$$\mathbf{F}'_i \colon \{0,1\}^i \to \mathbb{Z}_p^{2k},\ \ \mathbf{F}'_i(\nu) := \left(\mathbf{A}_0^\perp | \mathbf{A}_1^\perp\right) \begin{pmatrix} \mathit{\Gamma}_i(\nu) \\ \mathit{\Upsilon}_i(\nu) \end{pmatrix},$$

where $\nu \in \{0,1\}^i$ is a i-bit string and $\mathit{\Gamma}_i, \mathit{\Upsilon}_i : \{0,1\}^i \to \mathbb{Z}_p^k$ are two independent random functions. With probability $1 - \mathit{\Delta}_{\mathcal{D}_{2k,k}}$ the matrix $(\mathbf{A}_0^\perp | \mathbf{A}_1^\perp)$ has full rank. In this case, going from game $\mathsf{H}_{i.3}$ to game $\mathsf{H}_{i.3.1}$ consists merely in a change of basis, thus, these two games are perfectly indistinguishable. We obtain $|\widehat{\varepsilon}_{3.i} - \widehat{\varepsilon}_{3.i.1}| \leq \mathit{\Delta}_{\mathcal{D}_{2k,k}}$.

We now define

$$\mathbf{F}_{i+1} \colon \{0,1\}^{i+1} \to \mathbb{Z}_p^{2k},\ \ \mathbf{F}_{i+1}(\nu) := \begin{cases} \left(\mathbf{A}_0^\perp | \mathbf{A}_1^\perp\right) \begin{pmatrix} \mathit{\Gamma}'_i(\nu_{|i}) \\ \mathit{\Upsilon}_i(\nu_{|i}) \end{pmatrix} & \text{if } \nu_{i+1} = 0 \\ \left(\mathbf{A}_0^\perp | \mathbf{A}_1^\perp\right) \begin{pmatrix} \mathit{\Gamma}_i(\nu_{|i}) \\ \mathit{\Upsilon}'_i(\nu_{|i}) \end{pmatrix} & \text{else} \end{cases},$$

where $\mathit{\Gamma}'_i, \mathit{\Upsilon}'_i : \{0,1\}^i \to \mathbb{Z}_p^k$ are fresh independent random functions. Now $\mathbf{F}_{i+1}$ constitutes a random function $\{0,1\}^{i+1} \to \mathbb{Z}_p^{2k}$.

Replacing $\mathbf{F}'_i(\mathsf{ctr}_{|i})$ by $\mathbf{F}_{i+1}(\mathsf{ctr}_{|i+1})$ does not show up in any of the tag queries, as we have

$$\mathbf{F}_{i+1}(\mathsf{ctr}_{i+1})^\top [\mathbf{t}]_1 = \mathbf{F}_{i+1}(\mathsf{ctr}_{i+1})^\top [\mathbf{A}_{\mathsf{ctr}_{i+1}}]_1 \mathbf{r}$$

$$
\begin{aligned}
&= \begin{cases} \varGamma'_i(\mathsf{ctr}_{|i})\mathbf{A}_0^{\perp}[\mathbf{A}_0]_1\mathbf{r} + \varUpsilon_i(\mathsf{ctr}_{|i})\mathbf{A}_1^{\perp}[\mathbf{A}_0]_1\mathbf{r} & \text{if } \mathsf{ctr}_{i+1} = 0 \\ \varGamma_i(\mathsf{ctr}_{|i})\mathbf{A}_0^{\perp}[\mathbf{A}_1]_1\mathbf{r} + \varUpsilon'_i(\mathsf{ctr}_{|i})\mathbf{A}_1^{\perp}[\mathbf{A}_1]_1\mathbf{r} & \text{else} \end{cases} \\
&= \begin{cases} \varUpsilon_i(\mathsf{ctr}_{|i})\mathbf{A}_1^{\perp}[\mathbf{A}_0]_1\mathbf{r} & \text{if } \mathsf{ctr}_{i+1} = 0 \\ \varGamma_i(\mathsf{ctr}_{|i})\mathbf{A}_0^{\perp}[\mathbf{A}_1]_1\mathbf{r} & \text{else} \end{cases} \\
&= \mathbf{F}'_i(\mathsf{ctr}_{|i})^{\top}[\mathbf{A}_{\mathsf{ctr}_{i+1}}]_1\mathbf{r}.
\end{aligned}
$$

In the verification oracle we check $[\mathbf{t}]_1 \in \operatorname{span}([\mathbf{A}_0]) \cup \operatorname{span}([\mathbf{A}_1])$, define $d_{[\mathbf{t}]} = 0$ if $\mathbf{t} \in \operatorname{span}(\mathbf{A}_0)$ and $d_{[\mathbf{t}]} = 1$ if $\mathbf{t} \in \operatorname{span}(\mathbf{A}_1)$ and replace $\mathbf{F}_i(\mathsf{ctr}_{|i})$ by $\mathbf{F}_{i+1}(\mathsf{ctr}_{|i}|d_{[\mathbf{t}]})$. Thus, by similar reasoning as for tag queries, the change does not show up in the final verification query either.

Altogether, we obtain

$$|\widehat{\varepsilon}_{3.3} - \widehat{\varepsilon}_{3.4}| \leq \varDelta_{\mathcal{D}_{2k,k}}.$$

$\mathsf{H}_{i.4} \rightsquigarrow \mathsf{H}_{i.5}$**:** From game $\mathsf{H}_{i.5}$ on, we extend the set $\mathcal{S}$ in the verification oracle from $\mathcal{S}_{i.4} := \{\mathbf{F}_{i+1}(\mathsf{ctr}'_{|i}|d_{[\mathbf{t}]}) : \mathsf{ctr}' \leq \mathsf{ctr}\}$ to $\mathcal{S}_{i.5} := \{\mathbf{F}_{i+1}(\mathsf{ctr}'_{|i}|b) : \mathsf{ctr}' \leq \mathsf{ctr}, b \in \{0,1\}\}$. That is, we regard a verification query $([\mathbf{t}]_1, \varPi, [u']_1)$ as valid, if there exists a $\mathsf{ctr}' \leq \mathsf{ctr}$ such that $[u']_1 = (\mathbf{k}_0 + \mathbf{F}_{i+1}(\mathsf{ctr}'_{|i}|b))^{\top}[\mathbf{t}]_1$ for $b \in \{0,1\}$ arbitrary, instead of requiring $b = d_{[\mathbf{t}]}$ (where $d_{[\mathbf{t}]} = 0$ if $\mathbf{t} \in \operatorname{span}(\mathbf{A}_0)$ and $d_{[\mathbf{t}]} = 1$ if $\mathbf{t} \in \operatorname{span}(\mathbf{A}_1)$). As changing the verification oracle does not change the view of the adversary before providing its output and as we have $\mathcal{S}_{i.4} \subseteq \mathcal{S}_{i.5}$, the transition from game $\mathsf{H}_{i.4}$ to game $\mathsf{H}_{i.5}$ can only increase the chance of the adversary. We thus have

$$\widehat{\varepsilon}_{i.4} \leq \widehat{\varepsilon}_{i.5}.$$

$\mathsf{H}_{i.5} \rightsquigarrow \mathsf{H}_{i.6}$**:** The difference between game $\mathsf{H}_{i.5}$ and game $\mathsf{H}_{i.6}$ is that in the latter we only regard a verification query $([\mathbf{t}]_1, \varPi, [u]_1)$ valid, if there exists a $\mathsf{ctr}' \leq \mathsf{ctr}$ such that $[u]_1 = (\mathbf{k}_0 + \mathbf{F}_{i+1}(\mathsf{ctr}'_{|i}|\mathsf{ctr}'_{i+1}))^{\top}[\mathbf{t}]_1$ (instead of allowing the last bit to be arbitrary). As the only way an adversary can learn the image of $\mathbf{F}_{i+1}$ on a value is via tag queries and $\mathbf{F}_{i+1}$ is a random function, a union bound over the elements in $\mathcal{Q}_{\mathsf{tag}}$ yields

$$|\widehat{\varepsilon}_{i.5} - \widehat{\varepsilon}_{i.6}| \leq \frac{Q}{p}.$$

$\mathsf{H}_{i.6} \rightsquigarrow \mathsf{H}_{i.7}$**:** The oracle VerO does not perform the additional check $[\mathbf{t}]_1 \in \operatorname{span}([\mathbf{A}_0]_1) \cup \operatorname{span}([\mathbf{A}_1]_1)$ anymore from game $\mathsf{H}_{i.7}$ on. This is justified by the soundness of PS. As in transition $\mathsf{H}_{i.2} \rightsquigarrow \mathsf{H}_{i.3}$ we obtain

$$\widehat{\varepsilon}_{i.6} = \widehat{\varepsilon}_{i.7}.$$

$\mathsf{H}_{i.7} \rightsquigarrow \mathsf{H}_{i.8}$**:** This transition is similar to transition G_0 to G_1 in Theorem 1. Namely, for an adversary $\mathcal{A}$ distinguishing the two games, we can employ the composable zero-knowledge property of PS to obtain an adversary $\mathcal{B}_{i.7}$ such that $T(\mathcal{B}_{i.8}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ and

$$|\widehat{\varepsilon}_{i.7} - \widehat{\varepsilon}_{i.8}| \leq \mathrm{Adv}^{\mathrm{ZK}}_{\mathsf{PS},\mathcal{B}_{i.7}}(\lambda).$$

$\mathbf{H}_{i.8} \rightsquigarrow \mathbf{G}_{3.(i+1)}$**:** We switch $[\mathbf{t}]_1$ generated by TagO to uniformly random over $\mathbb{G}_1^{2k}$, using the $\mathcal{D}_{2k,k}$-MDDH assumption first on $[\mathbf{A}_0]_1$, then on $[\mathbf{A}_1]_1$. Similarly than for the transition $\mathsf{G}_{3.i} \rightsquigarrow \mathsf{H}_{i.1}$, we obtain an adversary $\mathcal{B}_{i.8}$ with $T(\mathcal{B}_{i.8}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ such that

$$|\widehat{\varepsilon}_{i.8} - \varepsilon_{3.(i+1)}| \leq 2k \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_1,\mathcal{D}_{2k,k},\mathcal{B}_{i.8}}(\lambda) + \tfrac{2}{p-1}.$$

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 Figure 2 is* UF-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* TagO, poly *is independent of Q, and*

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

Proof. We employ an intermediary game G_0 to prove UF-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 $\mathcal{A}$.

```
$\mathrm{Exp}^{\text{uf-cma}}_{\mathcal{A}}(\lambda)$, $\boxed{\mathsf{G}}$:
$\mathcal{Q}_{\mathsf{tag}} := \emptyset$
$\boxed{\mathsf{ctr} := 0}$
$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$
$\mathbf{A}_0, \mathbf{A}_1 \leftarrow_R \mathcal{D}_{2k,k}$
$pars := (\mathcal{PG}, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1)$
$crs \leftarrow \mathsf{PGen}(1^\lambda, pars)$
$\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}$
$pp := (\mathcal{PG}, [\mathbf{A}_0]_1, crs)$
$(\mu^\star, \mathsf{tag}^\star) \leftarrow \mathcal{A}^{\textsc{TagO}(\cdot)}(pp)$
if $\mu^\star \notin \mathcal{Q}_{\mathsf{tag}}$
    and $\textsc{VerO}(\mu^\star, \mathsf{tag}^\star) = 1$
    return 1
else return 0
```

```
$\textsc{TagO}(\mu)$:
$\mathcal{Q}_{\mathsf{tag}} := \mathcal{Q}_{\mathsf{tag}} \cup \{\mu\}$
$\boxed{\mathsf{ctr} := \mathsf{ctr} + 1}$
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{t}]_1 := [\mathbf{A}_0]_1 \mathbf{r}$
$\varPi \leftarrow \mathsf{PPrv}(crs, [\mathbf{t}]_1, \mathbf{r})$
$[u]_1 := (\mathbf{k}_0 + \mu \mathbf{k}_1 \boxed{+\mathbf{F}(\mathsf{ctr})})^\top [\mathbf{t}]_1$
$\mathsf{tag} := ([\mathbf{t}]_1, \varPi, [u]_1)$
return tag

$\textsc{VerO}(\mu^\star, \mathsf{tag}^\star)$ :
parse $\mathsf{tag}^\star =: ([\mathbf{t}]_1, \varPi, [u]_1)$
$b \leftarrow \mathsf{PVer}([\mathbf{t}]_1, \varPi)$
if $b = 1$ and $[u]_1 \neq [0]_1$ and $\boxed{\exists \mathsf{ctr}' \leq \mathsf{ctr}:}$
    $[u]_1 = (\mathbf{k}_0 + \mu^\star \mathbf{k}_1 \boxed{+\mathbf{F}_i(\mathsf{ctr}')})^\top [\mathbf{t}]_1$
     return 1
else return 0
```

Fig. 7: The UF-CMA security experiment and game G for the UF-CMA proof of MAC in Figure 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.

$\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{A}}(\boldsymbol{\lambda}) \rightsquigarrow \mathbf{G}_0$**:** 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 TagO oracle (which takes no input), receives $([\mathbf{t}]_1, \Pi, [u']_1)$, 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 $\mathrm{Adv}^{\text{uf-cma}}_{\mathsf{MAC},\mathcal{A}}(\lambda) = \mathrm{Adv}^{\text{core}}_{0,\mathcal{B}}(\lambda)$ and $\varepsilon_0 = \mathrm{Adv}^{\text{core}}_{1,\mathcal{B}}(\lambda)$. The core lemma yields

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

and thus altogether we obtain

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

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

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

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:

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

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 $\textsc{VerO}$ is (marginally) uniformly random over $\mathbb{G}_1$.

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

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

4 Tightly secure signature scheme

In this section, we present a signature scheme SIG for signing messages from $\mathbb{Z}_p$, described in Figure 8, 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 Section 3, and functions as a stepping stone to explain the main ideas of the upcoming structure-preserving signature in Section 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 [41, 39], structure-preserving signatures [40], and identity-based encryption schemes [14].

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

$\underline{\mathsf{Ver}(pk, \mu \in \mathbb{Z}_p, \sigma)}$ :
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 $e([\mathbf{u}]_1^\top, [\mathbf{A}]_2)$
   $= e([\mathbf{t}]_1^\top, [\mathbf{K}_0\mathbf{A}]_2 + \mu[\mathbf{K}_1\mathbf{A}]_2)$
    return 1
else return 0
```

Fig. 8: Tightly UF-CMA secure signature scheme SIG.

Theorem 2 (Security of SIG). *If* PS := (PGen, PPrv, PVer, 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 Figure 8 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*

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

By using the KMDH assumption, we verify the forgery with the signing key; then we introduce the MAC in the kernel of $\mathbf{A}$. Since we always know $\mathbf{A}$ over $\mathbb{Z}_p$, we extract the MAC tag from the forgery and break the MAC security. The proof idea is similar, but weaker than [14].

Proof. We proceed via a series of hybrid games G_0 to G_1, described in Figure 9. By ε_i we denote 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 $\mathcal{A}$.

$\mathbf{Exp}^{\text{uf-cma}}_{\mathsf{SIG},\mathcal{A}}(\boldsymbol{\lambda}) \rightsquigarrow \mathsf{G}_0$**:** Here we change the verification oracle as described in Fig. 9. Note that a pair $(\mu^\star, \sigma^\star)$ that passes VerO in G_0 always passes the VerO in $\mathrm{Exp}^{\text{uf-cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda)$. Thus, to bound $|\mathrm{Adv}^{\text{uf-cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda) - \varepsilon_1|$, it suffices to bound the probability that $\mathcal{A}$ produces $(\mu^\star, \sigma^\star)$ that passes VerO in $\mathrm{Exp}^{\text{uf-cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda)$ but not in G_0. We write $\sigma^\star := ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1)$, and the verification equation in $\mathrm{Exp}^{\text{uf-cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda)$ as:

$$e([\mathbf{u}]_1^\top, [\mathbf{A}]_2) = e([\mathbf{t}]_1^\top, [(\mathbf{K}_0 + \mu^\star\mathbf{K}_1)\mathbf{A}]_2)$$
$$\Leftrightarrow e([\mathbf{u}]_1 - [\mathbf{t}]_1^\top(\mathbf{K}_0 + \mu^\star\mathbf{K}_1), [\mathbf{A}]_2) = \mathbf{0}$$

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

$$[\mathbf{u}]_1 - [\mathbf{t}]_1^\top(\mathbf{K}_0 + \mu^\star\mathbf{K}_1)$$

```
$\mathsf{G}_0, \boxed{\mathsf{G}_1}$:
$\mathcal{Q}_{\mathsf{sign}} := \emptyset$
$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$
$\mathbf{A}_0, \mathbf{A}_1 \leftarrow \mathcal{D}_{2k,k}$
$pars := (\mathcal{PG}, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1)$
$crs \leftarrow \mathsf{PGen}(1^\lambda, pars)$
$\mathbf{A} \leftarrow_R \mathcal{D}_k$
$\boxed{\mathbf{a}^\perp \in \mathsf{orth}(\mathbf{A})}$
$\mathbf{K}_0, \mathbf{K}_1 \leftarrow_R \mathbb{Z}_p^{2k \times (k+1)}$
$\boxed{\mathbf{k}_0, \mathbf{k}_1 \leftarrow_R \mathbb{Z}_p^{2k}}$
$pk := (\mathcal{PG}, [\mathbf{A}_0]_1, crs,$
      $[\mathbf{A}]_2, [\mathbf{K}_0\mathbf{A}]_2, [\mathbf{K}_1\mathbf{A}]_2)$
$sk := (\mathbf{K}_0, \mathbf{K}_1)$
$(\mu^\star, \sigma^\star) \leftarrow_R \mathcal{A}^{\textsc{SignO}(\cdot)}(pk)$
if $\mu^\star \notin \mathcal{Q}_{\mathsf{sign}}$
  and $\textsc{VerO}(\mu^\star, \sigma^\star) = 1$
    return 1
else return 0
```

```
$\textsc{SignO}(\mu)$:
$\mathcal{Q}_{\mathsf{sign}} := \mathcal{Q}_{\mathsf{sign}} \cup \{\mu\}$
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$
$[\mathbf{t}]_1 := [\mathbf{A}_0]_1\mathbf{r}$
$\Pi \leftarrow \mathsf{PPrv}(crs, [\mathbf{t}]_1, \mathbf{r})$
$[\mathbf{u}]_1 := (\mathbf{K}_0 + \mu\mathbf{K}_1)^\top[\mathbf{t}]_1 \boxed{+\mathbf{a}^\perp(\mathbf{k}_0 + \mu\mathbf{k}_1)^\top[\mathbf{t}]_1}$
$\sigma := ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1)$
return $\sigma$

$\textsc{VerO}(\mu^\star, \sigma^\star)$:
parse $\sigma^\star := ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1)$
$b \leftarrow \mathsf{PVer}(pk, [\mathbf{t}]_1, \Pi)$
if $b = 1$ and $[\mathbf{u}]_1 \neq [\mathbf{0}]_1$ and
  $[\mathbf{u}]_1 = (\mathbf{K}_0 + \mu^\star\mathbf{K}_1)^\top[\mathbf{t}]_1 \boxed{+\mathbf{a}^\perp(\mathbf{k}_0 + \mu^\star\mathbf{k}_1)^\top[\mathbf{t}]_1}$
    return 1
else return 0
```

Fig. 9: Games G_0 to G_1 for proving Theorem 2. The instructions inside a solid frame are only present in the games marked by a solid frame.

is a non-zero vector in the kernel of $\mathbf{A}$.

Thus we can construct an adversary $\mathcal{B}$ with $T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ on the $\mathcal{D}_k$-KMDH assumption from $\mathcal{A}$ as follows. On receiving $(\mathcal{PG}, [\mathbf{A}]_2)$ from the $\mathcal{D}_k$-KMDH experiment, $\mathcal{B}$ can sample all other parameters itself and simulate G_0 for $\mathcal{A}$. If $\mathcal{A}$ outputs the tuple $(\mu^\star, ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1))$, then $\mathcal{B}$ outputs the value $[\mathbf{u}]_1 - [\mathbf{t}]_1^\top(\mathbf{K}_0 + \mu^\star\mathbf{K}_1)$ to its own experiment.

Lemma 2 finally yields an adversary $\mathcal{B}'$ with $T(\mathcal{B}') \approx T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ and

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

$\mathbf{\mathsf{G}_0} \rightsquigarrow \mathbf{\mathsf{G}_1}$**:** For $i \in \{0,1\}$ we can replace $\mathbf{K}_i$ by $\mathbf{K}_i + \mathbf{k}_i(\mathbf{a}^\perp)^\top$ for $\mathbf{a}^\perp \in \mathsf{orth}(\mathbf{A})$ and $\mathbf{k}_i \leftarrow_R \mathbb{Z}_p^{2k}$, as both are distributed identically. Further, as $(\mathbf{a}^\perp)^\top \cdot \mathbf{A} = \mathbf{0}$, this change does not show up in the public key pk. Thus, we have

$$\varepsilon_0 = \varepsilon_1.$$

Game G_1: We can employ the $\mathsf{UF\text{-}CMA}$ security of MAC given in Figure 2 to bound the probability of an adversary winning game G_1. Let $\mathcal{A}$ be an adversary on G_1. We construct an adversary $\mathcal{B}$ with $T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ on the $\mathsf{UF\text{-}CMA}$ security of MAC as follows.

On input $pp = (\mathcal{PG}, [\mathbf{A}_0]_1, crs)$ of $\mathrm{Exp}^{\text{uf-cma}}_{\mathcal{A}}(\lambda)$ adversary $\mathcal{B}$ samples $\mathbf{A} \leftarrow_R \mathbf{D}_{k+1,k}$ and $\mathbf{K}_0, \mathbf{K}_1 \leftarrow_R \mathbb{Z}_p^{2k\times(k+1)}$, chooses $\mathbf{a}^\perp \in \mathsf{orth}(\mathbf{A})$ and forwards $pk := (\mathcal{PG}, [\mathbf{A}_0]_1, crs, [\mathbf{A}]_2, [\mathbf{K}_0\mathbf{A}]_2, [\mathbf{K}_1\mathbf{A}]_2)$ to $\mathcal{A}$.

On a signing query μ of $\mathcal{A}$, the adversary $\mathcal{B}$ queries its own tag oracle to obtain $\mathsf{tag} = ([\mathbf{t}]_1, \Pi, [u]_1)$. Then, $\mathcal{B}$ computes $[\mathbf{u}]_1 := (\mathbf{K}_0 + \mu\mathbf{K}_1)^\top[\mathbf{t}]_1 + \mathbf{a}^\perp[u]_1$ and forwards $\sigma := ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1)$ to $\mathcal{A}$.

Let $(\mu^\star, \sigma^\star)$ be a forgery of $\mathcal{A}$ with $\sigma^\star = ([\mathbf{t}^\star]_1, \varPi^\star, [\mathbf{u}^\star]_1)$. Then $\mathcal{B}$ computes $[\mathbf{u}']_1 := (\mathbf{K}_0 + \mu^\star \mathbf{K}_1)^\top [\mathbf{t}]_1$ and (if possible) chooses $[u^\star]_1$ such that $\mathbf{a}^\perp [u^\star]_1 = [\mathbf{u}^\star]_1 - [\mathbf{u}']_1$ (note that this doable efficiently given $\mathbf{a}^\perp$) Finally, $\mathcal{B}$ outputs $(\mu^\star, \mathsf{tag}^\star)$ with $\mathsf{tag}^\star := ([\mathbf{t}^\star]_1, \varPi^\star, [u^\star]_1)$. If $(\mu^\star, \sigma^\star)$ was a successful forgery of $\mathcal{A}$ then, by the definition of game G_1, $(\mu^\star, \mathsf{tag}^\star)$ is a successful forgery in $\mathrm{Exp}^{\text{uf-cma}}_{\mathcal{A}}(\lambda)$. This yields

$$\varepsilon_1 \leq \mathrm{Adv}^{\text{uf-cma}}_{\mathsf{MAC},\mathcal{B}}(\lambda).$$

5 Tightly secure structure-preserving signature scheme

In this section we present a structure-preserving signature scheme SPS, described in Figure 10, 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 Section 4 by using a similar idea of [40]. 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.

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

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

$\mathsf{Ver}(pk, \sigma, [\mathbf{m}]_1)$:
parse $\sigma := ([\mathbf{t}]_1, \varPi, [\mathbf{u}]_1)$
$b \leftarrow \mathsf{PVer}(pk, [\mathbf{t}]_1, \varPi)$
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. 10: Tightly UF-CMA secure structure-preserving signature scheme SPS with message space $\mathbb{G}_1^n$.

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. 10 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*

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

When using PS *from Section 2.5, we obtain*

$$\mathrm{Adv}^{\text{uf-cma}}_{\mathsf{SPS},\mathcal{A}}(\lambda) \leq (4k\lceil \log Q \rceil + 2) \cdot \mathrm{Adv}^{\text{mddh}}_{\mathcal{PG},\mathbb{G}_1,\mathcal{D}_{2k,k},\mathcal{B}}(\lambda)$$

```
$\mathsf{G}_0, \boxed{\mathsf{G}_1, \dbox{\mathsf{G}_2, \boxed{\boxed{\mathsf{G}_3, \colorbox{gray}{$\mathsf{G}_4$}}}}}$:
$\mathcal{Q}_{\mathsf{sign}} := \emptyset$
$\dbox{\mathsf{ctr} := 0}$, $\colorbox{gray}{$\widetilde{\mathsf{ctr}} \leftarrow_R [Q]$}$
$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$
$\mathbf{A}_0, \mathbf{A}_1 \leftarrow_R \mathcal{D}_{2k,k}$
$pars := (\mathcal{PG}, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1)$
$\mathbf{A} \leftarrow_R \mathcal{D}_k$
$\boxed{\mathbf{a}^\perp \in \mathsf{orth}(\mathbf{A})}$
$crs \leftarrow \mathsf{PGen}(pars, 1^\lambda)$
$\mathbf{K}_0 \leftarrow_R \mathbb{Z}_p^{2k \times (k+1)}$
$\boxed{\mathbf{k}_0 \leftarrow_R \mathbb{Z}_p^{2k}}$
$\mathbf{K} \leftarrow_R \mathbb{Z}_p^{(n+1) \times (k+1)}$
$pk := (crs, pars, [\mathbf{A}]_2,$
        $[\mathbf{K}_0\mathbf{A}]_2, [\mathbf{K}\mathbf{A}]_2)$
$sk := (\mathbf{K}_0, \mathbf{K})$
$([\mathbf{m}^\star]_1, \sigma^\star) \leftarrow_R \mathcal{A}^{\textsc{SignO}(\cdot)}(pk)$
if $[\mathbf{m}^\star]_1 \notin \mathcal{Q}_{\mathsf{sign}}$
  and $\textsc{VerO}([\mathbf{m}^\star]_1, \sigma^\star) = 1$
    return 1
else return 0

$\textsc{SignO}([\mathbf{m}]_1 \in \mathbb{G}_1^n)$:
$\mathcal{Q}_{\mathsf{sign}} := \mathcal{Q}_{\mathsf{sign}} \cup \{[\mathbf{m}]_1\}$
$\dbox{\mathsf{ctr} := \mathsf{ctr} + 1}$
$\mathbf{r} \leftarrow_R \mathbb{Z}_p^k$, $\boxed{\boxed{\mathbf{r} \leftarrow_R (\mathbb{Z}_p^k)^*}}$, $[\mathbf{t}]_1 := [\mathbf{A}_0]_1\mathbf{r}$
$\Pi \leftarrow \mathsf{PPrv}(crs, [\mathbf{t}]_1, \mathbf{r})$
$[\mathbf{u}]_1 := \mathbf{K}_0^\top [\mathbf{t}]_1 + \mathbf{K}^\top \begin{bmatrix} \mathbf{m} \\ 1 \end{bmatrix}_1$
        $\boxed{+\mathbf{a}^\perp(\mathbf{k}_0 \dbox{+\mathbf{F}(\mathsf{ctr})})^\top [\mathbf{t}]_1}$
return $\sigma := ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1)$

$\textsc{VerO}([\mathbf{m}^\star]_1, \sigma^\star)$:
parse $\sigma =: ([\mathbf{t}]_1, \Pi, [\mathbf{u}]_1)$
$b \leftarrow \mathsf{PVer}(pk, [\mathbf{t}]_1, \Pi)$
if  $b = 1$ and $\dbox{\exists \mathsf{ctr}' \le \mathsf{ctr} :}$
  $\colorbox{gray}{$\mathsf{ctr}' = \widetilde{\mathsf{ctr}}$}$ and
  $[\mathbf{u}]_1 = \mathbf{K}_0^\top [\mathbf{t}]_1 + \mathbf{K}^\top \begin{bmatrix} \mathbf{m} \\ 1 \end{bmatrix}_1$
  $\boxed{+\mathbf{a}^\perp(\mathbf{k}_0 \dbox{+\mathbf{F}(\mathsf{ctr}')})^\top [\mathbf{t}]_1}$
    return 1
else return 0
```

Fig. 11: Games G_0 to G_4 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.

$$+ (2\lceil \log Q \rceil + 3) \cdot \mathrm{Adv}^{\mathrm{mddh}}_{\mathcal{PG},\mathbb{G}_2,\mathcal{D}_k,\mathcal{B}'}(\lambda) + \lceil \log Q \rceil \cdot \Delta_{\mathcal{D}_{2k,k}}$$
$$+ \tfrac{4\lceil \log Q \rceil + 2}{p-1} + \tfrac{(Q+1)\lceil \log Q \rceil + Q}{p} + \tfrac{Q}{p^k}.$$

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. (Inuitively, 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_4, described in Figure 11. By ε_i we denote the advantage of $\mathcal{A}$ to win G_i.

$\mathbf{Exp}^{\mathsf{uf\text{-}cma}}_{\mathsf{SPS},\mathcal{A}}(\boldsymbol{\lambda}) \rightsquigarrow \mathbf{G_0}$: Here we change the verification oracle as described in Fig. 11.

Note that a pair $(\mu^\star, \sigma^\star)$ that passes $\textsc{VerO}$ in G_0 always passes the $\textsc{VerO}$ 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 $\textsc{VerO}$ in $\mathrm{Exp}^{\text{uf-cma}}_{\mathsf{SPS},\mathcal{A}}(\lambda)$, 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 $\textsc{SignO}([\mathbf{m}]_1)$, $\mathcal{B}$ uses its oracle $\textsc{TagO}$, 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 $\big([\mathbf{m}^\star]_1, \sigma^\star\big)$ 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 $\textsc{VerO}(\mathsf{tag})$, and forwards the answer to $\mathcal{A}$.

$\mathsf{G_2} \rightsquigarrow \mathsf{G_3}$: In game G_2 the vectors $\mathbf{r}$ sampled by $\textsc{SignO}$ 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

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

$\mathsf{G_3} \rightsquigarrow \mathsf{G_4}$: 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

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

Game $\mathsf{G_4}$: 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

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

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 Figure 10 can sign only messages from $\mathbb{G}_1^n$. By applying the generic transformation from [40, Section 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 [40] 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.

Acknowledgments. We thank Carla Ràfols for insightful discussions regarding the OR proof, and Yohei Watanabe for spotting a mistake in the write-up.

References

[1] Masayuki Abe, Melissa Chase, Bernardo David, Markulf Kohlweiss, Ryo Nishimaki, and Miyako Ohkubo. “Constant-Size Structure-Preserving Signatures: Generic Constructions and Simple Assumptions”. In: *ASIACRYPT 2012*. Ed. by Xiaoyun Wang and Kazue Sako. Vol. 7658. LNCS. Springer, Heidelberg, Dec. 2012, pp. 4–24. DOI: 10.1007/978-3-642-34961-4_3.

[2] Masayuki Abe, Melissa Chase, Bernardo David, Markulf Kohlweiss, Ryo Nishimaki, and Miyako Ohkubo. “Constant-Size Structure-Preserving Signatures: Generic Constructions and Simple Assumptions”. In: *Journal of Cryptology* 29.4 (Oct. 2016), pp. 833–878. DOI: 10.1007/s00145-015-9211-7.

[3] Masayuki Abe, Bernardo David, Markulf Kohlweiss, Ryo Nishimaki, and Miyako Ohkubo. “Tagged One-Time Signatures: Tight Security and Optimal Tag Size”. In: *PKC 2013*. Ed. by Kaoru Kurosawa and Goichiro Hanaoka. Vol. 7778. LNCS. Springer, Heidelberg, 2013, pp. 312–331. DOI: 10.1007/978-3-642-36362-7_20.

[4] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. “Structure-Preserving Signatures and Commitments to Group Elements”. In: *Journal of Cryptology* 29.2 (Apr. 2016), pp. 363–421. DOI: 10.1007/s00145-014-9196-7.

[5] Masayuki Abe, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. “Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups”. In: *CRYPTO 2011*. Ed. by Phillip Rogaway. Vol. 6841. LNCS. Springer, Heidelberg, Aug. 2011, pp. 649–666.

[6] Masayuki Abe, Dennis Hofheinz, Ryo Nishimaki, Miyako Ohkubo, and Jiaxin Pan. “Compact Structure-Preserving Signatures with Almost Tight Security”. In: *CRYPTO 2017*. Springer, 2017, pp. 548–580.

[7] Tolga Acar, Kristin Lauter, Michael Naehrig, and Daniel Shumow. “Affine Pairings on ARM”. In: *PAIRING 2012*. Ed. by Michel Abdalla and Tanja Lange. Vol. 7708. LNCS. Springer, Heidelberg, May 2013, pp. 203–209. DOI: 10.1007/978-3-642-36334-4_13.

[8] Nuttapong Attrapadung, Goichiro Hanaoka, and Shota Yamada. “A Framework for Identity-Based Encryption with Almost Tight Security”. In: *ASIACRYPT 2015, Part I*. Ed. by Tetsu Iwata and Jung Hee Cheon. Vol. 9452. LNCS. Springer, Heidelberg, 2015, pp. 521–549. DOI: 10.1007/978-3-662-48797-6_22.

[9] Paulo S. L. M. Barreto, Craig Costello, Rafael Misoczki, Michael Naehrig, Geovandro C. C. F. Pereira, and Gustavo Zanon. “Subgroup Security in Pairing-Based Cryptography”. In: *LATINCRYPT 2015*. Ed. by Kristin E. Lauter and Francisco Rodríguez-Henríquez. Vol. 9230. LNCS. Springer, Heidelberg, Aug. 2015, pp. 245–265. DOI: 10.1007/978-3-319-22174-8_14.

[10] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. “P-signatures and Noninteractive Anonymous Credentials”. In: *TCC 2008*. Ed. by Ran Canetti. Vol. 4948. LNCS. Springer, Heidelberg, Mar. 2008, pp. 356–374.

[11] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. “Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements”. In: *EUROCRYPT 2000*. Ed. by Bart Preneel. Vol. 1807. LNCS. Springer, Heidelberg, May 2000, pp. 259–274.

[12] Mihir Bellare and Shafi Goldwasser. “New Paradigms for Digital Signatures and Message Authentication Based on Non-Interative Zero Knowledge Proofs”. In: *CRYPTO’89*. Ed. by Gilles Brassard. Vol. 435. LNCS. Springer, Heidelberg, Aug. 1990, pp. 194–211.

[13] Olivier Blazy, Georg Fuchsbauer, Malika Izabachène, Amandine Jambert, Hervé Sibert, and Damien Vergnaud. “Batch Groth-Sahai”. In: *ACNS 10*. Ed. by Jianying Zhou and Moti Yung. Vol. 6123. LNCS. Springer, Heidelberg, June 2010, pp. 218–235.

[14] Olivier Blazy, Eike Kiltz, and Jiaxin Pan. "(Hierarchical) Identity-Based Encryption from Affine Message Authentication". In: *CRYPTO 2014, Part I*. Ed. by Juan A. Garay and Rosario Gennaro. Vol. 8616. LNCS. Springer, Heidelberg, Aug. 2014, pp. 408–425. DOI: 10.1007/978-3-662-44371-2_23.

[15] Manuel Blum, Paul Feldman, and Silvio Micali. "Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract)". In: *20th ACM STOC*. ACM Press, May 1988, pp. 103–112.

[16] Dan Boneh, Ilya Mironov, and Victor Shoup. "A Secure Signature Scheme from Bilinear Maps". In: *CT-RSA 2003*. Ed. by Marc Joye. Vol. 2612. LNCS. Springer, Heidelberg, Apr. 2003, pp. 98–110.

[17] Jan Camenisch, Maria Dubovitskaya, and Kristiyan Haralambiev. "Efficient Structure-Preserving Signature Scheme from Standard Assumptions". In: *SCN 12*. Ed. by Ivan Visconti and Roberto De Prisco. Vol. 7485. LNCS. Springer, Heidelberg, Sept. 2012, pp. 76–94.

[18] Julien Cathalo, Benoît Libert, and Moti Yung. "Group Encryption: Non-interactive Realization in the Standard Model". In: *ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Vol. 5912. LNCS. Springer, Heidelberg, Dec. 2009, pp. 179–196.

[19] Melissa Chase and Markulf Kohlweiss. "A New Hash-and-Sign Approach and Structure-Preserving Signatures from DLIN". In: *SCN 12*. Ed. by Ivan Visconti and Roberto De Prisco. Vol. 7485. LNCS. Springer, Heidelberg, Sept. 2012, pp. 131–148.

[20] Jie Chen, Junqing Gong, and Jian Weng. "Tightly Secure IBE Under Constant-Size Master Public Key". In: *PKC 2017, Part I*. Ed. by Serge Fehr. Vol. 10174. LNCS. Springer, Heidelberg, Mar. 2017, pp. 207–231.

[21] Jie Chen and Hoeteck Wee. "Fully, (Almost) Tightly Secure IBE and Dual System Groups". In: *CRYPTO 2013, Part II*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8043. LNCS. Springer, Heidelberg, Aug. 2013, pp. 435–460. DOI: 10.1007/978-3-642-40084-1_25.

[22] Yevgeniy Dodis, Eike Kiltz, Krzysztof Pietrzak, and Daniel Wichs. "Message Authentication, Revisited". In: *EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. LNCS. Springer, Heidelberg, Apr. 2012, pp. 355–374.

[23] Andreas Enge and Jérôme Milan. "Implementing Cryptographic Pairings at Standard Security Levels". In: *SPACE 2014*. 2014, pp. 28–46.

[24] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. "An Algebraic Framework for Diffie-Hellman Assumptions". In: *CRYPTO 2013, Part II*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8043. LNCS. Springer, Heidelberg, Aug. 2013, pp. 129–147. DOI: 10.1007/978-3-642-40084-1_8.

[25] Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. "Tightly CCA-Secure Encryption Without Pairings". In: *EUROCRYPT 2016, Part I*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9665. LNCS. Springer, Heidelberg, May 2016, pp. 1–27. DOI: 10.1007/978-3-662-49890-3_1.

[26] Romain Gay, Dennis Hofheinz, and Lisa Kohl. "Kurosawa-Desmedt Meets Tight Security". In: *CRYPTO 2017*. Springer, 2017, pp. 133–160.

[27] Junqing Gong, Jie Chen, Xiaolei Dong, Zhenfu Cao, and Shaohua Tang. "Extended Nested Dual System Groups, Revisited". In: *PKC 2016, Part I*. Ed. by Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang. Vol. 9614. LNCS. Springer, Heidelberg, Mar. 2016, pp. 133–163. DOI: 10.1007/978-3-662-49384-7_6.

[28] Gurleen Grewal, Reza Azarderakhsh, Patrick Longa, Shi Hu, and David Jao. "Efficient Implementation of Bilinear Pairings on ARM Processors". In: *SAC 2012*. Ed. by Lars R. Knudsen and Huapeng Wu. Vol. 7707. LNCS. Springer, Heidelberg, Aug. 2013, pp. 149–165. DOI: 10.1007/978-3-642-35999-6_11.

[29] Jens Groth. "Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures". In: *ASIACRYPT 2006*. Ed. by Xuejia Lai and Kefei Chen. Vol. 4284. LNCS. Springer, Heidelberg, Dec. 2006, pp. 444–459.

[30] Jens Groth and Steve Lu. "A Non-interactive Shuffle with Pairing Based Verifiability". In: *ASIACRYPT 2007*. Ed. by Kaoru Kurosawa. Vol. 4833. LNCS. Springer, Heidelberg, Dec. 2007, pp. 51–67.

[31] Jens Groth, Rafail Ostrovsky, and Amit Sahai. "New Techniques for Noninteractive Zero-Knowledge". In: *J. ACM* 59.3 (June 2012), 11:1–11:35. ISSN: 0004-5411. DOI: 10.1145/2220357.2220358. URL: http://doi.acm.org/10.1145/2220357.2220358.

[32] Jens Groth and Amit Sahai. "Efficient Non-interactive Proof Systems for Bilinear Groups". In: *EUROCRYPT 2008*. Ed. by Nigel P. Smart. Vol. 4965. LNCS. Springer, Heidelberg, Apr. 2008, pp. 415–432.

[33] Dennis Hofheinz. "Adaptive Partitioning". In: *EUROCRYPT 2017, Part II*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10211. LNCS. Springer, Heidelberg, May 2017, pp. 489–518.

[34] Dennis Hofheinz. "Algebraic Partitioning: Fully Compact and (almost) Tightly Secure Cryptography". In: *TCC 2016-A, Part I*. Ed. by Eyal Kushilevitz and Tal Malkin. Vol. 9562. LNCS. Springer, Heidelberg, Jan. 2016, pp. 251–281. DOI: 10.1007/978-3-662-49096-9_11.

[35] Dennis Hofheinz and Tibor Jager. "Tightly Secure Signatures and Public-Key Encryption". In: *CRYPTO 2012*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. Vol. 7417. LNCS. Springer, Heidelberg, Aug. 2012, pp. 590–607.

[36] Dennis Hofheinz, Jessica Koch, and Christoph Striecks. "Identity-Based Encryption with (Almost) Tight Security in the Multi-instance, Multi-ciphertext Setting". In: *PKC 2015*. Ed. by Jonathan Katz. Vol. 9020. LNCS. Springer, Heidelberg, 2015, pp. 799–822. DOI: 10.1007/978-3-662-46447-2_36.

[37] Charanjit S. Jutla, Miyako Ohkubo, and Arnab Roy. "Improved (Almost) Tightly-Secure Structure-Preserving Signatures". In: *PKC 2018*. Springer, 2018, To appear.

[38] Charanjit S. Jutla and Arnab Roy. "Improved Structure Preserving Signatures Under Standard Bilinear Assumptions". In: *PKC 2017, Part II*. Ed. by Serge Fehr. Vol. 10175. LNCS. Springer, Heidelberg, Mar. 2017, pp. 183–209.

[39] Charanjit S. Jutla and Arnab Roy. "Switching Lemma for Bilinear Tests and Constant-Size NIZK Proofs for Linear Subspaces". In: *CRYPTO 2014, Part II*. Ed. by Juan A. Garay and Rosario Gennaro. Vol. 8617. LNCS. Springer, Heidelberg, Aug. 2014, pp. 295–312. DOI: 10.1007/978-3-662-44381-1_17.

[40] Eike Kiltz, Jiaxin Pan, and Hoeteck Wee. "Structure-Preserving Signatures from Standard Assumptions, Revisited". In: *CRYPTO 2015, Part II*. Ed. by Rosario Gennaro and Matthew J. B. Robshaw. Vol. 9216. LNCS. Springer, Heidelberg, Aug. 2015, pp. 275–295. DOI: 10.1007/978-3-662-48000-7_14.

[41] Eike Kiltz and Hoeteck Wee. "Quasi-Adaptive NIZK for Linear Subspaces Revisited". In: *EUROCRYPT 2015, Part II*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9057. LNCS. Springer, Heidelberg, Apr. 2015, pp. 101–128. DOI: 10.1007/978-3-662-46803-6_4.

[42] Kaoru Kurosawa and Yvo Desmedt. "A New Paradigm of Hybrid Encryption Scheme". In: *CRYPTO 2004*. Ed. by Matthew Franklin. Vol. 3152. LNCS. Springer, Heidelberg, Aug. 2004, pp. 426–442.

[43] Benoît Libert, Marc Joye, Moti Yung, and Thomas Peters. "Concise Multi-challenge CCA-Secure Encryption and Signatures with Almost Tight Security". In: *ASIACRYPT 2014, Part II*. Ed. by Palash Sarkar and Tetsu Iwata. Vol. 8874. LNCS. Springer, Heidelberg, Dec. 2014, pp. 1–21. DOI: 10.1007/978-3-662-45608-8_1.

[44] Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. "Compactly Hiding Linear Spans - Tightly Secure Constant-Size Simulation-Sound QA-NIZK Proofs and Applications". In: *ASIACRYPT 2015, Part I*. Ed. by Tetsu Iwata and Jung Hee Cheon. Vol. 9452. LNCS. Springer, Heidelberg, 2015, pp. 681–707. DOI: 10.1007/978-3-662-48797-6_28.

[45] Benoît Libert, Thomas Peters, and Moti Yung. "Short Group Signatures via Structure-Preserving Signatures: Standard Model Security from Simple Assumptions". In: *CRYPTO 2015, Part II*. Ed. by Rosario Gennaro and Matthew J. B. Robshaw. Vol. 9216. LNCS. Springer, Heidelberg, Aug. 2015, pp. 296–316. DOI: 10.1007/978-3-662-48000-7_15.

[46] Paz Morillo, Carla Ràfols, and Jorge Luis Villar. "The Kernel Matrix Diffie-Hellman Assumption". In: *ASIACRYPT 2016, Part I*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10031. LNCS. Springer, Heidelberg, Dec. 2016, pp. 729–758. DOI: 10.1007/978-3-662-53887-6_27.

[47] Carla Ràfols. "Stretching Groth-Sahai: NIZK Proofs of Partial Satisfiability". In: *TCC 2015, Part II*. Ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Vol. 9015. LNCS. Springer, Heidelberg, Mar. 2015, pp. 247–276. DOI: 10.1007/978-3-662-46497-7_10.