Weakly Secure Equivalence-Class Signatures from Standard Assumptions

Georg Fuchsbauer[1,2] and Romain Gay[1,2]

[1] Inria
[2] École normale supérieure, CNRS, PSL Research University, Paris, France
{fuchsbau,rgay}@di.ens.fr

Abstract. Structure-preserving signatures on equivalence classes, or equivalence-class signatures for short (EQS), are signature schemes defined over bilinear groups whose messages are vectors of group elements. Signatures are perfectly randomizable and given a signature on a vector, anyone can derive a signature on any multiple of the vector; EQS thus sign projective equivalence classes. Applications of EQS include the first constant-size anonymous attribute-based credentials, efficient round-optimal blind signatures without random oracles and efficient access-control encryption.
To date, the only existing instantiation of EQS is proven secure in the generic-group model. In this work we show that by relaxing the definition of unforgeability, which makes it efficiently verifiable, we can construct EQS from standard assumptions, namely the Matrix-Diffie-Hellman assumptions. We then show that our unforgeability notion is sufficient for most applications.

Keywords: Structure-preserving signatures on equivalence classes, standard assumptions.

1 Introduction

SPS. Structure-preserving signature (SPS) schemes [AFG⁺10] are defined over bilinear groups, which are described by three prime-order groups $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ and a bilinear map (pairing) $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Public keys, messages and signatures of SPS schemes all consist of elements from $\mathbb{G}_1$ and $\mathbb{G}_2$ and signatures are verified by comparing evaluations of pairings applied to elements of the key, the message and the signature. The primary motivation for the introduction of SPS was their smooth interoperability with the Groth-Sahai (GS) proof system [GS08], which provides efficient non-interactive zero-knowledge (NIZK) proofs proving knowledge of group elements that satisfy sets of pairing-product equations.

Together, SPS and GS proofs enable proving knowledge of signatures, keys and/or messages, and thereby modular constructions of privacy-preserving cryptographic protocols. A long line of research [AGHO11, ACD⁺12, AGOT14, BFF⁺15, AKOT15, KPW15, Gro15, Gha16, JR17, AHN⁺17] has led to schemes with improved efficiency, additional properties as well as schemes that are proven secure under standard computational hardness assumptions. Randomizable SPS

[AGHO11] allow for more efficient schemes in that parts of the signature can, after randomization, be given in the clear. However, for privacy-preserving applications, they still inherently require hiding the message and using NIZK proofs.

EQS. Structure-preserving signatures on equivalence classes, or *equivalence-class signatures* (EQS) for short, allow similar applications to SPS. Unlike the latter, they achieve them without requiring *any* NIZK proofs on top, thereby yielding more efficient schemes. Intuitively, this is because not only their signatures but also the *messages* can be randomized. Equivalence-class signatures were introduced by Hanser and Slamanig [HS14]. Their initial instantiation was only secure against random-message attacks [Fuc14], which is insufficient for the intended applications. With Fuchsbauer [FHS14] they subsequently presented a scheme that satisfies the stronger notion of unforgeability under chosen-message attacks (EUF-CMA) in the generic group model. They also strengthened the model of EQS, which later enabled further applications [FHS15].

As for regular SPS, the messages in an EQS system are vectors of group elements $[\mathbf{m}]_1 \in \mathbb{G}_1^\ell$ (which in our notation stands for $(m_1 \cdot P_1, \ldots, m_\ell \cdot P_1)$ with P_1 being a generator of $\mathbb{G}_1$). EQS provide an additional algorithm that, given a signature σ for message $[\mathbf{m}]_1$, allows to adapt σ to a signature for the message $[\mu \cdot \mathbf{m}]_1$ for any $\mu \in \mathbb{Z}_p^*$ without access to the signing key. A signature therefore actually signs all multiples of a message at once (as a signature can be adapted to any of them). In other words, signatures or on *equivalence classes* of the equivalence relation "$\sim$" on the message space $(\mathbb{G}_1^*)^\ell$ defined as $[\mathbf{m}]_1 \sim [\mathbf{n}]_1 \Leftrightarrow \exists\, s \in \mathbb{Z}_p^* : \mathbf{m} = s \cdot \mathbf{n}$.

The definition of EQS moreover requires that signatures are randomizable, in that adaptation to a new representative leads to a signature that is distributed like a fresh signature for the new representative. The DDH assumption in group $\mathbb{G}_1$ implies that given a message $[\mathbf{m}]_1 \in (\mathbb{G}_1^*)^\ell$, then $[\mu \cdot \mathbf{m}]_1$ for a random μ is indistinguishable from $[\mathbf{m}']_1$ for a random $\mathbf{m}'$. For EQS signatures DDH thus implies that given a message signature pair $([\mathbf{m}]_1, \sigma)$, an adapted signature on a random representative $([\mu \cdot \mathbf{m}]_1, \sigma')$ looks like a *fresh* signature on a *random* message.

It is the latter property that is central in applications that use EQS instead of SPS+GS-proofs. Instead of having users give (costly) zero-knowledge proofs that they possess a signature to protect their privacy, it suffices to use an EQS scheme and have the user *randomize* the message and adapt the signature every time they show it. (We discuss applications of EQS in more detail below.)

Existential unforgeability under chosen-message attacks (EUF-CMA) for EQS is defined with respect to equivalence classes: an adversary that can query signatures for messages $[\mathbf{m}_i]_1$ of its choice should be incapable of returning a signature for a message $[\mathbf{m}^*]_1$ such that $[\mathbf{m}^*]_1$ is not a multiple of any $[\mathbf{m}_i]_1$. (Note that this winning condition cannot be efficiently decided, as this would amount to breaking DDH.)

The first EQS scheme by FHS [FHS14] signs messages from $(\mathbb{G}_1^*)^\ell$ and signatures consist of 3 group elements. The authors show that this size is optimal by relying on an impossibility result [AGO11] for SPS. Security of the FHS scheme was proved directly in the generic-group model, which amounts to an interactive assumption. The same authors [FHS15] later provided a scheme from a *non*-interactive q-type assumption; the assumption is that the FHS scheme is secure against random-message attacks (where instead of a signing oracle the adversary is given signatures for q randomly chosen messages). They then build a scheme on top of the original scheme and prove EUF-CMA security. However, the signatures of their scheme are *not* randomizable, which is required for all applications of EQS.

The construction of an EQS scheme (which is randomizable) from any non-interactive assumption is still an open problem.

Applications of EQS. The first application of EQS was to anonymous (attribute-based) credentials [CL03, CL04, BCKL08, BCC$^+$09, Fuc11], yielding the first construction for which the cost of showing a credential is independent of the number of possessed, or showed, or existing attributes. In most schemes a credential is a signature from the credential-issuing authority on a message representing the user's attributes. When the user wishes to present her credential, previous constructions require her to give a zero-knowledge proof of possessing a valid signature from the organization. Using EQS [FHS14] these proofs can be avoided: the user randomizes the message of its credential by multiplying it with a random value $\mu \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$, adapts the authority's EQS signature on it and presents message and signature *in the clear*. By DDH and the properties of EQS, this pair looks like a fresh signature on a random message, which yields unlinkable user anonymity. (See Sect. 5 for more details of this construction.) Derler, Hanser and Slamanig later added the possibility of revoking users to the credential scheme [DHS15].

Fuchsbauer, Hanser and Slamanig [FHS15] used EQS to construct the first round-optimal blind signature scheme without random oracles nor CRS nor trusted setup with blindness against fully malicious signers. In order to obtain a blind signature, the user commits to her message as $[c]_1$, picks $\mu \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ and obtains an EQS signature from the signer on the randomized message $(\mu \cdot [c]_1, [\mu]_1)$. The blind signature is then an adapted signature to $([c]_1, [1]_1)$ together with an opening of $[c]_1$ to the message. While unforgeability relies on EUF-CMA of EQS and the binding property of the commitment, blindness is proved under an interactive variant of the DDH assumption.

In follow-up work [FHKS16], the authors changed the used commitment scheme from perfectly hiding to perfectly binding, which enabled them to prove blindness under a *non-interactive* (B)DDH-type assumption. (Another construction with blindness under a non-interactive assumption was also given by Hanzlik and Kluczniak [HK17].)

EQS were also used to construct verifiably encrypted signatures [HRS15] and group signatures without encryption [DS16] (see Sect. 6 for more on this).

Access-control encryption. Access-control encryption [DHO16] is a recently introduced primitive that models information flow between senders and receivers.

Whereas all forms of encryption only prevent unauthorized receivers from obtaining information, access-control encryption (ACE) additionally prevents unauthorized senders from *distributing* information.

ACE considers a relation on a set of senders and receivers that specifies who is allowed to send information to whom. To prevent unauthorized sending, all messages are sent via an authority, called the *sanitizer*, who can tweak the messages before broadcasting it; (the sanitizer should however not obtain information about sender, receiver or content of a message).

Access-control encryption was introduced by Damgård, Haagh and Orlandi [DHO16] and two papers have recently improved on it: Badertscher, Matt and Maurer [BMM17] strengthen the security model by requiring chosen-ciphertext security and non-malleability of messages, and Kim and Wu [KW17] give the first construction from standard assumptions for general policies. The only existing efficient constructions are for restricted classes of policies; the most efficient scheme is the one by Fuchsbauer, Gay, Kowalczyk and Orlandi [FGKO17] based on EQS.

In their construction any receiver Bob has a public key $[k]_1$ for ElGamal encryption. If Alice is allowed to send messages to Bob, she obtains an EQS signature σ on $([1]_1, [k]_1)$ from the authority, which serves as a certificate. When Alice wants to send a message to Bob, she first picks $s, r \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ and adapts σ to the new representative $([s]_1, s \cdot [k]_1)$. She then sends this new message/signature pair together with an ElGamal encryption $([r]_1, r \cdot [k]_1 + [m]_1)$ of her message $[m]_1$ to the sanitizer. The latter verifies the adapted signature and, if correct, picks $t \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ and sends the following to Bob: $([r]_1 + t \cdot [s]_1, [r \cdot k + m]_1 + t \cdot [s \cdot k]_1)$, which is a re-randomized encryption of $[m]_1$ under $[k]_1$.

Note that if the pair that the sanitizer uses to randomize the ciphertext was *not* a multiple of $([1]_1, [k]_1)$, then Bob would receive an encryption of a random message. Now EUF-CMA of EQS guarantees that the only way Alice can provide a signature on such a multiple is if she received a certificate for Bob's key; that is, if she is allowed to send messages to Bob. On the other hand, by the EQS randomization properties, the sanitizer does not learn anything about the intended receiver (nor the encrypted message), since $([s]_1, s \cdot [k]_1)$ and the ElGamal ciphertext look like random group elements to it.

Our contribution

In this work we present the first equivalence-class signature scheme based on standard assumptions; in particular the family of Matrix-Diffie-Hellman assumptions [EHK$^+$13], which encompasses well-known assumptions such as the decision-linear assumption [BBS04]. In order to achieve this, we need to make two modifications to the model of EQS: the first one is syntactical and the second one concerns the definition of unforgeability.

Syntax. Whereas in the original EQS model [HS14, FHS14] there is only one type of signatures, we distinguish between signatures that were output by the signing algorithm, and which can be adapted and perfectly randomized on the one hand; and signatures that have been randomized on the other. The latter type cannot be randomized any further.

We note that we are not aware of *any* applications where signatures that have been randomized need to be randomized again by an entity that does not know the original signature. In the application to credential systems, the first (randomizable) signature type would correspond to the credential that is stored by the user, whereas the second (smaller) type corresponds to *credential proofs*, that is, the object presented by the user when proving possession of a credential. For access control encryption, signatures of the first type are part of an encryption key, and randomized signatures are issued as part of a ciphertext produced from this encryption key. For the round-optimal blind signatures [FHS15], the user receives a (randomizable) signature from the signer, adapts it (to the second type) and includes it in her blind signature.

Security. We relax the notion of unforgeability considered in the original work [HS14, FHS14] and make it an *efficiently verifiable* notion. When the adversary queries its signing oracle for a signature on a message $[\mathbf{m}]$, we require it to present the discrete logarithm of its elements, that is, a query is of the form $\mathbf{m} \in (\mathbb{Z}_p^*)^\ell$. After the adversary has output a purported forgery on a message $[\mathbf{m}^*]_1$ (without giving its logarithm), the experiment can efficiently check whether it is contained in one of the classes defined by the queried messages. We call our weakened notion *existential unforgeability under chosen* open *message attacks* (EUF-CoMA).

In Sect. 4–6 we then argue that for most applications of EQS this security notion is sufficient, as constructions building on EQS either only require EUF-CoMA or they can be made to with very minor modifications. In particular, we show this for *all* applications of EQS in the literature, except for the one to round-optimal blind signatures.

Our scheme. Our scheme builds upon the affine MAC by Blazy, Kiltz and Pan [BKP14], which we first turn into a structure-preserving and "linear" MAC, that is, a MAC that allows for deriving a tag of a message $\mu \cdot [\mathbf{m}]_1$ from a tag for $[\mathbf{m}]_1 \in \mathbb{G}_1^\ell$ for any scalar $\mu \in \mathbb{Z}_p^*$. We then build upon Kiltz and Wee's [KW15] method of transforming a MAC into a signature, which has also been used in [KPW15] in the context of structure-preserving signatures. (Details on our scheme are provided in a technical overview at the beginning of Sect. 3.)

Overall, we obtain an EQS whose EUF-CoMA security is based on the bilateral variant of the DLIN assumption (where the challenge is given in both groups $\mathbb{G}_1$ and $\mathbb{G}_2$), and DDH in $\mathbb{G}_2$. More generally, we use the Matrix Decisional Diffie-Hellman (MDDH) assumption [EHK$^+$13], and its computational variant, the Kernel Matrix Diffie-Hellman (KMDH) assumption [MRV16], both of which are parameterized by a distribution of full-rank $\ell \times k$ matrices for some specified dimensions $\ell, k \in \mathbb{N}^*$, $\ell > k$, and which capture most known standard assumptions in pairing groups, such as DLIN and DDH which correspond to particular matrix distributions (of size $\ell := 3$ by $k := 2$ for DLIN, and $\ell := 2$ by $k := 1$ for DDH).

Table 1. Efficiency and security of our scheme for messages from $(\mathbb{G}_1^*)^\ell$ from general assumptions (middle) and for the most efficient setting $k := 2$, $k' := 1$ (right).

Signature size:	$\big(2k\ell + (k'+1)\big)\lvert\mathbb{G}_1\rvert + 2k\lvert\mathbb{G}_2\rvert$	$(4\ell+2)\lvert\mathbb{G}_1\rvert + 4\lvert\mathbb{G}_2\rvert$
Public key size:	$k'(k'+1+2k\ell)\lvert\mathbb{G}_2\rvert$	$(4\ell+2)\lvert\mathbb{G}_2\rvert$
Secret key size:	$2k(k'+1)\ell + 2k^2$ $\mathbb{Z}_p$ elements	$8(\ell+1)$ $\mathbb{Z}_p$ elements
Pairing to verify:	$4\ell k + k' + 1$	$8\ell + 2$
Assumption:	$\mathcal{D}_{2k,k}$-MDDH, $\mathcal{D}_{k'}$-KerMDH in $\mathbb{G}_2$	$\mathcal{D}_{4,2}$-MDDH and $\mathcal{D}_1$-KerMDH in $\mathbb{G}_2$

We adopt this matrix viewpoint for a more general and overall cleaner exposition of our construction. In particular, we give a construction that is secure under the $\mathcal{D}_{2k,k}$-MDDH assumption for any matrix distribution $\mathcal{D}_{2k,k}$ with $k \geq 2$, and the $\mathcal{D}_{k'}$-KerMDH assumption for any matrix distribution $\mathcal{D}_{k'}$ for $k' \geq 1$. We summarize the concrete efficiency of our scheme depending on the choices of k and k' in Table 1.

Concrete assumptions. Suppose we choose the matrix distribution $\mathcal{D}_{4,2}$ to be the uniform distribution $\mathcal{U}_{4,2}$ over all invertible matrices in $\mathbb{Z}_p^{4\times 2}$, and $\mathcal{D}_1$ to be the DDH distribution over $\mathbb{Z}_p^2$ defined as $\{(1,a) : a \leftarrow_{\mathrm{R}} \mathbb{Z}_p\}$. Then, $\mathcal{U}_{4,2}$-MDDH reduces to the bilateral variant of the DLIN assumption (Lemma 1), and $\mathcal{D}_1$-KerMDH in $\mathbb{G}_2$ reduces to DDH in $\mathbb{G}_2$ (Lemma 2). Thus, we obtain an EQS signature scheme whose EUF-CoMA security is based on DDH in $\mathbb{G}_2$ and bilateral DLIN (which is comparable to the original DLIN [BBS04], which was for symmetric bilinear groups).

2 Preliminaries

2.1 Notations

We denote by $x \leftarrow_{\mathrm{R}} \mathcal{B}$ the process of sampling an element x from set $\mathcal{B}$ uniformly at random. We denote by λ the security parameter, and by $\mathsf{negl}(\cdot)$ any negligible function of λ. For any $k, \ell \in \mathbb{N}^*$ such that $\ell > k$, and any matrix $\mathbf{A} \in \mathbb{Z}_p^{\ell\times k}$, we write $\mathsf{orth}(\mathbf{A}) := \{\mathbf{A}^\perp \in \mathbb{Z}_p^{\ell\times(\ell-k)} \mid \mathbf{A}^\top\mathbf{A}^\perp = \mathbf{0}$ and $\mathbf{A}^\perp$ has full rank$\}$.

2.2 Pairing Groups

Let GGen be a probabilistic polynomial-time (PPT) algorithm that on input 1^λ returns a description $\mathcal{PG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ of asymmetric bilinear groups where $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ are cyclic groups 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.

Namely, for $s \in \{1, 2, T\}$ and $a \in \mathbb{Z}_p$, we define $[a]_s = aP_s \in \mathbb{G}_s$ as the implicit representation of a in $\mathbb{G}_s$. More generally, for a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{Z}_p^{m \times n}$ we define $[\mathbf{A}]_s$ as the implicit representation of $\mathbf{A}$ in $\mathbb{G}_s$:

$$[\mathbf{A}]_s := \begin{pmatrix} a_{11}P & \dots & a_{1n}P \\ \vdots & & \vdots \\ a_{m1}P & \dots & a_{mn}P \end{pmatrix} \in \mathbb{G}_s^{m \times n}$$

Note that from $[a]_s \in \mathbb{G}_s$ it is generally hard to compute the value a (discrete logarithm problem in $\mathbb{G}_s$). Further, from $[b]_T \in \mathbb{G}_T$, it is hard to compute the value $[b]_1 \in \mathbb{G}_1$ and $[b]_2 \in \mathbb{G}_2$ (pairing inversion problem). Obviously, given $[a]_s \in \mathbb{G}_s$ and a scalar $x \in \mathbb{Z}_p$, one can efficiently compute $[ax]_s \in \mathbb{G}_s$. Further, 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 define $[\mathbf{A}]_1 \bullet [\mathbf{B}]_2 := [\mathbf{AB}]_T \in \mathbb{G}_T$, which can be computed efficiently using the pairing $e \colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

2.3 Matrix Diffie-Hellman Assumptions

We first recall the definitions of the Matrix Decisional Diffie-Hellman (MDDH) Assumption [EHK+13].

Definition 1 (Matrix Distribution). *Let $k, \ell \in \mathbb{N}^*$, such that $\ell > k$. We call $\mathcal{D}_{\ell,k}$ a matrix distribution if it outputs matrices in $\mathbb{Z}_p^{\ell \times k}$ of full rank k in polynomial time (w.l.o.g. we assume the first k rows of $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_{\ell,k}$ form an invertible matrix). We write $\mathcal{D}_k := \mathcal{D}_{k+1,k}$.*

We define a bilateral variant of the Matrix Decisional Diffie-Hellman (MDDH) assumption. Namely, [EHK+13] originally defines the $\mathcal{D}_{\ell,k}$-MDDH assumption in $\mathbb{G}_s$ for any $s \in \{1, 2, T\}$ to be distinguishing the two distributions: $([\mathbf{A}]_s, [\mathbf{Ar}]_s)$ and $([\mathbf{A}]_s, [\mathbf{u}]_s)$, whereas we use the bilateral variant which consists in distinguishing the two distributions: $([\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{Ar}]_1, [\mathbf{Ar}]_2)$ and $([\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{u}]_1, [\mathbf{u}]_2)$ where $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_{\ell,k}$, $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and $\mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^\ell$, for asymmetric pairings. Note that the bilateral variant is provably no weaker (in the generic group model) than the unilateral variant in symmetric bilinear groups. Bilateral variant of the DLIN assumption in asymmetric pairings has already been used in prior works [LPJY15, AC17].

Definition 2 ($\mathcal{D}_{\ell,k}$-Matrix Decisional Diffie-Hellman Assumption ($\mathcal{D}_{\ell,k}$-MDDH)). *Let $\lambda, k, \ell \in \mathbb{N}^*$ such that $\ell > k \geq 2$, and let $\mathcal{D}_{\ell,k}$ be a matrix distribution. We say that the $\mathcal{D}_{\ell,k}$-Matrix Decisional Diffie-Hellman ($\mathcal{D}_{\ell,k}$-MDDH) Assumption holds relative to* GGen *if for all PPT adversaries $\mathcal{A}$,*

$$\begin{aligned} \mathsf{Adv}^{\mathrm{MDDH}}_{\mathsf{GGen},\mathcal{D}_{\ell,k},\mathcal{A}}(\lambda) := \big| \Pr\left[\mathcal{A}\left(\mathcal{PG}, \{[\mathbf{A}]_s, [\mathbf{Ar}]_s\}_{s \in \{1,2\}}\right) = 1\right] \\ - \Pr\left[\mathcal{A}\left(\mathcal{PG}, \{[\mathbf{A}]_s, [\mathbf{u}]_s\}_{s \in \{1,2\}}\right) = 1\right] \big| = \mathsf{negl}(\lambda) \ , \end{aligned}$$

where the probability is taken over $\mathcal{PG} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2) \leftarrow_{\mathrm{R}}$ $\mathsf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k, \mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k, \mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^\ell$.

Let $Q \geq 1$. For $\mathbf{W} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k\times Q}, \mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{\ell\times Q}$, we consider the Q-fold $\mathcal{D}_{\ell,k}$-MDDH Assumption in $\mathbb{G}_s$, which consists in distinguishing the distributions $\{[\mathbf{A}]_s, [\mathbf{AW}]_s\}_{s\in\{1,2\}}$ from $\{[\mathbf{A}]_s, [\mathbf{U}]_s\}_{s\in\{1,2\}}$. That is, a challenge for the Q-fold $\mathcal{D}_{\ell,k}$-MDDH assumption consists of Q independent challenges of the $\mathcal{D}_k$-MDDH assumption (with the same $\mathbf{A}$ but different randomness $\mathbf{w}$).

Definition 3 (Uniform distribution). *Let $k, \ell \in \mathbb{N}^*$, with $\ell > k$. We denote by $\mathcal{U}_{\ell,k}$ the uniform distribution over all full-rank $\ell \times k$ matrices over $\mathbb{Z}_p$.*

Among all possible matrix distributions $\mathcal{D}_{\ell,k}$, the uniform matrix distribution $\mathcal{U}_{\ell',k}$ is the hardest possible instance.

Lemma 1 ($\mathcal{D}_{\ell,k}$-MDDH implies Q-fold $\mathcal{U}_{\ell',k}$-MDDH ([EHK+13, GHKW16])). *Let $k, \ell, \ell', Q \in \mathbb{N}^*$, such that $\ell > k$, $\ell' > k$, and let $\mathcal{D}_{\ell,k}$ be a matrix distribution. For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that $\mathsf{Adv}^{Q\text{-MDDH}}_{\mathsf{GGen},\mathcal{U}_{\ell',k},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{MDDH}}_{\mathsf{GGen},\mathcal{D}_{\ell,k},\mathcal{B}}(\lambda) + \frac{1}{p-1}$.*

Now we recall the definition of the $\mathcal{D}_k$-Kernel Matrix Decisional Diffie-Hellman assumption [MRV16], a natural computational analog of the $\mathcal{D}_k$-Matrix Decisional Diffie-Hellman assumption.

Definition 4 ($\mathcal{D}_k$-Kernel Matrix Diffie-Hellman ($\mathcal{D}_k$-KerMDH) assumption [MRV16]). *Let $\lambda, k \in \mathbb{N}^*$, and $\mathcal{D}_k$ be a matrix distribution. We say that the $\mathcal{D}_k$-Kernel Matrix Diffie-Hellman ($\mathcal{D}_k$-KerMDH) assumption holds relative to GGen in $\mathbb{G}_s$ for $s \in \{1,2\}$, if for all PPT adversaries $\mathcal{A}$,*

$$\mathsf{Adv}^{\mathrm{KerMDH}}_{\mathsf{GGen},\mathcal{D}_k,\mathbb{G}_s,\mathcal{A}}(\lambda) := \Pr\left[\mathbf{c} \in \mathsf{orth}(\mathbf{A}) \,\middle|\, [\mathbf{c}]_{3-s} \leftarrow \mathcal{A}(\mathcal{PG}, [\mathbf{A}]_s)\right] = \mathsf{negl}(\lambda) \ ,$$

where the probability is taken over $\mathcal{PG} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2) \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda)$, and $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k$.

Note that the winning condition is efficiently checkable using the pairing: $\mathbf{c} \in \mathsf{orth}(\mathbf{A}) \Leftrightarrow e([\mathbf{A}]_s, [\mathbf{c}]_{3-s}) = [\mathbf{0}]_T$.

For any matrix distribution $\mathcal{D}_k$, the $\mathcal{D}_k$-KerMDH assumption is weaker than its decisional counterpart:

Lemma 2 ($\mathcal{D}_k$-MDDH $\Rightarrow$ $\mathcal{D}_k$-KerMDH [MRV16]). *Let $k \in \mathbb{N}^*$, and let $\mathcal{D}_k$ be a matrix distribution. For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that $\mathsf{Adv}^{\mathrm{KerMDH}}_{\mathsf{GGen},\mathcal{D}_k,\mathbb{G}_s,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{MDDH}}_{\mathsf{GGen},\mathcal{D}_k,\mathbb{G}_s,\mathcal{B}}(\lambda)$.*

2.4 Equivalence-Class Signatures

We recall the definition of *structure preserving signatures on equivalence classes* from [FHS14], which we call equivalence-class signatures for short.

Let us denote $\mathsf{Span}([\mathbf{m}]_1) := \{[\mu\cdot\mathbf{m}]_1 \,|\, \mu \in \mathbb{Z}_p\}$ and $(\mathbb{G}_1^\ell)^* := \mathbb{G}_1^\ell \backslash \{[\mathbf{0}]_1 \in \mathbb{G}_1^\ell\}$. Let $\lambda, \ell \in \mathbb{N}^*$ and $\mathcal{PG} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ be an output of $\mathsf{GGen}(1^\lambda)$. An EQS scheme signs an equivalence class $\mathsf{Span}([\mathbf{m}]_1)$ for $[\mathbf{m}]_1 \in (\mathbb{G}_1^\ell)^*$, and it allows to derive from a signature for $[\mathbf{m}]_1$ a fresh signature for any vector in $\mathsf{Span}([\mathbf{m}]_1)$ without access to the secret key.

Our definition slightly differs from that of [FHS14], as we make a syntactical difference between signatures output by the signing algorithm, which can be re-randomized, and (final) signatures that have been re-randomized and cannot be re-randomized again. In [FHS14], these are the same object, but in our scheme, re-randomizable signatures are vectors of group elements of different dimension than final signatures. We call the re-randomizable signature pre-signature, and final signatures simply signatures. Note that the re-randomizability is crucial to obtain signature adaptation, defined below.

Definition 5 (EQS). *An equivalence-class signature scheme consists of the following PPT algorithms:*

- $\mathsf{Setup}(\mathcal{PG})$*, on input a pairing group* $\mathcal{PG} \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda)$*, outputs a secret key* sk *and a public key* pk*, which implicitly defines a pre-signature space* $\mathcal{R}$ *and a signature space* $\mathcal{S}$*.*
- $\mathsf{Sign}(sk, [\mathbf{m}]_1 \in (\mathbb{G}_1^\ell)^*)$*, on input the secret key* sk *and a representative* $[\mathbf{m}]_1 \in (\mathbb{G}_1^\ell)^*$ *of the equivalence class* $\mathsf{Span}([\mathbf{m}]_1)$*, outputs a pre-signature* ρ *that is valid for the representative* $[\mathbf{m}]_1$*.*
- $\mathsf{Adapt}(pk, \rho \in \mathcal{R}, \mu \in \mathbb{Z}_p^*)$*, on input the public key* pk*, a pre-signature* $\rho \in \mathcal{R}$*, and a scalar* $\mu \in \mathbb{Z}_p^*$*, outputs an updated signature* $\sigma \in \mathcal{S}$ *for the same equivalence class. If* ρ *is valid for representative* $[\mathbf{m}]_1$*, then* σ *is valid for representative* $[\mu \cdot \mathbf{m}]_1$ *of the same equivalence class.*
- $\mathsf{Ver}(pk, [\mathbf{m}]_1 \in (\mathbb{G}_1^\ell)^*, \sigma \in \mathcal{S})$*, on input the public key* pk*,* $[\mathbf{m}]_1 \in (\mathbb{G}_1^\ell)^*$*, and a signature* $\sigma \in \mathcal{S}$*, outputs* 1 *if the signature is valid for* $[\mathbf{m}]_1$ *under* pk*, and* 0 *otherwise.*
- $\mathsf{VerKey}(sk, pk)$*, is a deterministic algorithm that on input the secret key* sk *and the public key* pk *checks their consistency and outputs* 1 *in case the check is successful,* 0 *otherwise.*

Although there is no algorithm to verify pre-signatures, one can easily do so by first adapting them using $\mu := 1$ and then applying Ver to the result.

Remark 1. We note that we are not aware of any application of EQS where a signature that has been re-randomized by some entity A needs to be re-randomized again by another entity B. (Even in the application to blind signatures in [FHS15], the user only needs to adapt once.) In such applications, having signatures of different types would pose a problem.

EQS schemes that adhere to the type above can thus be used in all applications: a user obtains a (pre-)signature of type $\mathcal{R}$ and uses it to derive randomizations (and possibly adaptations to other messages) from it.

Correctness. An EQS $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{Adapt}, \mathsf{Ver}, \mathsf{VerKey})$ satisfies correctness if the following hold:

- $\Pr[\mathsf{VerKey}(sk, pk) = 1]$, where the probability is taken over $(sk, pk) \leftarrow_{\mathrm{R}} \mathsf{Setup}(1^\lambda)$;

- for all $[\mathbf{m}]_1 \in (\mathbb{G}_1^\ell)^*$, $\mu \in \mathbb{Z}_p^*$: $\Pr[\mathsf{Ver}(pk, [\mu \cdot \mathbf{m}]_1, \sigma) = 1] = 1$, where the probability is taken over $(sk, pk) \leftarrow \mathsf{Setup}(1^\lambda)$, $\rho \leftarrow \mathsf{Sign}(sk, [\mathbf{m}]_1)$, $\sigma \leftarrow_{\mathrm{R}} \mathsf{Adapt}(pk, \rho, \mu)$.

We define a new unforgeability notion, which is weaker than the original EUF-CMA security definition from [FHS14] (which we restate below for completeness), but still suffices for many applications, as we show in Sect. 5 and 6. An advantage of our new definition is that it is efficiently decidable whether the adversary has won the security game, contrary to EUF-CMA as originally defined.

EUF-CMA. An EQS scheme $\mathsf{EQS} := (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Adapt}, \mathsf{Ver}, \mathsf{VerKey})$ satisfies existential unforgeability under chosen-message attacks (EUF-CMA) if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{EQS},\mathcal{A}}(\lambda) := \Pr\left[\mathsf{Exp}^{\text{EUF-CMA}}_{\mathsf{EQS}}(1^\lambda, \mathcal{A}) = 1\right] = \mathsf{negl}(\lambda) ,$$

with game $\mathsf{Exp}^{\text{EUF-CMA}}_{\mathsf{EQS}}(1^\lambda, \mathcal{A})$ defined as follows:

Game Definition	**Oracle Definition**
$\underline{\mathsf{Exp}^{\text{EUF-CMA}}_{\mathsf{EQS}}(1^\lambda, \mathcal{A})\text{:}}$	$\underline{\mathsf{SignO}([\mathbf{m}]_1)\text{:}}$
$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$	$\mathcal{Q}_{\mathsf{sign}} := \mathcal{Q}_{\mathsf{sign}} \cup \{[\mathbf{m}]_1\}$
$\mathcal{Q}_{\mathsf{sign}} := \emptyset$	$\rho := \mathsf{Sign}(sk, [\mathbf{m}]_1)$
$(sk, pk) \leftarrow_{\mathrm{R}} \mathsf{Setup}(\mathcal{PG})$	Return ρ.
$([\mathbf{m}^\star]_1, \sigma^\star) \leftarrow \mathcal{A}^{\mathsf{SignO}(\cdot)}(pk)$	
Return 1 iff $\mathsf{Ver}(pk, [\mathbf{m}^\star]_1, \sigma^\star) = 1$ and	
$[\mathbf{m}^\star]_1 \notin \bigcup_{[\mathbf{m}]_1 \in \mathcal{Q}_{\mathsf{sign}}} \mathsf{Span}([\mathbf{m}]_1)$.	

It is still an open problem to construct an EQS scheme that achieves standard EUF-CMA under standard assumptions.

We now state our new notion, which we call *Existential UnForgeability under Chosen Open Message Attacks* (EUF-CoMA). The only difference to EUF-CMA isthat the adversary has to give the discrete logarithm of messages $\mathbf{m} \in \mathbb{Z}_p^\ell$ instead of $[\mathbf{m}]_1$ to SignO.

EUF-CoMA. An EQS scheme $\mathsf{EQS} := (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Adapt}, \mathsf{Ver}, \mathsf{VerKey})$ satisfies existential unforgeability under chosen open message attacks if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\text{EUF-CoMA}}_{\mathsf{EQS},\mathcal{A}}(\lambda) := \Pr\left[\mathsf{Exp}^{\text{EUF-CoMA}}_{\mathsf{EQS}}(1^\lambda, \mathcal{A}) = 1\right] = \mathsf{negl}(\lambda) ,$$

with game $\mathsf{Exp}^{\text{EUF-CoMA}}_{\mathsf{EQS}}(1^\lambda, \mathcal{A})$ defined as follows:

Game Definition	**Oracle Definition**
$\underline{\mathsf{Exp}^{\text{EUF-CoMA}}_{\mathsf{EQS}}(1^\lambda, \mathcal{A})}$:	$\underline{\mathsf{SignO}(\mathbf{m})}$:
$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$	$\mathcal{Q}_{\mathsf{sign}} := \mathcal{Q}_{\mathsf{sign}} \cup \{\mathbf{m}\}$
$\mathcal{Q}_{\mathsf{sign}} := \emptyset$	$\rho := \mathsf{Sign}(sk, [\mathbf{m}]_1)$
$(sk, pk) \leftarrow_{\mathrm{R}} \mathsf{Setup}(\mathcal{PG})$	Return ρ.
$([\mathbf{m}^\star]_1, \sigma^\star) \leftarrow \mathcal{A}^{\mathsf{SignO}(\cdot)}(pk)$	
Return 1 iff $\mathsf{Ver}(pk, [\mathbf{m}^\star]_1, \sigma^\star) = 1$ and	
$[\mathbf{m}^\star]_1 \notin \bigcup_{\mathbf{m} \in \mathcal{Q}_{\mathsf{sign}}} \mathsf{Span}([\mathbf{m}]_1)$.	

Remark 2 (Decidability of breaks). As opposed to the original definition [FHS14], our variant allows to efficiently check whether an adversary has won, since for all $\mathbf{m} \in \mathcal{Q}_{\mathsf{sign}}$, one can efficiently check whether $[\mathbf{m}^\star]_1 \in \mathsf{Span}([\mathbf{m}]_1)$ (or equivalently $\det[\mathbf{m}^\star \| \mathbf{m}]_1 = 0$) when given $\mathbf{m} \in \mathbb{Z}_p^\ell$ directly as follows: check whether for some $i \in [\ell]$: $m_i \cdot [m_1^\star]_1 \neq m_1 \cdot [m_i^\star]_1$.

Signature-Adaptation. An scheme $\mathsf{EQS} := (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Adapt}, \mathsf{Ver}, \mathsf{VerKey})$ perfectly adapts signatures if for all $(sk, pk, [\mathbf{m}]_1, \mu)$ with

$$\mathsf{VerKey}(sk, pk) = 1,\ [\mathbf{m}]_1 \in (\mathbb{G}_1^\ell)^*,\ \mu \in \mathbb{Z}_p^*,$$

the following are identically distributed:

$$\Big(\rho := \mathsf{Sign}(sk, [\mathbf{m}]_1), \mathsf{Adapt}(pk, \rho, \mu)\Big) \text{ and}$$
$$\Big(\rho := \mathsf{Sign}(sk, [\mathbf{m}]_1), \mathsf{Adapt}(pk, \mathsf{Sign}(sk, [\mu \cdot \mathbf{m}]_1), 1)\Big) .$$

3 EQS from Standard Assumptions

In this section we present our EQS scheme and prove it secure under the Matrix Diffie-Hellman assumption.

Overview of the construction. We first build a private-key variant of EQS, that is, a MAC on equivalence classes. Our starting point is a modification of the affine MAC by Blazy, Kiltz and Pan [BKP14, Section 3.3], which we make linear instead of affine. This then allows anyone to multiply the tag of $[\mathbf{m}]_1 \in \mathbb{G}_1^\ell$ to obtain a tag of any vector in $\mathsf{Span}([\mathbf{m}]_1)$. We start with recalling the MAC from [BKP14], which is based on the $\mathcal{D}_k$-MDDH assumption:

$$\text{BKP:}\quad sk := \big(\mathbf{k}_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1}, \mathbf{K}_1 \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{\ell \times (k+1)}, \mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k\big)$$
$$\mathsf{Tag}(sk, [\mathbf{m}]_1) := \big([(\mathbf{k}_0^\top + \mathbf{m}^\top \mathbf{K}_1)\mathbf{t}]_1, [\mathbf{t}]_1 := [\mathbf{A}\mathbf{u}]_1\big) \text{ with } \mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k.$$

A first idea to make this MAC an "equivalence-class MAC" would be to omit $\mathbf{k}_0$:

$$\begin{aligned}\text{First attempt: } & sk := \big(\mathbf{K} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{\ell\times(k+1)}, \mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k\big) \\ & \mathsf{Tag}(sk, [\mathbf{m}]_1) := \big([\mathbf{m}^\top \mathbf{K}\mathbf{t}]_1, [\mathbf{t}]_1 := [\mathbf{A}\mathbf{u}]_1\big) \text{ with } \mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k.\end{aligned}$$

Note that now it suffices to multiply $[\mathbf{m}^\top \mathbf{K}\mathbf{t}]_1$ by any scalar $\mu \in \mathbb{Z}_p^*$ to obtain a tag for $[\mu \cdot \mathbf{m}]_1$. One problem with this first attempt though is correctness: our goal is a structure-preserving MAC, where the verification takes as input a message $[\mathbf{m}]_1 \in \mathbb{G}_1^\ell$, and not $\mathbf{m} \in \mathbb{Z}_p^\ell$, as for BKP's MAC. Thus, we put the vector $[\mathbf{t}]_2$ in source group $\mathbb{G}_2$, and a tag $\tau := ([t_0]_1, [\mathbf{t}]_2)$ is considered valid for message $[\mathbf{m}]_1$ if $[\mathbf{m}^\top]_1 \mathbf{K} \bullet [\mathbf{t}]_2 = [t_0]_1 \bullet [1]_2$, where the product "$\bullet$" is computed using the pairing $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Note that this change requires to use the $\mathcal{D}_k$-MDDH assumption for $k \geq 2$, for instance DLIN, which allows to switch vectors $\{[\mathbf{A}\mathbf{u}]_s\}_{s\in\{1,2\}}$ given in *both* source groups $\mathbb{G}_1$ and $\mathbb{G}_2$ to uniformly random over these groups.

Still, we run into another problem when reducing the unforgeability of the MAC to MDDH: the reduction needs to compute tags for messages $[\mathbf{m}]_1$, given an MDDH challenge $\{[\mathbf{t}]_s\}_{s\in\{1,2\}}$. This is not possible, since each tag contains $[\mathbf{m}^\top \mathbf{K}\mathbf{t}]_1$, in source group $\mathbb{G}_1$. One solution is to put the latter in the target group: $[\mathbf{m}^\top \mathbf{K}\mathbf{t}]_T$ (note that correctness is maintained since we can simply check $[\mathbf{m}^\top]_1 \mathbf{K} \bullet [\mathbf{t}]_2 = [\mathbf{m}^\top \mathbf{K}\mathbf{t}]_T$), thereby allowing the reduction to simulate tags. However, looking ahead, this will pose problems when going from MAC to signature, since for signatures the public key contains group elements and not $\mathbb{Z}_p$ elements.

Another solution is to require the adversary against unforgeability of the MAC to know the discrete logarithm of its challenge messages, that is, the signing oracle takes as input $\mathbf{m} \in \mathbb{Z}_p^\ell$ instead of $[\mathbf{m}]_1 \in \mathbb{G}_1^\ell$ (cf. Def. 5). This way, the reduction (from unforgeability to MDDH), given $\mathbf{m} \in \mathbb{Z}_p^\ell$ and its MDDH challenge $\{[\mathbf{t}]_s\}_{s\in\{1,2\}}$, can compute tags.

$$\begin{aligned}\text{Successful attempt for MAC: } & sk := \big(\mathbf{K} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{\ell\times(k+1)}, \mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k \text{ for } k \geq 2\big) \\ & \mathsf{Tag}(sk, [\mathbf{m}]_1) := \big([\mathbf{m}^\top \mathbf{K}\mathbf{t}]_1, [\mathbf{t}]_2 := [\mathbf{A}\mathbf{u}]_2\big) \text{ with } \mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k.\end{aligned}$$

In order to transform the MAC into a signature, we use techniques similar to those used by Kiltz and Wee [KW15]. We first write the key $\mathbf{K}^\top := (\mathbf{k}_1 \| \cdots \| \mathbf{k}_\ell) \in \mathbb{Z}_p^{(k+1)\times\ell}$, and any tag for a message $[\mathbf{m}]_1$, as

$$\mathsf{Tag}(sk, [\mathbf{m}]_1) := \Big(\Big[\textstyle\sum_{i\in[\ell]} m_i \mathbf{k}_i^\top \mathbf{t}\Big]_1, [\mathbf{t}]_2\Big) \ .$$

Then we carry out the transformation $\mathbf{k}_i \in \mathbb{Z}_p^{k+1} \to \mathbf{K}_i \in \mathbb{Z}_p^{(k+1)\times(k'+1)}$, which allows us to publish $([\mathbf{B}]_2, \{[\mathbf{K}_i\mathbf{B}]_2\}_{i\in[\ell]})$ as the public key, where $\mathbf{B} \leftarrow_{\mathrm{R}} \mathcal{D}_{k'}$. To prove security, we first argue that the $\mathbf{K}_i$ have some entropy that is computationally inaccessible from $[\mathbf{K}_i\mathbf{B}]_2$, based on the KerMDH assumption with respect to $[\mathbf{B}]_2$. That entropy is then used to perform the security proof of the

```
$\underline{\mathsf{Setup}(\mathcal{PG})\text{:}}$
$\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_{2k,k},\ \mathbf{B} \leftarrow_{\mathrm{R}} \mathcal{D}_{k'},$ for all $i \in [\ell]$: $\mathbf{K}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2k \times (k'+1)}$
Return $pk := \big([\mathbf{B}]_2, \{[\mathbf{K}_i\mathbf{B}]_2\}_{i \in [\ell]}\big)$ and $sk := \big(\mathbf{A}, \{\mathbf{K}_i\}_{i \in [\ell]}\big)$

$\underline{\mathsf{Sign}(sk, [\mathbf{m}]_1 \in (\mathbb{G}_1^\ell)^*)\text{:}}$
$\mathbf{U} \leftarrow_{\mathrm{R}} \mathsf{GL}_k,\ \mathbf{S} := \mathbf{AU},$ for all $i \in [\ell]$: $[\mathbf{S}_i]_1 := [m_i]_1\mathbf{S},\ [\mathbf{S}_{\ell+1}]_1 := \sum_{i=1}^{\ell}[m_i]_1\mathbf{K}_i^\top\mathbf{S}$
Return $\rho := \big(\{[\mathbf{S}_i]_1\}_{i \in [\ell+1]}, [\mathbf{S}]_2\big)$

$\underline{\mathsf{Adapt}(pk, \rho := \big(\{[\mathbf{S}_i]_1\}_{i \in [\ell+1]}, [\mathbf{S}]_2\big), \mu \in \mathbb{Z}_p^*)\text{:}}$
$\mathbf{r} \leftarrow_{\mathrm{R}} (\mathbb{Z}_p^k)^*,\ [\mathbf{s}]_2 := [\mathbf{S}]_2\mathbf{r},$ for all $i \in [\ell+1]$: $[\mathbf{s}_i]_1 := \mu[\mathbf{S}_i]_1\mathbf{r}$
Return $\sigma := \big(\{[\mathbf{s}_i]_1\}_{i \in [\ell+1]}, [\mathbf{s}]_2\big)$

$\underline{\mathsf{Ver}\big(pk, [\mathbf{m}]_1, \sigma := \big(\{[\mathbf{s}_i]_1\}_{i \in [\ell+1]}, [\mathbf{s}]_2\big)\big)\text{:}}$
Return 1 if the followings conditions are true:
  ○ $[\mathbf{s}]_2 \neq [\mathbf{0}]_2$
  ○ $\forall i \in [\ell] : [\mathbf{s}_i]_1 \bullet [1]_2 = [m_i]_1 \bullet [\mathbf{s}]_2$
  ○ $\sum_{i=1}^{\ell}[\mathbf{s}_i^\top]_1 \bullet [\mathbf{K}_i\mathbf{B}]_2 = [\mathbf{s}_{\ell+1}^\top]_1 \bullet [\mathbf{B}]_2$
Return 0 otherwise
```

Fig. 1. EQS scheme that satisfies EUF-CoMA based on the $\mathcal{D}_{2k,k}$-MDDH (for $k \geq 2$) and $\mathcal{D}_{k'}$-KerMDH (for $k' \geq 1$) assumptions.

private-key variant of our scheme. To make signatures verifiable, we include the vectors $[m_i\mathbf{t}]_1$ for all $i \in [\ell]$ as part of the signature, and verify them as follows:

$$sk := \big(\{\mathbf{K}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times(k'+1)}\}_{i \in [\ell]}, \mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k \text{ for } k \geq 2\big)$$
$$pk := \big([\mathbf{B}]_2, \{[\mathbf{K}_i\mathbf{B}]_2\}_{i \in [\ell]}\big)$$
$$\sigma := \Big(\Big[\textstyle\sum_{i \in [\ell]} m_i\mathbf{K}_i^\top\mathbf{t}\Big]_1, \{[m_i\mathbf{t}]_1\}_{i \in [\ell]}, [\mathbf{t}]_2\Big) \text{ where } \mathbf{t} := \mathbf{Au}, \text{ and } \mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$$
$$\mathsf{Ver}(pk, [\mathbf{m}]_1, \sigma) : \text{ checks } \textstyle\sum_{i=1}^{\ell}[m_i\mathbf{t}^\top]_1 \bullet [\mathbf{K}_i\mathbf{B}]_2 = \Big[\sum_{i \in [\ell]} m_i\mathbf{t}^\top\mathbf{K}_i\Big]_1 \bullet [\mathbf{B}]_2.$$

Note that the verification also needs to check that the $[m_i\mathbf{t}]_1$ are consistent with $[\mathbf{t}]_2$ and $[\mathbf{m}]_1$, that is, for all $i \in [\ell]$: $[m_i\mathbf{t}]_1 \bullet [1]_2 = [m_i]_1 \bullet [\mathbf{t}]_2$, and that $[\mathbf{t}]_2 \neq [\mathbf{0}]_2$, to avoid trivial forgeries. As for the MAC, it is easy to change a signature for $[\mathbf{m}]_1$ to a signature for $[\mu \cdot \mathbf{m}]_1$ for any $\mu \in \mathbb{Z}_p^*$, only knowing pk.

Finally, we want to make it possible to re-randomize signatures (so that it is impossible to trace back the original signature from a fresh one): we change $[\mathbf{t}]_2 := [\mathbf{Au}]_2$ to $[\mathbf{S}]_2 := [\mathbf{AU}]_2$, where $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_k$, $\mathbf{u} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$, and $\mathbf{U} \leftarrow_{\mathrm{R}} \mathsf{GL}_k$. Here, GL_k denotes all invertible matrices in $\mathbb{Z}_p^{k \times k}$. This way, a fresh MDDH vector can be obtained by multiplying $[\mathbf{S}]_2$ by a random vector $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$.

For technical reasons (which we explain in step "$\mathsf{Game}_{i.3}$ to $\mathsf{Game}_{i.4}$" on page 20) we actually require a matrix distribution $\mathcal{D}_{2k,k}$ for $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_{2k,k}$, with $k \geq 2$, instead of $\mathcal{D}_k$. The size of the matrices $\mathbf{K}_i$ needs to be changed accordingly. Our scheme is given in Fig. 1.

Comparison with linearly homomorphic SPS. Note that the linear homomorphism property of our signatures is limited to produce signatures in the same equivalence class. In particular, when Sign is invoked first on $[\mathbf{m}]_1 \in (\mathbb{G}_1^\ell)^*$, then on another input $[\mathbf{m}']_1 \in (\mathbb{G}_1^\ell)^*$ it produces a signature with fresh randomness, that cannot be combined with the signature generated previously on input $[\mathbf{m}]_1$. In that respect, EQS differ from linearly homomorphic structure-preserving signatures, such as those from [KPW15].

Theorem 1 (EUF-CoMA). *If the $\mathcal{D}_{2k,k}$-MDDH and $\mathcal{D}_{k'}$-KerMDH assumptions hold relative to* GGen*, then the EQS scheme in Fig. 1 satisfies EUF-CoMA. In particular, for any PTT adversary $\mathcal{A}$, there exist PPT adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ such that:*

$$\begin{aligned}&\mathsf{Adv}^{\text{EUF-CoMA}}_{\mathsf{EQS},\mathcal{A}}(\lambda)\\ &\qquad\leq \mathsf{Adv}^{\text{KerMDH}}_{\mathsf{GGen},\mathcal{D}_{k'},\mathcal{B}_1}(\lambda) + 2Q_{\mathsf{Sign}} \cdot \mathsf{Adv}^{\text{MDDH}}_{\mathsf{GGen},\mathcal{D}_{2k,k},\mathcal{B}_2}(\lambda) + \frac{3kQ_{\mathsf{Sign}}+1}{p}\ .\end{aligned}$$

Proof of Theorem 1. We use hybrids Game_1 through Game_3 defined in Fig. 2. We denote by Adv_i the advantage of $\mathcal{A}$ in Game_i, that is $\Pr[\mathsf{Game}_i(1^\lambda, \mathcal{A}) = 1]$, where the probability is taken over the random coins of Game_i and $\mathcal{A}$. Note that Game_0 is $\mathsf{Exp}^{\text{EUF-CoMA}}_{\mathsf{EQS}}(1^\lambda, \mathcal{A})$.

From Game_0 to Game_1: We change the verification oracle, using the $\mathcal{D}_{k'}$-KerMDDH assumption on $[\mathbf{B}]_2$. A pair $\big([\mathbf{m}]_1, \sigma = (\{[\mathbf{s}_i]_1\}_{i\in[\ell+1]}, [\mathbf{s}]_2)\big)$ that passes VerO in Game_0 but not in Game_1 is such that $\big(\sum_{i=1}^{\ell} \mathbf{s}_i^\top \mathbf{K}_i - \mathbf{s}_{\ell+1}^\top\big)\mathbf{B} = 0$, and $\big(\sum_{i=1}^{\ell} \mathbf{s}_i^\top \mathbf{K}_i - \mathbf{s}_{\ell+1}^\top\big) \neq \mathbf{0}^\top$. We can thus build a PPT algorithm $\mathcal{B}_1$ such that:

$$|\mathsf{Adv}_0 - \mathsf{Adv}_1| \leq \mathsf{Adv}^{\text{KerMDH}}_{\mathsf{GGen},\mathcal{D}_{k'},\mathcal{B}_1}(\lambda)$$

as follows. $\mathcal{B}_1$ gets a challenge $[\mathbf{B}]_2$ for $\mathbf{B} \leftarrow_{\mathrm{R}} \mathcal{D}_{k'}$, picks $\mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_{2k,k}$ and $\mathbf{K}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2k\times(k'+1)}$ with which it simulates $\mathcal{A}$'s view. When $\mathcal{A}$ outputs a forgery $\Big([\mathbf{m}]_1, \sigma := \big(\{[\mathbf{s}_i]_2\}_{i\in[\ell+1]}, [\mathbf{s}]_1\big)\Big)$, $\mathcal{B}$ computes and returns $\sum_{i=1}^{\ell}[\mathbf{s}_i^\top]_1\mathbf{K}_i - [\mathbf{s}_{\ell+1}^\top]_1$, which breaks the KerMDDH assumption whenever Game_0 and Game_1 differed.

From Game_1 to Game_2: These two games are in fact equivalently distributed: for all $\mathbf{k}_i \in \mathbb{Z}_p^{2k}$, $\mathbf{b}^\perp \in \mathsf{orth}(\mathbf{B})$, the two following distributions are the same:

$$\mathbf{K}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2k\times(k'+1)} \quad \text{and} \quad \mathbf{K}_i + \boxed{\mathbf{k}_i\mathbf{b}^\perp}, \text{ with } \mathbf{K}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2k\times(k'+1)}\ .$$

Now any occurrence of $\mathbf{K}_i$ in Game_1 is replaced by $\mathbf{K}_i + \mathbf{k}_i\mathbf{b}^\perp$ in Game_2. Note that the extra term $\mathbf{k}_i\mathbf{b}^\perp$ does not appear in pk, since $(\mathbf{K}_i + \mathbf{k}_i\mathbf{b}^\perp)\mathbf{B} = \mathbf{K}_i\mathbf{B}$.

```
$\mathsf{Game}_0,\ \boxed{\mathsf{Game}_1,\ \colorbox{lightgray}{$\mathsf{Game}_2$}}$:
$\mathcal{Q}_{\mathsf{sign}} := \emptyset,\ \mathbf{A} \leftarrow_{\mathrm{R}} \mathcal{D}_{2k,k},\ \mathbf{B} \leftarrow_{\mathrm{R}} \mathcal{D}_{k'},\ \colorbox{lightgray}{pick $\mathbf{b}^{\perp} \in \mathsf{orth}(\mathbf{B})$}$
for all $i \in [\ell]$: $\mathbf{K}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2k \times (k'+1)},\ \colorbox{lightgray}{$\mathbf{k}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1}$}$
$pk := \big([\mathbf{B}]_2, \{[\mathbf{K}_i\mathbf{B}]_2\}_{i\in[\ell]}\big)$
$([\mathbf{m}^\star]_1, \sigma^\star) \leftarrow \mathcal{A}^{\mathsf{SignO}(\cdot)}(pk)$
Return 1 if $\mathsf{VerO}([\mathbf{m}^\star]_1, \sigma^\star) = 1$ and $[\mathbf{m}^\star]_1 \notin \bigcup_{\mathbf{m}\in\mathcal{Q}_{\mathsf{sign}}} \mathsf{Span}([\mathbf{m}]_1)$
Return 0 otherwise

$\mathsf{VerO}\Big([\mathbf{m}^\star]_1, \sigma^\star := \big(\{[\mathbf{s}_i]_1\}_{i\in[\ell+1]}\big), [\mathbf{s}]_2\Big)$:
  Return 1 if the following conditions are true:
    ○ $[\mathbf{s}]_2 \neq [\mathbf{0}]_2$
    ○ $\forall i \in [\ell]$: $[\mathbf{s}_i]_1 \bullet [1]_2 = [m_i^\star]_1 \bullet [\mathbf{s}]_2$
    ○ $\sum_{i=1}^{\ell} [\mathbf{s}_i^\top]_1 \bullet [\mathbf{K}_i\mathbf{B}]_2 = [\mathbf{s}_{\ell+1}^\top]_1 \bullet [\mathbf{B}]_2$
        $\boxed{\text{and } \sum_{i=1}^{\ell} [\mathbf{s}_i^\top]_1 \big(\mathbf{K}_i + \colorbox{lightgray}{$\mathbf{k}_i(\mathbf{b}^\perp)^\top$}\big) = [\mathbf{s}_{\ell+1}^\top]_1}$
Return 0 otherwise

$\mathsf{SignO}(\mathbf{m} \in (\mathbb{Z}_p^\ell)^*)$:
$\mathcal{Q}_{\mathsf{sign}} := \mathcal{Q}_{\mathsf{sign}} \cup \{\mathbf{m}\},\ \mathbf{U} \leftarrow_{\mathrm{R}} \mathsf{GL}_k,\ \mathbf{S} := \mathbf{AU}$
For all $i \in [\ell]$: $\mathbf{S}_i := [m_i]_1\mathbf{S}$
$[\mathbf{S}_{\ell+1}]_1 := \sum_{i=1}^{\ell} [m_i]_1 \mathbf{K}_i^\top \mathbf{S} + \colorbox{lightgray}{$\sum_{i=1}^{\ell} [m_i]_1 \mathbf{b}^\perp \mathbf{k}_i^\top \mathbf{S}$}$
Return $\rho := \big([\mathbf{S}]_2, \{[\mathbf{S}_i]_1\}_{i\in[\ell+1]}\big)$
```

Fig. 2. Game_0 through Game_2, for the proof of Theorem 1. In each procedure, the components inside a solid (gray) frame are only present in the games marked by a solid (gray) frame.

Game_2**:** We bound Adv_2 using a core lemma (Lemma 3), which essentially proves EUF-CoMA of a private-key variant of our EQS. Namely, we build a PTT adversary $\mathcal{A}'$ such that

$$\mathsf{Adv}_2 \leq \mathsf{Adv}_{\mathcal{A}'}^{\mathrm{core}}(\lambda) ,$$

where $\mathsf{Adv}_{\mathcal{A}'}^{\mathrm{core}}(\lambda) := \Pr[\mathsf{Exp}_{\mathrm{core}}(1^\lambda, \mathcal{A}') = 1]$ and $\mathsf{Exp}_{\mathrm{core}}(1^\lambda, \mathcal{A}')$ is defined in Fig. 3. Using the core lemma, we then get that there exists a PPT algorithm $\mathcal{B}_2$ such that:

$$\mathsf{Adv}_2 \leq 2Q_{\mathsf{Sign}} \cdot \mathsf{Adv}_{\mathsf{GGen},\mathcal{D}_{2k,k},\mathcal{B}_2}^{\mathrm{MDDH}}(\lambda) + \frac{3kQ_{\mathsf{Sign}} + 1}{p} .$$

```
$\underline{\mathsf{Exp}_{\mathsf{core}}(1^\lambda, \mathcal{A}')}$:
$\mathcal{Q}_{\mathsf{tag}} := \emptyset$, $pk := \mathcal{PG} \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow \mathcal{D}_{2k,k}$, $\mathbf{K} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{\ell \times 2k}$, $sk := (\mathbf{A}, \mathbf{K})$
$([\mathbf{m}^\star]_1, \tau^\star) \leftarrow {\mathcal{A}'}^{\mathsf{TagO}(\cdot)}(pk)$
Return 1 if $\mathsf{VerO}([\mathbf{m}^\star]_1, \tau^\star) = 1$ and $[\mathbf{m}^\star]_1 \notin \bigcup_{\mathbf{m} \in \mathcal{Q}_{\mathsf{tag}}} \mathsf{Span}([\mathbf{m}]_1)$, 0 otherwise

$\underline{\mathsf{VerO}\big([\mathbf{m}^\star]_1, \tau^\star\big)}$:
Parse $\tau^\star := ([t_0]_1, [\mathbf{t}]_2) \in \mathbb{G}_1 \times \mathbb{G}_2^{2k}$
Return 1 if $[\mathbf{t}]_2 \neq [\mathbf{0}]_2$ and $[{\mathbf{m}^\star}^\top]_1 \mathbf{K} \bullet [\mathbf{t}]_2 = [t_0]_1 \bullet [1]_2$
Return 0 otherwise

$\underline{\mathsf{TagO}(\mathbf{m} \in (\mathbb{Z}_p^\ell)^*)}$:
$\mathcal{Q}_{\mathsf{tag}} := \mathcal{Q}_{\mathsf{tag}} \cup \{\mathbf{m}\}$, $\mathbf{U} \leftarrow_{\mathrm{R}} \mathsf{GL}_k$, $\mathbf{T} := \mathbf{A}\mathbf{U}$, $\mathbf{t}_0^\top := \mathbf{m}^\top \mathbf{K}\mathbf{T}$
Return $\tau := ([\mathbf{t}_0]_1, [\mathbf{T}]_1, [\mathbf{T}]_2)$
```

Fig. 3. Experiment for Lemma 3.

We now describe the adversary $\mathcal{A}'$ playing in the security game $\mathsf{Exp}_{\mathsf{core}}$ in Fig. 3. It first gets the public key $\mathcal{PG} \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda)$, then samples $\mathbf{B} \leftarrow_{\mathrm{R}} \mathcal{D}_{k'}$, picks $\mathbf{b}^\perp \in \mathsf{orth}(\mathbf{B})$, $\mathbf{K}_i \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2k \times (k'+1)}$ for all $i \in [\ell]$, and runs $\mathcal{A}$ on input $pk := \big([\mathbf{B}]_2, \{[\mathbf{K}_i\mathbf{B}]_2\}_{i \in [\ell]}\big)$.

Then, to simulate oracle SignO in Game_2 for query $\mathbf{m} \in (\mathbb{Z}_p^\ell)^*$, $\mathcal{A}'$ queries its own oracle $\mathsf{TagO}(\mathbf{m})$ to obtain $([\mathbf{t}_0]_1, [\mathbf{T}]_1, [\mathbf{T}]_2)$. It sets $[\mathbf{S}]_2 := [\mathbf{T}]_2$, and computes for all $i \in [\ell]$: $[\mathbf{S}_i]_1 := m_i[\mathbf{T}]_1$, and $[\mathbf{S}_{\ell+1}]_1 := \sum_{i=1}^{\ell} m_i \mathbf{K}_i^\top [\mathbf{T}]_1 + \mathbf{b}^\perp [\mathbf{t}_0^\top]_1$. Note that with $\mathbf{K}^\top =: (\mathbf{k}_1 \| \cdots \| \mathbf{k}_\ell)$ we have $\mathbf{t}_0^\top = \sum_{i \in [\ell]} m_i \mathbf{k}_i^\top \mathbf{T}$, thus the values $\mathbf{k}_i$ in the simulation of Game_2 are implicitly defined by $\mathbf{K}$ from Fig. 3, chosen by $\mathcal{A}'$'s challenger. $\mathcal{A}'$ returns $\sigma := \big(\{[\mathbf{S}_i]_1\}_{i \in [\ell+1]}, [\mathbf{S}]_2\big)$ to $\mathcal{A}$.

Finally, when $\mathcal{A}$ sends its forgery $\big([\mathbf{m}^\star]_1, \sigma^\star := \big(\{[\mathbf{s}_i]_1\}_{i \in [\ell+1]}, [\mathbf{s}]_2\big)\big)$, $\mathcal{A}'$ uses it to create a forgery on its own as follows. First, $\mathcal{A}'$ checks that

$$[\mathbf{s}]_2 \neq [\mathbf{0}]_2 \tag{1}$$

$$\forall\, i \in [\ell]: \quad [\mathbf{s}_i]_1 \bullet [1]_2 = [m_i^\star]_1 \bullet [\mathbf{s}]_2 \tag{2}$$

$$\exists\, [t_0]_1 \in \mathbb{G}_1: \quad [\mathbf{s}_{\ell+1}^\top]_1 - \textstyle\sum_{i=1}^{\ell} [\mathbf{s}_i^\top]_1 \mathbf{K}_i = (\mathbf{b}^\perp)^\top \cdot [t_0]_1 \tag{3}$$

Note that $\mathcal{A}'$ can efficiently check (3) since it knows $\mathbf{b}^\perp \in \mathbb{Z}_p^{k'+1}$. Indeed, given any vector $[\mathbf{x}]_1 \in \mathbb{G}_1^n$ and $\mathbf{y} \in \mathbb{Z}_p^n$ for some $n \in \mathbb{N}^*$, one can efficiently compute $[\det(\mathbf{x} \| \mathbf{y})]_1$ since this only requires computing exponentiations in $\mathbb{G}_1$.

Note that if the forgery submitted by $\mathcal{A}$ is successful, it must satisfy (1), (2), and the following equation (cf. Fig. 2):

$$\textstyle\sum_{i=1}^{\ell} [\mathbf{s}_i^\top]_1 \big(\mathbf{K}_i + \mathbf{k}_i (\mathbf{b}^\perp)^\top\big) = [\mathbf{s}_{\ell+1}^\top]_1 \tag{4}$$

which implies (3), with $[t_0]_1 := \sum_{i=1}^{\ell} [\mathbf{s}_i^\top]_1 \cdot \mathbf{k}_i$.

Thus, if either (1), (2) or (3) fails, then $\mathcal{A}$ produced an unsuccessful forgery and $\mathcal{A}'$ can abort.

Otherwise, $\mathcal{A}$ can efficiently compute $[t_0]_1 \in \mathbb{G}_1$ satisfying (3), from $(\mathbf{b}^\perp)^\top \cdot [t_0]_1$ and $\mathbf{b}^\perp$: let $i \in [k'+1]$ be such that the i-th coordinate of $\mathbf{b}^\perp$ is non-zero (recall $\mathbf{b}^\perp \neq \mathbf{0}$); then $[t_0]_1$ is the i-th coordinate of $(\mathbf{b}^\perp)^\top \cdot [t_0]_1$ divided by the i-th coordinate of $\mathbf{b}^\perp$. Finally, $\mathcal{A}'$ sets $[\mathbf{t}]_2 := [\mathbf{s}]_2$, and returns the forgery $\big([\mathbf{m}^\star]_1, ([t_0]_1, [\mathbf{t}]_2)\big)$ in $\mathsf{Exp}_{\text{core}}$.

When $\mathcal{A}$ submits a successful forgery $\big([\mathbf{m}^\star]_1, \sigma^\star := \big(\{[\mathbf{s}_i]_1\}_{i\in[\ell+1]}, [\mathbf{s}]_2\big)\big)$, it satisfies (1), (2). Moreover, it satisfies (4), which means the value computed by $\mathcal{A}'$ is

$$[t_0]_1 := \textstyle\sum_{i=1}^{\ell} [\mathbf{s}_i^\top]_1 \cdot \mathbf{k}_i \ . \tag{5}$$

This implies that the forgery produced by $\mathcal{A}'$ is also successful, since it satisfies

$$[\mathbf{t}]_2 \neq [\mathbf{0}]_2 \quad \text{by (1), and}$$
$$\textstyle\sum_{i=1}^{\ell} [m_i^\star]_1 \mathbf{k}_i^\top \bullet [\mathbf{t}]_2 = [t_0]_1 \bullet [1]_2 \quad \text{by (2) and (5).}$$

This concludes the proof that $\mathsf{Adv}_2 \leq \mathsf{Adv}^{\text{core}}_{\mathcal{A}'}(\lambda)$. □

To prove the above theorem, we use the following core lemma, which essentially proves the security of a private-key variant of our EQS scheme.

Lemma 3 (Core lemma). *For an adversary $\mathcal{A}'$ and a security parameter $\lambda \in \mathbb{N}^*$, let $\mathsf{Adv}^{\text{core}}_{\mathcal{A}'}(\lambda) := \Pr[\mathsf{Exp}_{\text{core}}(1^\lambda, \mathcal{A}') = 1]$, with $\mathsf{Exp}_{\text{core}}(1^\lambda, \mathcal{A}')$ depicted in Fig. 3. Then for any PPT adversary $\mathcal{A}'$, there exists a PPT algorithm $\mathcal{B}$ such that:*

$$\mathsf{Adv}^{\text{core}}_{\mathcal{A}'}(\lambda) \leq 2kQ_{\mathsf{Tag}} \cdot \mathsf{Adv}^{\text{MDDH}}_{\mathsf{GGen},\mathcal{D}_{2k,k},\mathcal{B}}(\lambda) + \frac{3kQ_{\mathsf{Tag}}+1}{p} \ ,$$

where Q_{Tag} is the number of tag queries.

Proof of Lemma 3. We use hybrids $\mathsf{Game}_{i.1}$ for $i \in [Q_{\mathsf{Sign}}+1]$, and $\mathsf{Game}_{i.2}$, $\mathsf{Game}_{i.3}$ for $i \in [Q_{\mathsf{Sign}}]$, described in Fig. 4, and we denote by Adv_i the advantage of $\mathcal{A}'$ in Game_i, that is $\Pr[\mathsf{Game}_i(1^\lambda, \mathcal{A}') = 1]$, where the probability is taken over the random coins of Game_i and $\mathcal{A}'$.

$\mathsf{Game}_{1.1}$ is $\mathsf{Exp}_{\text{core}}(1^\lambda, \mathcal{A}')$.

From $\mathsf{Game}_{i.1}$ to $\mathsf{Game}_{i.2}$: We switch the matrices $[\mathbf{T}]_1$ and $[\mathbf{T}]_2$ computed by TagO on its i-th query to uniformly random over $\mathbb{Z}_p^{2k\times k}$, using the $\mathcal{D}_{2k,k}$-MDDH assumption. Namely, we show that for all $i \in [Q_{\mathsf{Tag}}]$, there is a PPT algorithm $\mathcal{B}_{i.1}$ such that

$$|\mathsf{Adv}_{i.1} - \mathsf{Adv}_{i.2}| \leq k \cdot \mathsf{Adv}^{\text{MDDH}}_{\mathsf{GGen},\mathcal{D}_{2k,k},\mathbb{G}_2,\mathcal{B}_{i.1}}(\lambda) + \frac{k}{p} \ .$$

First, we argue that the distribution $\mathbf{U} \leftarrow_{\text{R}} \mathsf{GL}_k$ and $\mathbf{U} \leftarrow_{\text{R}} \mathbb{Z}_p^{k\times k}$ are $\frac{k}{p}$-close. Then, we use the k-fold $\mathcal{D}_{2k,k}$-MDDH assumption (which reduces to its 1-fold variant with a security loss of k, via a hybrid argument) to switch

```
$\mathsf{Game}_{i.1}$, $\boxed{\mathsf{Game}_{i.2}, \colorbox{lightgray}{$\mathsf{Game}_{i.3}$}, \boxed{\mathsf{Game}_{i.4}}}$:
$\mathcal{Q}_{\mathsf{tag}} := \emptyset$, $pk := \mathcal{PG} \leftarrow_{\mathrm{R}} \mathsf{GGen}(1^\lambda)$, $\mathbf{A} \leftarrow \mathcal{D}_{2k,k}$, $\mathbf{K} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{\ell \times 2k}$, $sk := (\mathbf{A}, \mathbf{K})$
$\boxed{\text{Pick } \mathbf{A}^\perp \in \mathsf{orth}(\mathbf{A}) \text{ such that } (\mathbf{A}^\perp)^\top \mathbf{A}^\perp = \mathsf{Id}_{k \times k}}$
$([\mathbf{m}^\star]_1, \tau^\star) \leftarrow \mathcal{A}^{\mathsf{TagO}(\cdot)}(pk)$
Return 1 if $\mathsf{VerO}([\mathbf{m}^\star]_1, \tau^\star) = 1$ and $[\mathbf{m}^\star]_1 \notin \bigcup_{\mathbf{m} \in \mathcal{Q}_{\mathsf{tag}}} \mathsf{Span}([\mathbf{m}]_1)$, 0 otherwise

$\mathsf{VerO}\big([\mathbf{m}^\star]_1, \tau^\star := ([t_0]_1, [\mathbf{t}]_2\big)$:
Return 1 if $[\mathbf{t}]_2 \neq [\mathbf{0}]_2$ and $[\mathbf{m}^{\star\top}]_1 \mathbf{K} \bullet [\mathbf{t}]_2 = [t_0]_1 \bullet [1]_2$
Return 0 otherwise

$\mathsf{TagO}(\mathbf{m})$:
On the $\nu$'th query, $\mathcal{Q}_{\mathsf{tag}} := \mathcal{Q}_{\mathsf{tag}} \cup \{\mathbf{m}\}$, then:
If $\nu < i$: $\mathbf{U} \leftarrow_{\mathrm{R}} \mathsf{GL}_k$, $\mathbf{T} := \mathbf{A}\mathbf{U}$, $\mathbf{t}_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$

If $\nu = i$: $\mathbf{U} \leftarrow_{\mathrm{R}} \mathsf{GL}_k$, $\mathbf{T} := \mathbf{A}\mathbf{U}$, $\boxed{\mathbf{U}, \mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k \times k}, \mathbf{T} := \mathbf{A}\mathbf{U} + \mathbf{A}^\perp \mathbf{V}}$
$\colorbox{lightgray}{$\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$}$, $\mathbf{t}_0^\top := \mathbf{m}^\top \mathbf{K}\mathbf{T} + \colorbox{lightgray}{$\mathbf{w}^\top \mathbf{V}$}$, $\boxed{\mathbf{t}_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k}$

If $\nu > i$: $\mathbf{U} \leftarrow_{\mathrm{R}} \mathsf{GL}_k$, $\mathbf{T} := \mathbf{A}\mathbf{U}$, $\mathbf{t}_0^\top := \mathbf{m}^\top \mathbf{K}\mathbf{T}$

Return $\tau := ([\mathbf{t}_0]_1, [\mathbf{T}]_1, [\mathbf{T}]_2, )$
```

Fig. 4. $\mathsf{Game}_{i.1}$ for $i \in [\mathcal{Q}_{\mathsf{tag}} + 1]$ and $\mathsf{Game}_{i.2}$, $\mathsf{Game}_{i.3}$ for $i \in [\mathcal{Q}_{\mathsf{tag}}]$ in the proof of Lemma 3. In each procedure, the components inside a solid (dotted, gray) frame are only present in the games marked by a solid (dotted, gray) frame. In particular, the $\boxed{\text{solid}}$ frame is present in all games except $\mathsf{Game}_{i.1}$.

$\{[\mathbf{A}]_s, [\mathbf{A}\mathbf{U}]_s\}_{s \in \{1,2\}}$ to $\{[\mathbf{A}]_s, [\mathbf{T}]_s\}_{s \in \{1,2\}}$ where $\mathbf{T} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$. We give a precise description of the reduction $\mathcal{B}'_{i.1}$ to the k-fold $\mathcal{D}_{2k,k}$-MDDH assumption below. Finally, we use the basis $(\mathbf{A}|\mathbf{A}^\perp)$ of $\mathbb{Z}_p^{2k}$, where $\mathbf{A}^\perp \in \mathsf{orth}(\mathbf{A})$, and $(\mathbf{A}^\perp)^\top \mathbf{A}^\perp = \mathsf{Id}_{k \times k}$, the identity matrix in $\mathbb{Z}_p^{k \times k}$, which allows us to write $\mathbf{T} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2k \times k}$ as $\mathbf{T} := \mathbf{A}\mathbf{U} + \mathbf{A}^\perp \mathbf{V}$, with $\mathbf{U}, \mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k \times k}$.

We now describe adversary $\mathcal{B}'_{i.1}$ playing against the k-fold $\mathcal{D}_{2k,k}$-MDDH assumption. Given a challenge $\big(\mathcal{PG}, \{[\mathbf{A}]_s, [\mathbf{Z}]_s\}_{s \in \{1,2\}}\big)$, where $[\mathbf{Z}]_s \in \mathbb{G}_s^{2k \times k}$ is either of the form $[\mathbf{A}\mathbf{U}]_s$ for $\mathbf{U} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k \times k}$ or uniformly random over $\mathbb{G}_s^{2k \times k}$, $\mathcal{B}'_{i.1}$ samples $\mathbf{K} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{\ell \times 2k}$, which it uses to simulate VerO. To simulate $\mathsf{TagO}(\mathbf{m} \in \mathbb{Z}_p^\ell)$ on its ν-th query, it does the following:

- if $\nu < i$: $\mathcal{B}'_{i.1}$ samples $\mathbf{U} \leftarrow_{\mathrm{R}} \mathsf{GL}_k$, $\mathbf{t}_0 \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$, computes $[\mathbf{T}]_s := [\mathbf{A}]_s\mathbf{U}$ for all $s \in \{1,2\}$, and returns $([\mathbf{t}_0]_1, [\mathbf{T}]_1, [\mathbf{T}]_2)$ to $\mathcal{A}'$.
- if $\nu = i$: $\mathcal{B}'_{i.1}$ sets $[\mathbf{T}]_s := [\mathbf{Z}]_s$ for all $s \in \{1,2\}$, computes $[\mathbf{t}_0^\top]_1 := \mathbf{m}^\top\mathbf{K}[\mathbf{T}]_1$, and returns $([\mathbf{t}_0]_1, [\mathbf{T}]_1, [\mathbf{T}]_2)$ to $\mathcal{A}'$.
- if $\nu > i$: $\mathcal{B}'_{i.1}$ samples $\mathbf{U} \leftarrow_{\mathrm{R}} \mathsf{GL}_k$, computes $[\mathbf{T}]_s := [\mathbf{A}]_s\mathbf{U}$ for all $s \in \{1,2\}$, $[\mathbf{t}_0^\top]_1 := \mathbf{m}^\top\mathbf{K}[\mathbf{T}]_1$, and returns $([\mathbf{t}_0]_1, [\mathbf{T}]_1, [\mathbf{T}]_2)$ to $\mathcal{A}'$.

From $\mathsf{Game}_{i.2}$ to $\mathsf{Game}_{i.3}$: We show that

$$|\mathsf{Adv}_{i.2} - \mathsf{Adv}_{i.3}| = 0 \ .$$

To do so, first consider the *selective* variant of these games, that is, $\mathsf{Game}^\star_{i.2}$ and $\mathsf{Game}^\star_{i.3}$, which are as $\mathsf{Game}_{i.2}$ and $\mathsf{Game}_{i.3}$ except that the adversary has to commit to the forgery message $[\mathbf{m}^\star]_1$ beforehand. We will then show that $|\mathsf{Adv}^\star_{i.2} - \mathsf{Adv}^\star_{i.3}| = 0$. Using complexity leveraging,[3] we obtain $\mathsf{Adv}^\star_{i.2} = p^{-\ell} \cdot \mathsf{Adv}_{i.2}$ and $\mathsf{Adv}^\star_{i.2} = p^{-\ell} \cdot \mathsf{Adv}_{i.2}$, which allows to conclude.

We now prove that $|\mathsf{Adv}^\star_{i.2} - \mathsf{Adv}^\star_{i.3}| = 0$. We use the fact that the distributions:

$$\mathbf{K} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{\ell \times 2k} \quad \text{and} \quad \mathbf{K} + \mathbf{M}^\perp\mathbf{Z}(\mathbf{A}^\perp)^\top \text{ with } \mathbf{K} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{\ell \times 2k} \ ,$$

are identical, where $\mathbf{M}^\perp \in \mathsf{orth}(\mathbf{m}^{\star\top})$, that is, $\mathbf{M}^\perp \in \mathbb{Z}_p^{\ell \times (\ell-1)}$ is a full-rank matrix such that $\mathbf{m}^{\star\top}\mathbf{M}^\perp = \mathbf{0}$; $\mathbf{Z} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(\ell-1) \times 2k}$; and $\mathbf{A}^\perp \in \mathsf{orth}(\mathbf{A})$ such that $(\mathbf{A}^\perp)^\top\mathbf{A}^\perp = \mathsf{Id}_{k \times k}$.

Since $\mathbf{K}$ is distributed like $\mathbf{K} + \mathbf{M}^\perp\mathbf{Z}(\mathbf{A}^\perp)^\top$, we replace the former by the latter in $\mathsf{Game}^\star_{i.2}$ and then show that the resulting game is distributed like $\mathsf{Game}^\star_{i.3}$.

We start with the oracle $\mathsf{VerO}\big([\mathbf{m}^\star]_1, \tau^\star := ([t_0]_1, [\mathbf{t}]_2,)\big)$, which checks:

$$[\mathbf{m}^{\star\top}]_1(\mathbf{K} + \mathbf{M}^\perp\mathbf{Z}(\mathbf{A}^\perp)^\top) \bullet [\mathbf{t}]_2 = [\mathbf{m}^{\star\top}]_1\mathbf{K} \bullet [\mathbf{t}]_2 \overset{?}{=} [t_0]_1 \bullet [1]_2 \ ,$$

where the first equality always holds, since $\mathbf{m}^{\star\top}\mathbf{M}^\perp = \mathbf{0}$.

Let us now analyze the TagO queries. For the first $i-1$ queries, the output of TagO is independent of $\mathbf{K}$.

Consider the i-th query $[\mathbf{m}]_1 \in (\mathbb{G}_1^\ell)^*$ to the TagO oracle. We have:

$$\mathbf{t}_0^\top := \mathbf{m}^\top\mathbf{K}\mathbf{T} + \mathbf{m}^\top\mathbf{M}^\perp\mathbf{Z}(\mathbf{A}^\perp)^\top(\mathbf{A}\mathbf{U} + \mathbf{A}^\perp\mathbf{V}) = \mathbf{m}^\top\mathbf{K}\mathbf{T} + \mathbf{m}^\top\mathbf{M}^\perp\mathbf{Z}\mathbf{V} \ ,$$

where for the last equality we used $(\mathbf{A}^\perp)^\top\mathbf{A} = \mathbf{0}$ and $(\mathbf{A}^\perp)^\top\mathbf{A}^\perp = \mathsf{Id}_{k \times k}$. Moreover, if the adversary wins the game then $\mathbf{m}^\top\mathbf{M}^\perp \neq \mathbf{0}$, as otherwise $\mathbf{m}$ is a multiple of $\mathbf{m}^\star$, thus the latter is not a valid forgery. Now $\mathbf{m}^\top\mathbf{M}^\perp \neq \mathbf{0}$ implies that $\mathbf{m}^\top\mathbf{M}^\perp\mathbf{Z}$ is identically distributed to $\mathbf{w}^\top \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{1 \times 2k}$, as in $\mathsf{Game}_{i.3}$.

[3] Complexity leveraging is a technique that allows to prove adaptive from selective security: the reduction (playing in the selective game) simply guesses the (adaptive) adversary's forgery at the beginning of the game and aborts if its guess later turns out wrong. The security loss of the reduction is therefore inversely proportional to the number of guesses (here: number of messages, i.e. p^ℓ).

For the remaining TagO queries, $\mathsf{TagO}(\mathbf{m})$ computes $\mathbf{t}_0^\top := \mathbf{m}^\top(\mathbf{K} + \mathbf{M}^\perp\mathbf{Z}(\mathbf{A}^\perp)^\top)\mathbf{AU} = \mathbf{m}^\top\mathbf{KAU}$, since $(\mathbf{A}^\perp)^\top\mathbf{A} = \mathbf{0}$.

All in all, we have thus shown that the modified game (which is distributed like $\mathsf{Game}^\star_{i.2}$) is distributed equivalently to $\mathsf{Game}^\star_{i.3}$.

From $\mathsf{Game}_{i.3}$ to $\mathsf{Game}_{i.4}$: We show that these two games are statistically close. This follows from the fact that with probability at least $1 - \frac{k}{p}$ over the choice of $\mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k \times k}$, $\mathbf{V}$ is invertible. In that case, $\mathbf{w}^\top\mathbf{V}$ for $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ is uniformly random over $\mathbb{Z}_p^{1\times k}$, which means the vector $\mathbf{t}_0$ computed by TagO on its i-th query is itself uniformly random over $\mathbb{Z}_p^k$, as in $\mathsf{Game}_{i.4}$.

We note that for this step is was crucial that V is a $k \times k$ matrix. For the definition of $\mathbf{T}$ from $\mathsf{Game}_{i.2}$ on, we therefore require that $\mathbf{A}^\perp \in \mathbb{Z}_p^{2k\times k}$, which is what forced us to choose $\mathbf{A} \in \mathbb{Z}_p^{2k\times k}$ (rather than $\mathbf{A} \in \mathbb{Z}_p^{(k+1)\times k}$).

From $\mathsf{Game}_{i.4}$ to $\mathsf{Game}_{i+1.1}$: We switch back the matrices $[\mathbf{T}]_1$ and $[\mathbf{T}]_2$ computed by TagO on its i-th query to $[\mathbf{AU}]_1$ and $[\mathbf{AU}]_2$ with $\mathbf{U} \leftarrow_{\mathrm{R}} \mathsf{GL}_k$, using the k-fold $\mathcal{D}_{2k,k}$-MDDH assumption. This transition is similar to the transition from $\mathsf{Game}_{i.1}$ to $\mathsf{Game}_{i.2}$. We defer to the latter for further details.

$\mathsf{Game}_{\mathcal{Q}_{\mathsf{tag}}+1.1}$: We show that $\mathsf{Adv}_{\mathcal{Q}_{\mathsf{tag}}+1.1} = \frac{1}{p}$. In this game there is no information leaked about $\mathbf{K}$ prior to $\mathcal{A}'$'s query to VerO, since all the tags generated by TagO contain a uniformly random vector $[\mathbf{t}_0]_1 \leftarrow_{\mathrm{R}} \mathbb{G}_1^k$. Therefore, the vector $[\mathbf{m}^{\star\top}]_1\mathbf{K}$ computed by $\mathsf{VerO}\big([\mathbf{m}^\star]_1, \tau^\star := ([t_0]_1, [\mathbf{t}]_2)\big)$ is uniformly random over $\mathbb{G}_1^{1\times 2k}$, which means $[\mathbf{m}^{\star\top}]_1\mathbf{K} \bullet [\mathbf{t}]_2$ is uniformly random over $\mathbb{G}_T$, independent of $[t_0]_1$, when $[\mathbf{t}]_2 \neq [\mathbf{0}]_2$. Thus, we have: $\mathsf{Adv}_{\mathcal{Q}_{\mathsf{tag}}+1.1} = \frac{1}{p}$. □

4 Application to Access Control Encryption

Access Control Encryption. Damgård, Haagh and Orlandi [DHO16] introduced the notion of access control encryption (ACE), which allows to control the information flow between senders and receivers. In their model each sender $i \in \{0,1\}^n$ has an encryption key ek_i, and each receiver $j \in \{0,1\}^n$ has a decryption key dk_j; the system specifies an access control policy $P\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, and communication is allowed from sender i to receiver j iff $P(i,j) = 1$. Thus, ACE restricts both what information is being received (this is captured by a so-called *No-Read rule*), and what can be sent (captured by a so-called *No-Write rule*). To prevent sending of information by unauthorized senders (No-Write rule), it is necessary to assume that messages are relayed via a special party, called the *sanitizer*, which is assumed to be honest (it will behave according to the protocol specification) but curious (it will try to learn additional information by colluding with other parties in the system).

More precisely, the No-Read rule stipulates that given all encryptions keys, if the sanitizer colludes with a set of unauthorized receivers $\mathcal{J} \subset \{0,1\}^n$, it should not be able to learn any information from an encryption by sender $i \in \{0,1\}^n$ if $P(i,j) = 0$ for all $j \in \mathcal{J}$. In particular, both the underlying plaintext and the identity of i should remain hidden. The No-Write rule roughly says that a collusion of senders $\mathcal{I} \subset \{0,1\}^n$ and receivers $\mathcal{J} \subset \{0,1\}^n$ such that $P(i,j) = 0$ for all $i \in \mathcal{I}$, $j \in \mathcal{J}$ that tries to exchange information will be prevented from doing so by the (in this case honest) sanitizer. (If the sanitizer is corrupt then it can always distribute information and thereby break the No-Write rule.) We recall the formal definitions below for completeness.

Construction from EQS. Fuchsbauer, Gay, Kowalczyk and Claudio Orlandi [FGKO17] built the first pairing-based ACE for predicates such as equality ($P(i,j) = 1 \Leftrightarrow i = j$) and range ($P(i,j) = 1 \Leftrightarrow i \leq j$), whose ciphertexts contain $O(n)$ group elements. The work introducing the concept [DHO16] had built ACE from indistinguishability obfuscation for general circuits and gave an inefficient construction from DDH with ciphertexts of size $O(2^n)$.

One construction from [FGKO17] generically uses EQS, which they instantiated with the scheme from [FHS14] and thus relies on an interactive assumption. When replacing their EQS with our scheme from Sect. 3, we obtain another efficient ACE. We need to show that the relaxed unforgeability notion satisfied by our EQS (namely EUF-CoMA; Def. 5) suffices for the security of the ACE. We note that the resulting ACE (as for [FGKO17]), does *not* require a private key for the sanitizer, unlike the original schemes from [DHO16].

Related works. Recent work [KW17] builds ACE for *arbitrary* access control policies based on standard assumptions (such as DDH or LWE), using (single-key) general-purpose functional encryption and predicate encryption. Our scheme has the advantage of being much more efficient, although specialized to the equality and range predicates. In [BMM17], the authors define new, stronger security notions for ACE and give constructions that achieve them under standard assumptions for the equality predicate, which can be lifted to a disjunction of equalities and to predicates such a range, as shown in [FGKO17].

In the rest of this section, we first recall the definition of ACE from [FGKO17] and the construction for the equality predicate [FGKO17, Construction 2]. We then give a proof of its security when the underlying EQS is only EUF-CoMA. ACE for range can then be obtained from the ACE for equality generically, as shown in [FGKO17].

Definition 6 (ACE). *An* access control encryption *(ACE) [FGKO17] scheme is defined by the following PPT algorithms:*

- $\mathsf{Setup}(1^\lambda, P)$*, on input the security parameter* $\lambda \in \mathbb{N}$ *and a policy* $P\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$*, outputs a master secret key msk and public parameters pp (which implicitly define the message space* $\mathcal{M}$ *and ciphertext spaces* $\mathcal{C}, \mathcal{C}'$*).*
- $\mathsf{Gen}(msk, i, t)$ *is a* deterministic *algorithm that on input the master secret key msk, an identity* $i \in \{0,1\}^n$ *and a type* $t \in \{\mathtt{sen}, \mathtt{rec}\}$*, specifying whether i is a sender or a receiver, outputs a key k. We use the following notation for the types of keys:*

- $ek_i \leftarrow \mathsf{Gen}(msk, i, \mathtt{sen})$ *and call it an* encryption key *for* $i \in \{0,1\}^n$*,*
- $dk_j \leftarrow \mathsf{Gen}(msk, j, \mathtt{rec})$ *and call it a* decryption key *for* $j \in \{0,1\}^n$*.*

– $\mathsf{Enc}(ek_i, m)$*, on input an encryption key* ek_i *and a message* $m \in \mathcal{M}$*, outputs a ciphertext* $c \in \mathcal{C}$*.*

– $\mathsf{San}(pp, c)$*, on input the public parameters pp and a ciphertext* $c \in \mathcal{C}$*, outputs a sanitized ciphertext* $c' \in \mathcal{C}'$*.*

– $\mathsf{Dec}(dk_j, c')$ *is a deterministic algorithm that on input a decryption key* dk_j*, a ciphertext* $c' \in \mathcal{C}'$*, outputs a message* $m \in \mathcal{M} \cup \{\bot\}$*.*

Definition 7 (Correctness). *For all* $m \in \mathcal{M}$*,* $i, j \in \{0,1\}^n$ *with* $P(i,j) = 1$*:*

$$\Pr\left[\mathsf{Dec}\big(dk_j, \mathsf{San}(pp, \mathsf{Enc}\left(ek_i, m\right))\big) = m\right] \geq 1 - \mathsf{negl}(\lambda) \ ,$$

where the probability is taken over $(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda, P)$*,* $ek_i \leftarrow \mathsf{Gen}(msk, i, \mathtt{sen})$*, and* $dk_j \leftarrow \mathsf{Gen}(msk, j, \mathtt{rec})$*.*

Complementary to correctness, we require that it is detectable when decryption does not succeed, formalized as follows.

Definition 8 (Detectability). *For all* $m \in \mathcal{M}$*,* $i, j \in \{0,1\}^n$ *with* $P(i,j) = 0$*:*

$$\Pr\left[\mathsf{Dec}\big(dk_j, \mathsf{San}(pp, \mathsf{Enc}\left(ek_i, m\right))\big) = \bot\right] \geq 1 - \mathsf{negl}\left(\lambda\right) \ ,$$

where the probability is taken over $(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda, P)$*,* $ek_i \leftarrow \mathsf{Gen}(msk, i, \mathtt{sen})$*, and* $dk_j \leftarrow \mathsf{Gen}(msk, j, \mathtt{rec})$*.*

No-Read Rule. An access control encryption scheme $\mathsf{ACE} := (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ is said to satisfy the No-Read rule if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\text{No-Read}}_{\mathsf{ACE},\mathcal{A}}(\lambda) := \Pr\left[\mathsf{Exp}^{\text{No-Read}}_{\mathsf{ACE}}(1^\lambda, \mathcal{A}) = 1\right] - \tfrac{1}{2} = \mathsf{negl}(\lambda) \ ,$$

where the game $\mathsf{Exp}^{\text{No-Read}}_{\mathsf{ACE}}(1^\lambda, \mathcal{A})$ is defined as follows:

Game Definition	**Oracle Definition**
$\underline{\mathsf{Exp}^{\text{No-Read}}_{\mathsf{ACE}}(1^\lambda, \mathcal{A})\text{:}}$	$\underline{\mathcal{O}_G(j, t)\text{:}}$
$\mathcal{Q}_{\mathsf{key}} := \emptyset$	$\mathcal{Q}_{\mathsf{key}} := \mathcal{Q}_{\mathsf{key}} \cup \{(j, t)\}$
$(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda, P)$	Return $k \leftarrow \mathsf{Gen}(msk, j, t)$
$(m_0, m_1, i_0, i_1) \leftarrow \mathcal{A}^{\mathcal{O}_G(\cdot), \mathcal{O}_E(\cdot)}(pp)$	
$b \leftarrow \{0,1\}$; $c \leftarrow \mathsf{Enc}\big(\mathsf{Gen}(msk, i_b, \mathtt{sen}), m_b\big)$	$\underline{\mathcal{O}_E(i, m)\text{:}}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_G(\cdot), \mathcal{O}_E(\cdot)}(c)$	$ek_i \leftarrow \mathsf{Gen}(msk, i, \mathtt{sen})$
Return 1 iff $b' = b$, $\lvert m_0 \rvert = \lvert m_1 \rvert$, $i_0, i_1 \in \{0,1\}^n$ and	Return $c \leftarrow \mathsf{Enc}(ek_i, m)$
$\forall (j, \mathtt{rec}) \in \mathcal{Q}_{\mathsf{key}}$, $P(i_0, j) = P(i_1, j) = 0$	

Recall that Gen is assumed to be a deterministic algorithm, which is why the experiment need not do any bookkeeping of already-generated keys.

No-Write Rule. An access control encryption scheme $\mathsf{ACE} := (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ is said to satisfy the No-Write rule if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\text{No-Write}}_{\mathsf{ACE},\mathcal{A}}(\lambda) := \Pr\left[\mathsf{Exp}^{\text{No-Write}}_{\mathsf{ACE}}(1^\lambda, \mathcal{A}) = 1\right] - \tfrac{1}{2} = \mathsf{negl}(\lambda) \ ,$$

where the game $\mathsf{Exp}^{\text{No-Write}}_{\mathsf{ACE}}(1^\lambda, \mathcal{A})$ is defined as follows:

Game Definition	**Oracle Definition**
$\underline{\mathsf{Exp}^{\text{No-Write}}_{\mathsf{ACE}}(1^\lambda, \mathcal{A})\text{:}}$	$\underline{\mathcal{O}_S(j,t)\text{:}}$
$\mathcal{Q}_{\mathsf{sen}}, \mathcal{Q}_{\mathsf{rec}} := \emptyset$	$\mathcal{Q}_t := \mathcal{Q}_t \cup \{j\}$
$(pp, msk) \leftarrow \mathsf{Setup}(1^\lambda, P)$	Return $k \leftarrow \mathsf{Gen}(msk, j, t)$
$b \leftarrow \{0,1\};\ \ m' \leftarrow \mathcal{M}$	
$(c^{(0)}, i') \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot),\mathcal{O}_E(\cdot)}(pp)$	$\underline{\mathcal{O}_R(j,t)\text{:}}$
$c^{(1)} \leftarrow \mathsf{Enc}\big(\mathsf{Gen}(msk, i', \mathtt{sen}), m'\big)$	if $t = \mathtt{rec}$, $\mathcal{Q}_{\mathsf{rec}} := \mathcal{Q}_{\mathsf{rec}} \cup \{j\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_R(\cdot),\mathcal{O}_E(\cdot)}(\mathsf{San}(pp, c^{(b)}))$	Return $dk \leftarrow \mathsf{Gen}(msk, j, \mathtt{rec})$
Return 1 iff $b' = b$, $i' \in \mathcal{Q}_{\mathsf{sen}}$, $\mathsf{San}(pp, c^{(0)}) \neq \bot$	
and $\forall i \in \mathcal{Q}_{\mathsf{sen}}, j \in \mathcal{Q}_{\mathsf{rec}}$, $P(i,j) = 0$	$\underline{\mathcal{O}_E(i,m)\text{:}}$
	$ek_i \leftarrow \mathsf{Gen}(msk, i, \mathtt{sen})$
	Return $c \leftarrow \mathsf{San}\big(pp, \mathsf{Enc}(ek_i, m)\big)$

Remark 3 (Definition of the No-Write experiment). Oracle $\mathcal{O}_S$ needs to keep track of encryption query, since an encryption ek_i for i such that $P(i,j) = 1$ for some $j \in \mathcal{Q}_{\mathsf{rec}}$ would allow $\mathcal{A}$ to produce a ciphertext $c^{(0)}$ that once sanitized, could be decrypted using dk_j, unlike $c^{(1)}$, thus trivially breaking the game. However, encryption keys queried after the adversary committed to $c^{(0)}$ are useless in breaking No-Write, as they do not allow for extracting meaningful information from $c^{(0)}$ by the No-Real rule.

Note that $\mathcal{O}_E$ needs to return a sanitized ciphertext, since an unsanitized ciphertext would allow the following attack: an adversary queries $ek_{i'}$, dk_j for arbitrary $i', j \in \{0,1\}^n$ such that $P(i',j) = 0$, then gets $\mathsf{Enc}(ek_i, m)$ from $\mathcal{O}_E(i,m)$ for arbitrary message m and $i \in \{0,1\}^n$ such that $P(i,j) = 1$. It then sets $c^{(0)} := \mathsf{Enc}(ek_i, m)$ and sends $(c^{(0)}, i)$ to the No-Write experiment. By correctness, sanitized $c^{(0)}$ could be decrypted using dk_j. By detectability, decryption of sanitized $c^{(1)}$ with dk_j will output $\bot$ with overwhelming probability.

ACE for equality. An overview of the ACE by Fuchsbauer et al. [FGKO17, Construction 2] was given in the introduction (page 4); we recall it in Fig. 5. For ease of readability, we used randomized notion in the definition of Gen but emphasize that all randomness is derived deterministically from the PRF key K. Plugging in our new EQS from Sect. 3 yields an ACE for equality, disjunction of equality, and for range, as we show that EUF-CoMA of our EQS is sufficient to prove security of the ACE.

```
$\underline{\mathsf{Setup}(1^\lambda, P):}$
$\mathcal{PG} \leftarrow \mathsf{GGen}(1^\lambda)$;  $(sk, vk) \leftarrow \mathsf{EQS.Setup}(\mathcal{PG})$;  pick a PRF key $K$
Return $pp := (\mathcal{PG}, vk)$ and $msk := (sk, K)$

$\underline{\mathsf{Gen}(msk = (sk, K), i, t):}$
Use $K$ to pseudorandomly generate all needed randomness
$dk_i \leftarrow_{\mathrm{R}} \mathbb{Z}_p$
If $t = \mathtt{rec}$ then return $dk_i$
$pk_i := [dk_i]_1$;  $\sigma_i \leftarrow \mathsf{EQS.Sign}(sk, [1, dk_i]_1)$
Return $ek_i := (pk_i, \sigma_i)$

$\underline{\mathsf{Enc}(ek_i = (pk_i, \sigma_i), [m]_1 \in \mathbb{G}_1):}$
$r \leftarrow_{\mathrm{R}} \mathbb{Z}_p$;  $[c_0]_1 := [r]_1$;  $[c_1]_1 := r \cdot pk_i + [m]_1$
$s \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$;  $[c_2]_1 := [s]_1$;  $[c_3]_1 := s \cdot pk_i$;  $\sigma' \leftarrow \mathsf{EQS.Adapt}(vk, \sigma_i, s)$
Return $([c_0]_1, [c_1]_1, [c_2]_1, [c_3]_1, \sigma')$

$\underline{\mathsf{San}\Big(pp, ([c_0]_1, [c_1]_1, [c_2]_1, [c_3]_1, \sigma')\Big):}$
If $\mathsf{EQS.Ver}(vk, [c_2, c_3]_1, \sigma') = 0$ then return $\perp$.
Else, $t \leftarrow_{\mathrm{R}} \mathbb{Z}_p$;  $[c_0']_1 := [c_0]_1 + t \cdot [c_2]_1$;  $[c_1']_1 := [c_1]_1 + t \cdot [c_3]_1$
Return $([c_0']_1, [c_1']_1)$

$\underline{\mathsf{Dec}\big(dk_i, ([c_0']_1, [c_1']_1)\big):}$
Return $[c_1']_1 - dk_i \cdot [c_0']_1$
```

Fig. 5. ACE for equality, using an EQS ($\mathsf{EQS.Setup}, \mathsf{EQS.Sign}, \mathsf{EQS.Adapt}, \mathsf{EQS.Ver}$, $\mathsf{EQS.VerKey}$) and a PRF that takes a key K and outputs an element in $\mathbb{Z}_p$.

Correctness and **detectability** follow by inspection.

No-Read rule. The proof does not rely on the EUF-CMA security of the used EQS scheme and can be found in [FGKO17, Theorem 3]. We provide a sketch of the proof here. The proof goes through a sequence of hybrids, where in the first hybrid, we change the way the challenge ciphertext is computed: instead of containing a signature of the form $\sigma' \leftarrow \mathsf{EQS.Adapt}\big(vk, \mathsf{Sign}(sk, [1, dk_{i_b}]_1, s)\big)$, it is computed as $\sigma' \leftarrow \mathsf{EQS.Adapt}\big(vk, \mathsf{Sign}(sk, [s, s \cdot dk_{i_b}]_1, 1)\big)$. By the perfect adaptation of the signatures of the EQS, this does not change the distribution of the adversary's view.

Then, we use the DDH assumption in $\mathbb{G}_1$ to switch the vectors $[s, s \cdot dk_{i_b}]_1$ and $[r, r \cdot dk_{i_b} + m]_1$ one after the other to uniformly random elements from $\mathbb{G}_1^2$. The underlying plaintext and identity of the sender are then perfectly hidden. We can do so since by definition of the security game, the adversary is not allowed to query the decryption key dk_{i_b} (which the simulator does not know when relying on DDH during the game hops).

No-Write rule. Since our EQS achieves a weaker unforgeability notion, we need to show that it is still sufficient for the ACE to satisfy the No-Write rule. The proof follows closely the one from [FGKO17], which first replaces the pseudorandomness used in Gen by real randomness. Consider the following event E: the adversary $\mathcal{A}$ returns $c^{(0)} = ([c_0]_1, [c_1]_1, [c_2]_1, [c_3]_1, \sigma')$, which contains a successful EQS forgery. That is, $([c_2, c_3]_1, \sigma')$ passes the verification and $[c_2, c_3]_1$ is not a multiple of any $[1, dk_j]$ for $j \in \mathcal{Q}_{\mathsf{sen}}$ (where $\mathcal{Q}_{\mathsf{sen}}$ is the set of identities queried to $\mathcal{O}_S(\cdot, \mathtt{sen})$).

We (1) bound the probability of event E happening using the EUF-CoMA security of the EQS, and we (2) show that $\Pr[\mathsf{Exp}^{\text{No-Write}}_{\mathsf{ACE}}(1^\lambda, \mathcal{A}) = 1 \mid \neg E] - \frac{1}{2}$ is negligible, using the DDH assumption in $\mathbb{G}_1$ and the KEA [BP04].

(1) The reduction $\mathcal{B}$ playing the EUF-CoMA game of the EQS simulates the No-Write experiment for $\mathcal{A}$ as follows. Whenever $\mathcal{A}$ makes a query containing an identity i for the first time, $\mathcal{B}$ samples $dk_i \leftarrow_{\text{R}} \mathbb{Z}_p$. If it is a $\mathtt{sen}$ query, $\mathcal{B}$ queries its signing oracle SignO on $(1, dk_i) \in \mathbb{Z}_p^2$ to obtain σ_i and returns $ek_i := ([dk_i]_1, \sigma_i)$. Note that since the reduction picks the secret keys dk_i itself, it knows the discrete logarithms of the message being signed by the EQS: thus EUF-CoMA is sufficient. When $\mathcal{A}$ returns $\big(c^{(0)} := ([c_0]_1, [c_1]_1, [c_2]_1, [c_3]_1, \sigma'), i'\big)$, $\mathcal{B}$ then returns $([c_2, c_3]_1, \sigma')$ as its forgery. This is a successful forgery exactly when E happens.

(2) $\Pr[\mathsf{Exp}^{\text{No-Write}}_{\mathsf{ACE}}(1^\lambda, \mathcal{A}) \to 1 \mid \neg E]$ is bounded exactly as in the original proof [FGKO17, Theorem 4]. It requires KEA relative to GGen, which states that for every PPT algorithm $\mathcal{A}$, which given $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2) \leftarrow \mathsf{GGen}(1^\lambda)$ and a random $[r]_1 \leftarrow_{\text{R}} \mathbb{G}_1$, outputs $[s]_1, [r \cdot s]_1$ for some $s \in \mathbb{Z}_p$, there exists a PPT extractor which, when given the coins of $\mathcal{A}$, extracts s with non-negligible probability.

5 Application to Attribute-Based Credentials

Their main application of EQS in the work introducing the concept [HS14, FHS14] is an anonymous (multi-show) attribute-based credential (ABC) scheme, for which the authors introduce set commitment schemes with randomizable commitments.

ABCs. Credential schemes that we consider here let users obtain *credentials* for certain *attributes* that they possess from an *organization*. The users can then later *show* that they possess a credential for any subset of their attributes. Unforgeability requires that no user can show possession of attributes for which he was not issued a credential (moreover, users cannot combine their attributes). Anonymity requires that different showings of the same credential are unlinkable (credentials are thus *multi-show*) and that moreover, nothing is leaked about the contained attributes that are not shown. This property should hold even against a malicious organization. (See [FHS14] for the formal definitions.)

FHS's construction. Besides EQS, the second ingredient to constructing ABCs is a *set commitment scheme* that the authors introduce. These let one commit to sets and, besides regular commitment opening, one can open a commitment to any subset of the committed elements, without revealing anything about the committed elements that were not opened. Their construction is similar to polynomial commitments [KZG10] and it is perfectly hiding. The size of a commitment key is linear in the maximum size of the committed sets, whereas a commitment consists of a single group element from $\mathbb{G}_1^*$ and openings are in $\mathbb{Z}_p^*$. Openings to subsets (which hide the remaining elements) are in $\mathbb{G}_1^*$. Moreover, if $[c]_1$ is a commitment with opening ρ, then $s \cdot [c]_1$ is a commitment to the same set with opening $s \cdot \rho$.

Let us sketch the ABC scheme from [FHS14]:

1. A credential for a user consists of a commitment $[c]_1$ to the user's attributes, and an EQS signature σ by the organization on $([c]_1, [r \cdot c]_1, [1]_1)$; it also contains the opening ρ of $[c]_1$ and the value r.
2. When being issued a credential, the user chooses $\rho, r \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ and sends $[c]_1$ and $r \cdot [c]_1$ to obtain σ. In addition, the user gives an interactive zero-knowledge proof of knowledge (zkPoK) [CDM00] of ρ and the organization proves knowledge of its signing key.
3. When showing a credential, the user picks $s \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ and shows an adaptation of σ to $(s \cdot [c]_1, s \cdot [r \cdot c]_1, [s]_1)$. The user also presents an opening of the randomized commitment $s \cdot [c]_1$ to the subset of showed attributes.

Unforgeability of the ABC is showed [FHS14] by reducing a forgery to either a forgery of an EQS signature or a "forgery" of a subset opening of the commitment. After a slight modification of the issuing protocol, it suffices that the used EQS scheme satisfies our EUF-CoMA notion of unforgeability:

2'. Credentials are obtained as in [FHS14] (see 2. above), except that the user gives a zkPoK of ρ $\underline{\text{and } r}$.

In the proof of anonymity, these zkPoK are simulated anyway, so additionally proving knowledge of r does not break anything. In the proof of unforgeability, the simulation can now extract the value r in addition to ρ, which together with the randomness used to set up the commitment key completely define the logarithm of a tuple $([c]_1, [rc]_1, [1]_1)$. In the reduction of ABC unforgeability to EQS unforgeability, the simulator can thus make its signature queries using the logarithms $(c, r \cdot c, 1)$ instead of the group elements $([c]_1, [r \cdot c]_1, [1]_1)$. An EQS that is secure under our definition is thus sufficient for the application to anonymous ABCs.

ABCs with revocation. Derler, Hanser and Slamanig [DHS15] extend the protocol from [HS14] (which considers a trusted setup of parameters and achieves thus weaker security than the scheme from [FHS14]) to incorporate revocation of users.

It is easily seen that our slight modifications carry over to their protocol: extend the interactive proof of knowledge done by the user when obtaining a credential, so that the simulator in the unforgeability game can extract the logarithm of the message sent by the user. Again, EUF-CoMA of the EQS scheme then suffices to prove security.

6 Further Applications

For completeness, let us mention two more applications that only require our relaxed definition of unforgeability.

6.1 Group Signatures without Encryption

Inspired by the construction of ABC from EQS, Derler and Slamanig [DS16] use EQS to construct a dynamic (users can join at any point) group-signature scheme, which they show satisfies the formal model by Bellare Shi and Zhang [BSZ05]. In particular, the scheme is fully (i.e. CCA2-) anonymous (that is, in the anonymity game the adversary has access to an opening oracle). The scheme (roughly) works as follows:

- When joining the group, a user first chooses $q, r \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$. The value r will be linked to the user's identity and she creates an encryption of $[r]_2$ under the opener's public key, which she sends to the issuer together with $([q \cdot r]_1, [q]_1)$. She also proves that the ciphertext encrypts the correct value. The issuer replies with an EQS signature on the sent pair, from which the user derives a signature on $([r]_1, [1]_1)$, which serves as the signing key.
- When making a group signature, the user randomizes her key to $([\rho \cdot r]_1, [\rho]_1)$ and makes a "signature of knowledge" proving knowledge of the randomizer ρ.

As for the construction of a credential scheme from EQS in the previous section, a minor modification of the scheme suffices so that we can use EQS schemes that are EUF-CoMA secure: during issuing we require the user to make a zero-knowledge proof of knowledge of r and q. In the proof of *traceability* (which is the security notion that relies on unforgeability the EQS scheme), the reduction can extract these values and thus make an open-message query (qr, q) to its signing oracle.

6.2 Verifiably Encrypted Signatures

Hanser, Rabkin and Schröder [HRS15] use EQS to construct verifiably encrypted signatures. In their scheme, messages are elements from $\mathbb{Z}_p$ (rather than group elements) and they are signed by picking $s \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ and producing an EQS signature on $(s \cdot [m]_1, [s]_1, [1]_1)$. The arbiter's public key (who can decrypt verifiably encrypted signatures in case of dispute) is $[a]_1$ and a verifiably encrypted signature is defined as an EQS signature on $([m \cdot s \cdot a]_1, [s \cdot a]_1, [a]_1)$.

In the games defining the different security notions the adversary can either query signatures on messages $m \in \mathbb{Z}_p^*$ or verifiably encrypted signatures under the arbiter's public key. Since the latter is trusted in all notions, the security reduction always knows the discrete logarithms of the message for which it needs to produce an EQS signature; an EUF-CoMA-secure EQS scheme is thus sufficient.

7 Conclusion

We have presented the first EQS scheme from standard assumptions and showed that the relaxed unforgeability notion that it achieves is sufficient for *all* applications that have been considered in the literature, *except* the one to round-optimal blind signatures.

Acknowledgements. We would like to thank the anonymous reviewers for their valuable comments that greatly helped to improve the paper. The first author is supported by the French ANR EfTrEC project (ANR-16-CE39-0002). The second author is partially supported by ERC Project aSCEND (639554), and a Google PhD fellowship.

References

AC17. Shashank Agrawal and Melissa Chase. FAME: Fast attribute-based message encryption. In *ACM CCS 17*, pages 665–682. ACM Press, 2017.

ACD$^+$12. Masayuki Abe, Melissa Chase, Bernardo David, Markulf Kohlweiss, Ryo Nishimaki, and Miyako Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 4–24. Springer, Heidelberg, December 2012.

AFG$^+$10. Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 209–236. Springer, Heidelberg, August 2010.

AGHO11. Masayuki Abe, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 649–666. Springer, Heidelberg, August 2011.

AGO11. Masayuki Abe, Jens Groth, and Miyako Ohkubo. Separating short structure-preserving signatures from non-interactive assumptions. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 628–646. Springer, Heidelberg, December 2011.

AGOT14. Masayuki Abe, Jens Groth, Miyako Ohkubo, and Mehdi Tibouchi. Unified, minimal and selectively randomizable structure-preserving signatures. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 688–712. Springer, Heidelberg, February 2014.

AHN$^+$17. Masayuki Abe, Dennis Hofheinz, Ryo Nishimaki, Miyako Ohkubo, and Jiaxin Pan. Compact structure-preserving signatures with almost tight security. LNCS, pages 548–580. Springer, Heidelberg, 2017.

AKOT15. Masayuki Abe, Markulf Kohlweiss, Miyako Ohkubo, and Mehdi Tibouchi. Fully structure-preserving signatures and shrinking commitments. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 35–65. Springer, Heidelberg, April 2015.

BBS04. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Heidelberg, August 2004.

BCC$^+$09. Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable proofs and delegatable anonymous credentials. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 108–125. Springer, Heidelberg, August 2009.

BCKL08. Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 356–374. Springer, Heidelberg, March 2008.

BFF$^+$15. Gilles Barthe, Edvard Fagerholm, Dario Fiore, Andre Scedrov, Benedikt Schmidt, and Mehdi Tibouchi. Strongly-optimal structure preserving signatures from type II pairings: Synthesis and lower bounds. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 355–376. Springer, Heidelberg, March / April 2015.

BKP14. Olivier Blazy, Eike Kiltz, and Jiaxin Pan. (Hierarchical) identity-based encryption from affine message authentication. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 408–425. Springer, Heidelberg, August 2014.

BMM17. Christian Badertscher, Christian Matt, and Ueli Maurer. Strengthening access control encryption. In *ASIACRYPT 2017, Part I*, LNCS, pages 502–532. Springer, Heidelberg, December 2017.

BP04. Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 273–289. Springer, Heidelberg, August 2004.

BSZ05. Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 136–153. Springer, Heidelberg, February 2005.

CDM00. Ronald Cramer, Ivan Damgård, and Philip D. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In Hideki Imai and Yuliang Zheng, editors, *PKC 2000*, volume 1751 of *LNCS*, pages 354–372. Springer, Heidelberg, January 2000.

CL03. Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN 02*, volume 2576 of *LNCS*, pages 268–289. Springer, Heidelberg, September 2003.

CL04. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 56–72. Springer, Heidelberg, August 2004.

DHO16. Ivan Damgård, Helene Haagh, and Claudio Orlandi. Access control encryption: Enforcing information flow with cryptography. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 547–576. Springer, Heidelberg, October / November 2016.

DHS15. David Derler, Christian Hanser, and Daniel Slamanig. A new approach to efficient revocable attribute-based anonymous credentials. In Jens Groth, editor, *15th IMA International Conference on Cryptography and Coding*, volume 9496 of *LNCS*, pages 57–74. Springer, Heidelberg, December 2015.

DS16. David Derler and Daniel Slamanig. Fully-anonymous short dynamic group signatures without encryption. Cryptology ePrint Archive, Report 2016/154, 2016. http://eprint.iacr.org/2016/154.

EHK$^+$13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013.

FGKO17. Georg Fuchsbauer, Romain Gay, Lucas Kowalczyk, and Claudio Orlandi. Access control encryption for equality, comparison, and more. LNCS, pages 88–118. Springer, Heidelberg, March 2017.

FHKS16. Georg Fuchsbauer, Christian Hanser, Chethan Kamath, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model from weaker assumptions. In Vassilis Zikas and Roberto De Prisco, editors, *SCN 16*, volume 9841 of *LNCS*, pages 391–408. Springer, Heidelberg, August / September 2016.

FHS14. Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. Cryptology ePrint Archive, Report 2014/944, 2014. http://eprint.iacr.org/2014/944, to appear at *Journal of Cryptology*.

FHS15. Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 233–253. Springer, Heidelberg, August 2015.

Fuc11. Georg Fuchsbauer. Commuting signatures and verifiable encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 224–245. Springer, Heidelberg, May 2011.

Fuc14. Georg Fuchsbauer. Breaking existential unforgeability of a signature scheme from asiacrypt 2014. Cryptology ePrint Archive, Report 2014/892, 2014. http://eprint.iacr.org/2014/892.

Gha16. Essam Ghadafi. Short structure-preserving signatures. In Kazue Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 305–321. Springer, Heidelberg, February / March 2016.

GHKW16. Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Tightly CCA-secure encryption without pairings. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 1–27. Springer, Heidelberg, May 2016.

Gro15. Jens Groth. Efficient fully structure-preserving signatures for large messages. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 239–259. Springer, Heidelberg, November / December 2015.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008.

HK17. Lucjan Hanzlik and Kamil Kluczniak. Two-move and setup-free blind signatures with perfect blindness. In *Proceedings of the 4th ACM International Workshop on ASIA Public-Key Cryptography*, APKC '17, pages 1–11, New York, NY, USA, 2017. ACM.

HRS15. Christian Hanser, Max Rabkin, and Dominique Schröder. Verifiably encrypted signatures: Security revisited and a new construction. In Günther Pernul, Peter Y. A. Ryan, and Edgar R. Weippl, editors, *ESORICS 2015, Part I*, volume 9326 of *LNCS*, pages 146–164. Springer, Heidelberg, September 2015.

HS14. Christian Hanser and Daniel Slamanig. Structure-preserving signatures on equivalence classes and their application to anonymous credentials. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 491–511. Springer, Heidelberg, December 2014.

JR17. Charanjit S. Jutla and Arnab Roy. Improved structure preserving signatures under standard bilinear assumptions. LNCS, pages 183–209. Springer, Heidelberg, March 2017.

KPW15. Eike Kiltz, Jiaxin Pan, and Hoeteck Wee. Structure-preserving signatures from standard assumptions, revisited. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 275–295. Springer, Heidelberg, August 2015.

KW15. Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Heidelberg, April 2015.

KW17. Sam Kim and David J. Wu. Access control encryption for general policies from standard assumptions. In *ASIACRYPT 2017, Part I*, LNCS, pages 471–501. Springer, Heidelberg, December 2017.

KZG10. Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 177–194. Springer, Heidelberg, December 2010.

LPJY15. 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 Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 681–707. Springer, Heidelberg, November / December 2015.

MRV16. Paz Morillo, Carla Ràfols, and Jorge Luis Villar. The kernel matrix Diffie-Hellman assumption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 729–758. Springer, Heidelberg, December 2016.