Threshold Structure-Preserving Signatures

Elizabeth Crites[1], Markulf Kohlweiss[1,2], Bart Preneel[3], Mahdi Sedaghat[3⋆], and Daniel Slamanig[4]

[1] University of Edinburgh, Edinburgh, UK
ecrites@ed.ac.uk, mkohlwei@inf.ed.ac.uk
[2] IOG
[3] imec-COSIC, KU Leuven, Leuven, Belgium
ssedagha@esat.kuleuven.be, bart.preneel@esat.kuleuven.be
[4] AIT Austrian Institute of Technology, Vienna, Austria
daniel.slamanig@ait.ac.at

Abstract. Structure-preserving signatures (SPS) are an important building block for privacy-preserving cryptographic primitives, such as electronic cash, anonymous credentials, and delegatable anonymous credentials. In this work, we introduce the first *threshold* structure-preserving signature scheme (TSPS). This allows multiple parties to jointly sign a message, resulting in a standard, single-party SPS signature, and can thus be used as a replacement for applications based on SPS.

We begin by defining and constructing SPS for indexed messages, which are messages defined relative to a unique index. We prove its security in the random oracle model under a variant of the generalized Pointcheval-Sanders assumption (PS). We then generalize this scheme to an indexed multi-message SPS for signing vectors of indexed messages, which we prove secure under the same assumption. Finally, we formally define the notion of a TSPS and propose a construction based on our indexed multi-message SPS. Our TSPS construction is *fully non-interactive*, meaning that signers simply output partial signatures without communicating with the other signers. Signatures are short: they consist of 2 group elements and require 2 pairing product equations to verify. We prove the security of our TSPS under the security of our indexed multi-message SPS scheme.

Keywords: Threshold Signatures, Structure-Preserving Signatures, Indexed Message Structure-Preserving Signatures.

1 Introduction

Threshold cryptography [DDFY94, Des90, DF90] was designed to reduce the trust in single entities and improve the availability of keying material. It allows a secret key to be shared among a set of parties [Sha79, Bla79] such that the task

⋆ Corresponding author.

involving the key can only be performed if some threshold of them collaborates. Threshold signatures [Sho00, DK01], threshold encryption [SG98, CGJ+99], and threshold verifiable unpredictable functions [GJM+21] enable distributed protocols, such as e-voting systems [CGS97, CFSY96] and multi-party computation [CDN01, DN03].

Threshold signatures in particular have attracted significant interest recently, in part because of advances in distributed ledger technologies, cryptocurrencies, and decentralized identity management [Lin17, DKLs19, CGG+20, KMOS21]. They are also the subject of current standardization efforts by NIST [BDV+20]. Signatures used by certification authorities to issue credentials or to secure digital wallets make attractive targets for misuse or forgery. To mitigate these risks, an (n, t)-threshold signature scheme distributes the signing key among n parties such that any quorum of at least t signers can jointly generate a signature, but the scheme remains secure as long as fewer than t key shares are known to the adversary.

A threshold signature that is *fully non-interactive* consists of a single round of communication. On input the message, each signer computes its partial signature independently of other signers, and aggregation of at least t partial signatures results in a single signature representing the group. Interactive signing protocols involving two or more rounds add complexity and are error prone [TS21, DEF+19]. Thus, fully non-interactive schemes are preferable, the canonical example being threshold BLS [BLS04, Bol03].

Structure-preserving signatures. Structure-preserving signatures (SPS) [AFG+10] are pairing-based signatures where the message, signature, and verification key consist of source group elements only (in one or both groups), and signature verification checks group membership and pairing product equations. SPS have been studied extensively, with a focus on short signatures [AGHO11, AGOT14, Gha16, Gha17], lower bounds [AGHO11, AGO11, AAOT18], and (tight) security under well-known assumptions [ACD+12, HJ12, KPW15, LPY15, JR17, GHKP18, AJO+19].

SPS are compatible with Groth-Sahai non-interactive zero-knowledge proofs (NIZKs) [GS08] and, more generally, help avoid the expensive extraction of exponents in security proofs. This makes them attractive for the modular design of protocols relying on signatures and NIZKs. Indeed, SPS have seen widespread adoption in privacy-preserving applications, such as group signatures [AFG+10, LPY15], traceable signatures [ACHO11], blind signatures [AFG+10, FHS15], attribute-based signatures [EGK14], malleable signatures [ALP12], anonymous credentials [Fuc11, CDHK15, FHS19], delegatable anonymous credentials [BCC+09, CL19], and anonymous e-cash [BCF+11].

For such signature-based applications, compromise of the signing key represents a single point of attack and failure. Replacing the use of SPS with TSPS together with distributed key generation (DKG) would help to reduce the trust in a single authority and increase the availability of the respective signing service. While many of the aforementioned applications of SPS would benefit from thresholdization, until now there was no known threshold construction of SPS that could serve as their basis. We provide the first candidate TSPS scheme as the main contribution of this work.

Towards constructing a threshold SPS. Our goal is to construct threshold SPS that are fully non-interactive, i.e., there is no coordination among signers. This puts some requirements on the used SPS and in particular prevents the use of nonlinear operations of the signing randomness and secret keys (cf. Section 2), which existing SPS fail to satisfy. Thus, as a starting point for our TSPS, we consider the pairing-based Pointcheval-Sanders signature scheme (PS) [PS16] (cf. Section 3.2), as its randomness is simply a random base group element and it avoids hashing during verification. Recall that the signing key is $\mathsf{sk} = (x, y)$ with corresponding verification key $\mathsf{vk} = (\hat{g}^x, \hat{g}^y)$. The signing algorithm takes as input a scalar message $m \in \mathbb{Z}_p$ and outputs a signature

$$\sigma = (h, s) = (g^r, h^{x+my}) \ .$$

Importantly, the nonce r (or equivalently the base h) is sampled fresh for each message. This scheme fails to be an SPS because the message is not a group element (or elements). Ghadafi [Gha16] made the observation that a PS-like SPS scheme can be constructed for a group element message (M_1, M_2) for which there exists a scalar message $m \in \mathbb{Z}_p$ such that $M_1 = g^m$ and $M_2 = \hat{g}^m$. This is referred to as a Diffie-Hellman message [Fuc09, AFG+10]. A Ghadafi SPS signature (cf. Section 3.2) has the form:

$$\sigma = (h, s, t) = (g^r, M_1^r, h^x s^y) \ .$$

Let us see how one might construct a threshold version of this scheme. Suppose each signer possesses a share $\mathsf{sk}_i = (x_i, y_i)$ of the secret key $\mathsf{sk} = (x, y)$. A naive attempt to construct a non-interactive scheme might have each signer output a partial signature of the form:

$$\sigma_i = (h_i, s_i, t_i) = (g^{r_i}, M_1^{r_i}, h_i^{x_i} s_i^{y_i}) \ ,$$

with aggregation of the third term having the form:

$$t = \prod_{i \in \mathcal{T}} t_i^{\lambda_i} = \prod_{i \in \mathcal{T}} g^{r_i x_i \lambda_i} M_1^{r_i y_i \lambda_i} \ ,$$

where λ_i is the Lagrange coefficient for party i in the signing set $\mathcal{T}$ of size at least t. As with other existing SPS, this however does not allow reconstruction via Lagrange interpolation because each term in the exponent is multiplied by a distinct random integer r_i. To overcome this, due to the specific form of the signatures, the signers can agree on a common random element $h = g^r$ via a random oracle. Then each partial signature has the form:

$$\sigma_i = (h, s, t_i) = (g^r, M_1^r, h^{x_i} s^{y_i}) \ .$$

However, M_1^r cannot be computed without knowledge of the discrete logarithm $\mathsf{dlog}_h(M_1)$. To overcome this, we borrow techniques from Sonnino et al. [SAB+19] and Camenisch et al. [CDL+20] to sign *indexed* Diffie-Hellman messages (id, M_1, M_2), a concept that we formalize in this paper. Indexing can be understood as requiring the existence of an injective function f that maps each scalar message $m \in \mathbb{Z}_p$ to an index $id = f(m)$. We then have $h = \mathsf{H}(id)$, where H is modeled as a random oracle, and $M_1 = \mathsf{H}(id)^m$. Then each partial signature has the form:

$$\sigma_i = (h, s_i) = (\mathsf{H}(id), h^{x_i} M_1^{y_i}) \ ,$$

and the aggregated signature has the form:

$$\sigma = (h, s) = (\mathsf{H}(id), h^x M_1^y) \ . \tag{1}$$

This is exactly our TSPS construction, with underlying SPS signature defined by Equation (1). We extend these techniques to vectors of indexed Diffie-Hellman messages $(id, \vec{M}_1, \vec{M}_2)$, which allows additional elements to be signed, such as attributes when used within anonymous credentials [PS16, SAB+19]. Note that the index is not needed for verification (and therefore $\mathsf{H}(id)$ is not computed), so our schemes are indeed structure preserving.

We define an appropriate notion of unforgeability for indexed messages: existential unforgeability under chosen indexed message attack (EUF-CiMA) and prove the security of our constructions under this notion. We discuss various ways of defining the index function, depending on the application. For example, if privacy is not required and the message and public key are known, the index function may simply be the identity function: $id = f(m) = m$, capturing the intuitive notion that each nonce r is associated with a single scalar message m.

Bypassing impossibility results. Our SPS and threshold SPS constructions output signatures that are *unilateral*: they contain elements from only one source group ($\mathbb{G}_1$ in our case). The impossibility of unilateral SPS in the Type-III setting [AGHO11] does not apply to constructions in which the message space is dual in both source groups, and thus, like Ghadafi [Gha16], we sidestep this result.

1.1 Our Contributions

Our contributions can be summarized as follows:

- We formally define the notion of structure-preserving signatures (SPS) over indexed message spaces and corresponding notion of security: existential unforgeability under chosen indexed message attack (EUF-CiMA).
- We propose a concrete SPS construction over indexed Diffie-Hellman messages, called $\mathsf{IM\text{-}SPS}$, and prove its EUF-CiMA security under a new variant

of the generalized Pointcheval-Sanders assumption. We reduce this assumption to the hardness of the $(2, 1)$-discrete logarithm problem in the algebraic group model (AGM).

– We provide an indexed multi-message SPS construction, called $\mathsf{IMM\text{-}SPS}$, which allows vectors of indexed Diffie-Hellman messages to be signed, and prove its EUF-CiMA security.
– We introduce the notion of a threshold structure-preserving signature (TSPS) scheme and propose a fully non-interactive TSPS based on our EUF-CiMA secure SPS scheme. Signatures contain only 2 group elements and verification consists of 2 pairing product equations. We prove the security of our TSPS under the EUF-CiMA security of $\mathsf{IMM\text{-}SPS}$.

2 Related Work

We provide an overview of pairing-based non-interactive threshold signature schemes in Table 1 and structure-preserving signature schemes (SPS) in Table 2 and discuss how these schemes fail to meet our requirements.

Table 1: Table of pairing-based non-interactive threshold signature schemes. iDH refers to indexed Diffie-Hellman messages (Definition 7). ✓: Satisfied. ✗: Not satisfied.

Scheme	Message Space	Sig. Size	Structure-Pres.
BLS [Bol03, BL22]	$\{0,1\}^*$	$1\mathbb{G}_1$	✗
LJY ‡1 [LJY16]	$\{0,1\}^*$	$2\mathbb{G}_1$	✗
LJY ‡2 [LJY16]	$\{0,1\}^*$	$4\mathbb{G}_1 + 2\mathbb{G}_2$	✗
GJMMST [GJM+21]	$\{0,1\}^*$	$4\mathbb{G}_1 + 2\mathbb{G}_2$	✗
PS [SAB+19, TBA+22]	$\mathbb{Z}_p$	$2\mathbb{G}_1$	✗
Our TSPS	iDH	$2\mathbb{G}_1$	✓

Threshold Signatures. BLS [BLS04] and its threshold version [Bol03, BL22] are not structure preserving, as they map bitstring messages $\{0,1\}^*$ to the group using a random oracle. Libert et al. [LJY14, LJY16] propose a secure non-interactive threshold signature scheme based on linearly-homomorphic SPS (LHSPS) [LPJY13]. While this construction meets many of our requirements, the resulting threshold signature is not structure preserving. It either relies on random oracles to hash bitstring messages to group elements (‡1 [LJY16]) or, when avoiding random oracles, a bit-wise encoding of the message is required (‡2 [LJY16]). Gurkan et al. [GJM+21] propose a pairing-based threshold Verifiable Unpredictable Function (VUF), which is essentially a unique threshold signature [MRV99]. However, their construction is not structure preserving: it hashes bitstring messages to the group using a random oracle. Sonnino et al. [SAB+19] and Tomescu et al. [TBA+22] present non-interactive threshold versions of Pointcheval-Sanders (PS) signatures; however, verification takes place over scalar vectors, and is thus not structure preserving. We note that signatures for scalar vectors are intuitively closer to SPS than ones for bitstring messages, as evidenced, for example, by Ghadafi's scheme [Gha16]. We do not know of a general conversion technique, however.

Structure-Preserving Signatures. Most structure-preserving signatures in the literature fail to be good candidates for thresholdization due to nonlinear operations of signer-specific randomness and secret key elements, which are not amenable to Lagrange interpolation (e.g., [AFG+10, AGHO11, AGOT14, BFF+15, Gha17, Gro15]). However, there are two promising approaches: linearly-homomorphic SPS (LHSPS) [LPJY13] and the SPS by Ghadafi [Gha16]. The former is a one-time signature, meaning that a key pair can only sign a single message. The SPS by Ghadafi [Gha16] lends itself to thresholdization, but does not yield a non-interactive TSPS scheme because there is no common base h for the signers.

Table 2: Table of structure-preserving signature schemes (SPS). DH refers to Diffie-Hellman messages (Definition 2), and iDH refers to indexed Diffie-Hellman messages (Definition 7). ✓: Satisfied. ✗: Not satisfied.

Scheme	Message Space	Sig. Size	Avoids Nonlinearity
AFGHO[AFG+10]	$\mathbb{G}_1$	$5\mathbb{G}_1 + 2\mathbb{G}_2$	✗
AGHO [AGHO11]	$\mathbb{G}_1 \times \mathbb{G}_2$ / $\mathbb{G}_2$	$2\mathbb{G}_1 + 1\mathbb{G}_2$	✗
AGOT [AGOT14]	$\mathbb{G}_1$	$2\mathbb{G}_1 + 1\mathbb{G}_2$	✗
BFFSST [BFF+15]	$\mathbb{G}_2$	$1\mathbb{G}_1 + 2\mathbb{G}_2$	✗
Ghadafi [Gha17]	DH	$2\mathbb{G}_1$	✗
Ghadafi [Gha16]	DH	$3\mathbb{G}_1$	✗
Groth [Gro15]	$\mathbb{G}_2$	$1\mathbb{G}_1 + 2\mathbb{G}_2$	✗
LPJY [LPJY13]*	$\mathbb{G}_1$	$2\mathbb{G}_1$	✓
Our SPS	iDH	$2\mathbb{G}_1$	✓

*One-time: a key pair can only sign a single message.

3 Preliminaries

3.1 General

Let $\kappa \in \mathbb{N}$ denote the security parameter and 1^κ its unary representation. Let p be a κ-bit prime. For all positive polynomials $f(\kappa)$, a function $\nu : \mathbb{N} \to \mathbb{R}^+$ is called *negligible* if $\exists\ \kappa_0 \in \mathbb{N}$ such that $\forall\ \kappa > \kappa_0$ it holds that $\nu(\kappa) < 1/f(\kappa)$. We denote by $\mathbb{G}^*$ the set $\mathbb{G} \setminus 1_{\mathbb{G}}$, where $1_{\mathbb{G}}$ is the identity element of the group $\mathbb{G}$.

We denote the group of integers mod p by $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$, its multiplicative group of units by $\mathbb{Z}_p^*$, and the polynomial ring over $\mathbb{Z}_p$ by $\mathbb{Z}_p[X]$. For a group $\mathbb{G}$ of order p with generator g, we denote the discrete logarithm $m \in \mathbb{Z}_p$ of $M \in \mathbb{G}$ base g by $\mathsf{dlog}_g(M)$ (i.e., $M = g^m$). We denote the set of integers $\{1, \ldots, n\}$ by $[1, n]$ and the vector A by $\vec{A}$. Let $Y \leftarrow\$ F(X)$ denote running probabilistic algorithm F on input X and assigning its output to Y. Let $x \leftarrow\$ \mathbb{Z}_p$ denote sampling an element of $\mathbb{Z}_p$ uniformly at random. All algorithms are randomized unless expressly stated otherwise. PPT refers to probabilistic polynomial time. We denote the output of a security game $G^{\textsc{Game}}$ between a challenger and a PPT adversary $\mathcal{A}$ by $G_{\mathcal{A}}^{\textsc{Game}}$, where $\mathcal{A}$ wins the game if $G_{\mathcal{A}}^{\textsc{Game}} = 1$.

Definition 1 (Bilinear Group). *A bilinear group generator $\mathcal{BG}(1^\kappa)$ returns a tuple $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g})$ such that $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are finite groups of the same prime order p, $g \in \mathbb{G}_1$ and $\hat{g} \in \mathbb{G}_2$ are generators, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear pairing, which satisfies the following:*

1. *$e(g, \hat{g}) \neq 1_{\mathbb{G}_T}$ (non-degeneracy).*
2. *$\forall\, a, b \in \mathbb{Z}_p$, $e(g^a, \hat{g}^b) = e(g, \hat{g})^{ab} = e(g^b, \hat{g}^a)$ (bilinearity).*
3. *e is efficiently computable.*

We rely on bilinear groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with no efficiently computable isomorphism between them [GPS08], also called Type-III or asymmetric bilinear groups. To date, they are the most efficient choice for relevant security levels.

Definition 2 (Diffie-Hellman Message Space [Fuc09, AFG+10]). *Over an asymmetric bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g})$, a pair $(M_1, M_2) \in \mathbb{G}_1 \times \mathbb{G}_2$ belongs to the Diffie-Hellman (DH) message space $\mathcal{M}_{\mathsf{DH}}$ if there exists $m \in \mathbb{Z}_p$ such that $M_1 = g^m$ and $M_2 = \hat{g}^m$.*

One can efficiently verify whether $(M_1, M_2) \in \mathcal{M}_{\mathsf{DH}}$ by checking $e(M_1, \hat{g}) = e(g, M_2)$.

Definition 3 (Algebraic Group Model [FKL18]). *An adversary is algebraic if for every group element $h \in \mathbb{G} = \langle g \rangle$ that it outputs, it is required to output a representation $\vec{h} = (\eta_0, \eta_1, \eta_2, \ldots)$ such that $h = g^{\eta_0} \prod {g_i}^{\eta_i}$, where $g, g_1, g_2, \cdots \in \mathbb{G}$ are group elements that the adversary has seen thus far.*

The original definition of the algebraic group model (AGM) [FKL18] only captures regular cyclic groups $\mathbb{G} = \langle g \rangle$. Mizuide et al. [MTT19] extend this definition to include symmetric pairing groups ($\mathbb{G}_1 = \mathbb{G}_2$), such that the adversary is also allowed to output target group elements (in $\mathbb{G}_T$) and their representations. Recently, Couteau and Hartmann [CH20] defined the Algebraic Asymmetric Bilinear Group Model, which extends the AGM definition for asymmetric pairings by allowing the adversary to output multiple elements from all three groups. The definition can be found in Appendix A.

3.2 Schemes

Pointcheval-Sanders Signatures [PS16]. The PS signature scheme is defined over the message space $\mathcal{M}$ of scalar messages $m \in \mathbb{Z}_p$ and consists of the following PPT algorithms:

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$: Compute $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$. Output pp.
- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}(\mathsf{pp})$: Sample $x, y \leftarrow\$\, \mathbb{Z}_p^*$ and set $\mathsf{sk} = (\mathsf{sk}_1, \mathsf{sk}_2) = (x, y)$ and $\mathsf{vk} = (\mathsf{vk}_1, \mathsf{vk}_2) = (\hat{g}^x, \hat{g}^y)$. Output $(\mathsf{sk}, \mathsf{vk})$.
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, m)$: Sample $r \leftarrow\$\, \mathbb{Z}_p^*$ and compute $\sigma = (h, s) = (g^r, h^{x+my})$. Output σ.
- $0/1 \leftarrow \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, m, \sigma)$: If $h \in \mathbb{G}_1, h \neq 1_{\mathbb{G}_1}$, and the pairing product equation $e(h, \mathsf{vk}_1 \mathsf{vk}_2^m) = e(s, \hat{g})$ holds, output 1 (accept); else, output 0 (reject).

Pointcheval-Sanders signatures are EUF-CMA secure under the PS assumption (Definition 5) [PS16].

Ghadafi SPS [Gha16]. The Ghadafi structure-preserving signature scheme is defined over the message space $\mathcal{M}_{\mathsf{DH}}$ of Diffie-Hellman pairs $(M_1, M_2) \in \mathbb{G}_1 \times \mathbb{G}_2$ such that $e(M_1, \hat{g}) = e(g, M_2)$ and consists of the following PPT algorithms:

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$: Compute $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$. Output pp.
- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}(\mathsf{pp})$: Sample $x, y \leftarrow\$\, \mathbb{Z}_p^*$ and set $\mathsf{sk} = (\mathsf{sk}_1, \mathsf{sk}_2) = (x, y)$ and $\mathsf{vk} = (\mathsf{vk}_1, \mathsf{vk}_2) = (\hat{g}^x, \hat{g}^y)$. Output $(\mathsf{sk}, \mathsf{vk})$.
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, M_1, M_2)$: Sample $r \leftarrow\$\, \mathbb{Z}_p^*$ and compute $\sigma = (h, s, t) = (g^r, M_1^r, h^x s^y)$. Output σ.
- $0/1 \leftarrow \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \sigma, M_1, M_2)$: If $h, s, t \in \mathbb{G}_1, h \neq 1_{\mathbb{G}_1}$, and both pairing product equations $e(h, M_2) = e(s, \hat{g})$ and $e(t, \hat{g}) = e(h, \mathsf{vk}_1) e(s, \mathsf{vk}_2)$ hold, output 1 (accept); else, output 0 (reject).

The Ghadafi SPS is weakly EUF-CMA secure in the generic group model (GGM) [Gha16].

Shamir Secret Sharing [Sha79]. An (n, t)-Shamir secret sharing divides a secret s among n shareholders such that each subset of at least t shareholders can reconstruct s, but fewer than t cannot (and s remains information-theoretically hidden). A dealer who knows the secret s forms a polynomial $f(x)$ of degree t with randomly chosen coefficients from $\mathbb{Z}_p$ such that $f(0) = s$. The dealer then securely provides each shareholder with $s_i = f(i), i \in [1, n]$. Let $\vec{s} \leftarrow\$\, \mathsf{Share}(s, p, n, t)$ denote the process of computing shares $\vec{s} = (s_1, \ldots, s_n)$ of a secret s. Each subset $\mathcal{T} \subset [1, n]$ of size at least t can pool their shares to reconstruct the secret s using Lagrange interpolation, as $s = f(0) = \sum_{i \in \mathcal{T}} s_i \lambda_i$, where $\lambda_i = \prod_{j \in \mathcal{T}, j \neq i} \frac{j}{j-i}$.

3.3 Assumptions

Definition 4 ($(2,1)$-Discrete Logarithm Assumption [BFL20]). *Let $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$ be an asymmetric bilinear group. The $(2,1)$-discrete logarithm assumption holds with respect to $\mathcal{BG}$ if for all PPT adversaries $\mathcal{A}$, there exists a negligible function ν such that* $\Pr\left[z \leftarrow\!\$\, \mathbb{Z}_p^*;\ (Z, Z', \hat{Z}) \leftarrow (g^z, g^{z^2}, \hat{g}^z);\ z' \leftarrow\!\$\, \mathcal{A}(\mathcal{BG}, Z, Z', \hat{Z}) :\ z' = z\right] < \nu(\kappa)$.

Definition 5 (PS Assumption [PS16]). *Let the advantage of an adversary $\mathcal{A}$ against the PS game $\mathbf{G}^{PS}$, as defined in Figure 1, be as follows:*

$$Adv_{\mathcal{A}}^{PS}(\kappa) = \Pr\left[\mathbf{G}_{\mathcal{A}}^{PS} = 1\right] \ .$$

The PS assumption holds if for all PPT adversaries $\mathcal{A}$, there exists a negligible function ν such that $Adv_{\mathcal{A}}^{PS}(\kappa) < \nu(\kappa)$.

$\mathbf{G}^{\mathrm{PS}}(1^\kappa)$	$\mathcal{O}^{\mathrm{PS}}(m)//\ m \in \mathbb{Z}_p$
$\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$	$h \leftarrow\!\$\, \mathbb{G}_1$
$x, y \leftarrow\!\$\, \mathbb{Z}_p^*$	$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
$(m^*, h^*, s^*) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathrm{PS}}}(\mathsf{pp}, \hat{g}^x, \hat{g}^y)$	**return** (h, h^{x+my})
return $((1)\ h^* \neq 1_{\mathbb{G}_1} \wedge\ m^* \neq 0 \wedge$	
$(2)\ s^* = {h^*}^{x+m^*y} \wedge$	
$(3)\ m^* \notin \mathcal{Q})$	

Fig. 1: Game defining the PS assumption.

The validity of the tuple (m^*, h^*, s^*) is decidable by checking $e(s^*, \hat{g}) = e(h^*, \hat{g}^x(\hat{g}^y)^{m^*})$. The PS assumption is an interactive assumption defined by Pointcheval and Sanders [PS16] to construct an efficient randomizable signature. The assumption has been shown to hold in the Generic Group Model (GGM).

Kim et al. [KLAP20] introduce a generalized version of the PS assumption (GPS) that splits the PS oracle $\mathcal{O}^{\mathrm{PS}}(\cdot)$ into two oracles $\mathcal{O}_0^{\mathrm{GPS}}(), \mathcal{O}_1^{\mathrm{GPS}}(\cdot)$: the first samples $h \leftarrow\!\$\, \mathbb{G}_1$, and the second takes h and m as input and generates the PS value h^{x+my}. Recently, Kim et al. [KSAP22] extended the GPS assumption (GPS_2), replacing field element inputs, such as m, with group element inputs. The GPS_2 assumption holds under the $(2,1)$-DL assumption (Definition 4) in the algebraic group model. Both the GPS and GPS_2 assumptions can be found in Appendix A.2.

Owing to the fact that our SPS and TSPS constructions rely on a different message space, we introduce an analogous generalized PS assumption (GPS_3), defined as follows.

Definition 6 (GPS_3 **Assumption).** *Let the advantage of an adversary $\mathcal{A}$ against the GPS_3 game $\mathbf{G}^{GPS_3}$, as defined in Figure 2, be as follows:*

$$Adv_{\mathcal{A}}^{GPS_3}(\kappa) = \Pr\left[\mathbf{G}_{\mathcal{A}}^{GPS_3} = 1\right] \ .$$

The GPS_3 assumption holds if for all PPT adversaries $\mathcal{A}$, there exists a negligible function ν such that $Adv_{\mathcal{A}}^{GPS_3}(\kappa) < \nu(\kappa)$.

$\mathbf{G}^{\mathrm{GPS}_3}(1^\kappa)$

$$
\begin{array}{l}
\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa) \\
x, y \leftarrow\$\, \mathbb{Z}_p^* \\
(M_1^*, M_2^*, h^*, s^*) \leftarrow \mathcal{A}^{\mathcal{O}_0^{\mathrm{GPS}_3}, \mathcal{O}_1^{\mathrm{GPS}_3}}(\mathsf{pp}, \hat{g}^x, \hat{g}^y) \\
\textbf{return } ((1)\ M_1^* \neq 1_{\mathbb{G}_1} \wedge\ h^* \neq 1_{\mathbb{G}_1} \wedge \\
\qquad\qquad (2)\ s^* = {h^*}^x {M_1^*}^y \wedge \\
\qquad\qquad (3)\ \mathsf{dlog}_{h^*}(M_1^*) = \mathsf{dlog}_{\hat{g}}(M_2^*) \wedge \\
\qquad\qquad (4)\ (\star, M_2^*) \notin \mathcal{Q}_1)
\end{array}
$$

$\mathcal{O}_0^{\mathrm{GPS}_3}()$

$$
\begin{array}{l}
r \leftarrow\$\, \mathbb{Z}_p^* \\
\mathcal{Q}_0 \leftarrow \mathcal{Q}_0 \cup \{g^r\} \\
\textbf{return } g^r
\end{array}
$$

$\mathcal{O}_1^{\mathrm{GPS}_3}(h, M_1, M_2) // M_1 \in \mathbb{G}_1, M_2 \in \mathbb{G}_2$

$$
\begin{array}{l}
\textbf{if } (h \notin \mathcal{Q}_0 \vee \mathsf{dlog}_h(M_1) \neq \mathsf{dlog}_{\hat{g}}(M_2)) : \\
\quad \textbf{return } \perp \\
\textbf{if } (h, \star) \in \mathcal{Q}_1 : \\
\quad \textbf{return } \perp \\
\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{(h, M_2)\} \\
\textbf{return } (h^x M_1^y)
\end{array}
$$

Fig. 2: Game defining our GPS_3 assumption.

4 Indexed Message Structure-Preserving Signatures

We introduce the notion of structure-preserving signatures (SPS) on indexed messages as well as a corresponding notion of security: unforgeability against chosen indexed message attack (EUF-CiMA). We provide an indexed message SPS construction, called IM-SPS, and prove its EUF-CiMA security under the GPS_3 assumption (Definition 6) in the random oracle model (ROM) (Theorem 2). We also propose an indexed *multi*-message SPS construction, called IMM-SPS, which allows vectors of indexed messages to be signed, and prove its EUF-CiMA security under the same assumptions (Theorem 3). IMM-SPS are useful for applications where additional elements, such as attributes, are signed.

Indexing can be understood as requiring the existence of an injective function f that maps each message to an index. We model this by requiring that for all index/message pairs in an indexed message space $\mathcal{M}$, the following uniqueness property holds: $(id, \tilde{M}) \in \mathcal{M}$, $(id', \tilde{M}') \in \mathcal{M}$, $id = id' \Rightarrow \tilde{M} = \tilde{M}'$. That is, no two messages use the same index. We refer to index/message pairs as $M = (id, \tilde{M})$.

Indexing is useful, as signatures can depend on the index; for example, in our schemes, signing involves evaluating a hash-to-curve function H on the index to obtain a base element $h \leftarrow \mathsf{H}(id)$. Verifying a message/signature pair does not require availability of the index, making it structure preserving. Consequently, the verification message space $\tilde{\mathcal{M}}$ is obtained from $\mathcal{M}$ by omitting the index.

For our schemes, we need to consider that in verification one can provide a base element h^r obtained by randomizing the original base element h. This is due to the partial randomizability of the signatures. Thus, different messages $\tilde{M}, \tilde{M}'$ may be valid representations for the same scalar message m. Consequently, similar to SPS on equivalence classes (SPS-EQ) [FHS19], the verification message space $\tilde{\mathcal{M}}$ is expanded to consider equivalent (randomized) messages: $\tilde{\mathcal{M}} = \{\tilde{M} \mid \exists\ (\cdot, \tilde{M}') \in \mathcal{M}\ ,\ \tilde{M} \in \mathsf{EQ}(\tilde{M}')\}$. The function EQ depends on the concrete message space and determines the respective set of equivalent messages.

Next, we define the indexed Diffie-Hellman message space used by our $\mathsf{IM\text{-}SPS}$ scheme (cf. Figure 3 for its encoding function).

Definition 7 (Indexed Diffie-Hellman Message Space). *Given an asymmetric bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$, an index set $\mathcal{I}$, and a random oracle $\mathsf{H} : \mathcal{I} \rightarrow \mathbb{G}_1$, $\mathcal{M}^{\mathsf{H}}_{\mathsf{iDH}}$ is an indexed Diffie-Hellman (DH) message space if $\mathcal{M}^{\mathsf{H}}_{\mathsf{iDH}} \subset \{(id, \tilde{M}) \mid id \in \mathcal{I}, m \in \mathbb{Z}_p, \tilde{M} = (\mathsf{H}(id)^m, \hat{g}^m) \in \mathbb{G}_1 \times \mathbb{G}_2\}$ and the following index uniqueness property holds: for all $(id, \tilde{M}) \in \mathcal{M}^{\mathsf{H}}_{\mathsf{iDH}}$, $(id', \tilde{M}') \in \mathcal{M}^{\mathsf{H}}_{\mathsf{iDH}}$, $id = id' \Rightarrow \tilde{M} = \tilde{M}'$.*

We define the equivalence class for each message $\tilde{M} = (M_1, M_2) \in \tilde{\mathcal{M}}^{\mathsf{H}}_{\mathsf{iDH}}$, as $\mathsf{EQ}_{\mathsf{iDH}}(M_1, M_2) = \{(M_1^r, M_2) \mid \exists\ r \in \mathbb{Z}_p\}$.

$\mathsf{iDH}^{\mathsf{H}}(id, m)$	$\mathsf{H}(id)$
$h \leftarrow \mathsf{H}(id)$	**if** $\mathcal{Q}_{\mathsf{H}}[id] = \perp$:
$\tilde{M} \leftarrow (h^m, \hat{g}^m)$	$\quad \mathcal{Q}_{\mathsf{H}}[id] \leftarrow\$\ \mathbb{G}_1$
return $(id, \tilde{M})$	**return** $\mathcal{Q}_{\mathsf{H}}[id]$

Fig. 3: Encoding function of indexed Diffie-Hellman message space in the ROM.

The subset membership is efficiently decidable by checking $e(M_1, \hat{g}) = e(h, M_2)$ for $h \leftarrow \mathsf{H}(id)$. Note that, in addition, one needs to guarantee that no two messages use the same index. This is the responsibility of the signer.[5] As mentioned above, messages $\tilde{M}$ lie in a different verification message space $\tilde{\mathcal{M}}^{\mathsf{H}}_{\mathsf{iDH}}$ that is uniquely determined by $\mathcal{M}^{\mathsf{H}}_{\mathsf{iDH}}$ and $\mathsf{EQ}_{\mathsf{iDH}}$. Note that most $\tilde{M} \in \tilde{\mathcal{M}}^{\mathsf{H}}_{\mathsf{iDH}}$ are not indexed Diffie-Hellman messages. In particular, when expanding the definition of $\mathsf{EQ}_{\mathsf{iDH}}$, the verification message space is $\tilde{\mathcal{M}}^{\mathsf{H}}_{\mathsf{iDH}} = \{(M_1^r, M_2) \mid \exists\ r \in \mathbb{Z}_p\ ,\ \exists\ (\cdot, M_1, M_2) \in \mathcal{M}^{\mathsf{H}}_{\mathsf{iDH}}\}$.

Does $\tilde{M}$ depend on id or does id depend on $\tilde{M}$? One might observe the above apparent circularity with respect to the indexing technique. On the one hand, we require existence of an injective function f that maps (M_1, M_2) to id. On the other hand, M_1 is computed as $M_1 = \mathsf{H}(id)^{\mathsf{dlog}_{\hat{g}}(M_2)}$. This circularity is avoided by computing id from the partial message M_2, or more commonly its discrete logarithm m.

$$\underbrace{(\star, M_2) \xrightarrow{\mathsf{dlog}_{\hat{g}}(M_2)} m \in \mathbb{Z}_p \xrightarrow{f} id}_{\text{Message Indexing}} \overbrace{\xrightarrow{\mathsf{iDH}^{\mathsf{H}}(id,m)} (id, M_1, M_2) \in \mathcal{M}^{\mathsf{H}}_{\mathsf{iDH}}}^{\text{Indexed DH message space in ROM}}$$

Fig. 4: Towards an Indexed Message SPS.

As illustrated in Figure 4, the indexing function f assigns an index id to each scalar message $m \in \mathbb{Z}_p$. We then use a hash-to-curve function $\mathsf{H} : \{0,1\}^* \to \mathbb{G}_1$ (modeled as a random oracle) to generate a unique base element h. A source group message (M_1, M_2) can then be obtained using h. In an indexed message SPS, the signing algorithm takes as input the source group message together with an index and generates the underlying signature with access to H. Note that the index does not destroy the structure since the verifier does not need to know id to verify a signature on message $\tilde{M} = (M_1, M_2)$.

Indexing function instantiations. Depending on the application, the indexing function f can be instantiated differently. For example, if messages and signatures are allowed to be public, the indexing function can be instantiated by using the scalar message m itself as the index: $f(m) \leftarrow m = id$.

If message and signatures must be hidden, as in the case of applications to anonymous credentials, one can take the approach of committing to the scalar message and providing a proof of well-formedness of the commitment, as done by Sonnino et al. [SAB+19]. As it is impossible to open a well-formed commitment to two different messages, this guarantees uniqueness of the index. Camenisch et al. [CDL+20] take yet another approach for indexing messages: they assume the existence of a pre-defined and publicly available indexing function. That is, there is a unique index value for each message that is known to all signers. The

[5] To highlight this responsibility, we enforce uniqueness both in the message space, and later on in Line 1 of the of $\mathcal{O}_{\mathsf{Sign}}(\cdot)$ oracle of Figure 5.

corresponding base element can be obtained by evaluating the hash-to-curve function at the given index. As the authors note, if the size of the message space is polynomial and known in advance, then this approach is secure, since it is equivalent to including the base element in the public parameters. However, this is impractical for large message spaces.

4.1 Definition of Unforgeability for Indexed Message SPS

We adapt the notion of EUF-CMA security (Appendix A) [GMR88] for digital signatures to existential unforgeability against chosen *indexed* message attack (EUF-CiMA). In the security game, the adversary can make queries to the signing oracle by providing index/message pairs.

In our unforgeability definition, we expand the set of signed messages $\mathcal{Q}_{\mathsf{S}} = \{(id_i, \tilde{M}_i)\}_i$ to the set of trivially forgeable messages $\mathcal{Q}_{\mathsf{EQ}} = \{\mathsf{EQ}(\tilde{M}_i)\}_i$ and use it in the validity condition of the adversary.

Definition 8 (Existential Unforgeability under Chosen Indexed Message Attack (EUF-CiMA)). *A digital signature scheme over indexed message space $\mathcal{M}$ is EUF-CiMA secure if for all PPT adversaries $\mathcal{A}$ playing game $\mathbf{G}^{\textsf{EUF-CiMA}}$ (Figure 5), there exists a negligible function ν such that*

$$Adv_{\mathcal{A}}^{\textsf{EUF-CiMA}}(\kappa) = \Pr\left[\mathbf{G}_{\mathcal{A}}^{\mathsf{EUF\text{-}CiMA}}(1^\kappa) = 1\right] \leq \nu(\kappa) \enspace .$$

$\mathbf{G}_{\mathcal{A}}^{\text{EUF-CiMA}}(1^\kappa)$

$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$
$(\mathsf{sk}, \mathsf{vk}) \leftarrow\$ \mathsf{KGen}(\mathsf{pp})$
$\left(\tilde{M}^*, \sigma^*\right) \leftarrow\$ \mathcal{A}^{\mathsf{H}, \mathcal{O}_{\mathsf{Sign}}}(\mathsf{pp}, \mathsf{vk})$
return $\left(\tilde{M}^* \notin \mathcal{Q}_{\mathsf{EQ}} \wedge \right.$
$\left. \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \tilde{M}^*, \sigma^*)\right)$

$\mathcal{O}_{\mathsf{Sign}}(id, \tilde{M})$

if $(id, \star) \in \mathcal{Q}_{\mathsf{S}}$:
  return $\perp$
else :
  $\sigma = \mathsf{Sign}\left(\mathsf{pp}, \mathsf{sk}, (id, \tilde{M})\right)$
  $\mathcal{Q}_{\mathsf{S}} \leftarrow \mathcal{Q}_{\mathsf{S}} \cup \{(id, \tilde{M})\}$
  $\mathcal{Q}_{\mathsf{EQ}} \leftarrow \mathcal{Q}_{\mathsf{EQ}} \cup \{\mathsf{EQ}(\tilde{M})\}$
  return σ

Fig. 5: Game $\mathbf{G}_{\mathcal{A}}^{\text{EUF-CiMA}}(1^\kappa)$.

4.2 Our Indexed Message SPS

In Figure 6, we present our indexed message SPS construction IM-SPS over the indexed Diffie-Hellman message space $\mathcal{M}_{\mathsf{iDH}}^{\mathsf{H}}$.

$\mathsf{Setup}(1^\kappa)$	$\mathsf{KGen}(\mathsf{pp})$
$(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$	Parse pp
$\mathsf{pp} \leftarrow (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g})$	$x, y \leftarrow\!\$\, \mathbb{Z}_p^*$
return pp	$\mathsf{sk} \leftarrow (\mathsf{sk}_1, \mathsf{sk}_2) = (x, y)$
	$\mathsf{vk} \leftarrow (\mathsf{vk}_1, \mathsf{vk}_2) = (\hat{g}^x, \hat{g}^y)$
	return $(\mathsf{sk}, \mathsf{vk})$

$\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, (id, M_1, M_2))$	$\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, (M_1, M_2), \sigma)$
Parse $(\mathsf{pp}, \mathsf{sk})$	// does not invoke H
$h \leftarrow\!\$\, \mathsf{H}(id)$	Parse $(\mathsf{pp}, \mathsf{vk}, \sigma)$
if $e(h, M_2) = e(M_1, \hat{g})$:	**return** $\big(h \neq 1_{\mathbb{G}_1} \wedge M_1 \neq 1_{\mathbb{G}_1} \wedge$
$\quad s = h^{\mathsf{sk}_1} M_1^{\mathsf{sk}_2}$	$e(h, M_2) = e(M_1, \hat{g}) \wedge$
$\quad$ **return** $\sigma \leftarrow (h, s)$	$e(h, \mathsf{vk}_1) e(M_1, \mathsf{vk}_2) = e(s, \hat{g})\big)$
else :	
$\quad$ **return** $\perp$	

Fig. 6: Our Indexed Message SPS Construction IM-SPS.

4.3 Security of IM-SPS

We now prove that our proposed IM-SPS construction (Figure 6) is EUF-CiMA secure under the GPS_3 assumption in the random oracle model. We begin by defining the GPS_3 assumption in the algebraic group model (Figure 15), which we reduce to the hardness of the $(2,1)$-DL problem (Definition 4). Then, based on the GPS_3 assumption, we show the security of IM-SPS and IMM-SPS. Our security reductions from IM-SPS and IMM-SPS to GPS_3 are tight. Finally, we show the tight security of our TSPS under the security of IMM-SPS.

Figure 7 defines a roadmap for our IM-SPS, IMM-SPS, and TSPS constructions and their underlying assumptions.

$$
\begin{tikzcd}
(2,1)\text{-}\mathsf{DL} \arrow[r, "\text{Thm. 1 (AGM)}"] & \mathsf{GPS}_3 \arrow[r, "\text{Thm. 2 (ROM)}"] \arrow[dr, bend right, "\text{Thm. 3 (ROM)}"] & \mathsf{IM\text{-}SPS} & \\
& & \mathsf{IMM\text{-}SPS} \arrow[r, "\text{Thm. 4 (ROM)}"] & \mathsf{TSPS}
\end{tikzcd}
$$

Fig. 7: The proposed constructions and underlying assumptions.

Theorem 1. *The* GPS_3 *assumption (Definition 6) holds in the asymmetric algebraic bilinear group model (Definition 18) under the hardness of the* $(2,1)$*-DL problem (Definition 4).*

Proof Outline. To prove this theorem, we borrow the proof technique of Kim et al. [KSAP22, Theorem 2] and define a challenger $\mathcal{B}_{alg}$ who can simulate the defined oracles in the GPS_3 game in the AGM. The defined extractor can successfully extract the scalar message m_j on the j^{th} query to the $\mathcal{O}_1^{\mathrm{GPS}_3}(.)$ oracle by having access to the representations of inputs to the oracle. We show that if the extractor fails, then we can build an algebraic algorithm to solve the $(2,1)$-DL problem. We then demonstrate that no algebraic adversary $\mathcal{A}_{alg}$ can produce a valid output that satisfies all the conditions in the security game. Note that the GPS_3 assumption and the GPS_2 assumption defined by Kim et al. [KSAP22, Theorem 2] (Appendix A) seem incomparable since messages consist of elements from both source groups.

The full proof can be found in Appendix B.1.

Theorem 2. *The indexed message SPS scheme* IM-SPS *(Figure 6) is correct and* EUF-CiMA *secure (Definition 8) under the* GPS_3 *assumption (Definition 6) in the random oracle model.*

We first present an attack to motivate the need for uniqueness in the indexed message space. Assume there were no uniqueness requirement, and suppose the redundant check in Line 1 of the of $\mathcal{O}_{\mathsf{Sign}}(\cdot)$ oracle of Figure 5 were not present. Then, a forger could obtain two signatures $s = h^x M_1^y$, $s' = h^x {M_1'}^y$ and compute a forgery $s^* = s^2/s = h^x(M_1^2/M_1')^y$.

Proof Outline. Let $\mathcal{A}$ be a PPT adversary against the EUF-CiMA security of IM-SPS. We construct a PPT reduction $\mathcal{B}$ against the GPS_3 assumption as follows. When $\mathcal{A}$ queries the random oracle H on a fresh id, $\mathcal{B}$ queries its oracle $\mathcal{O}_0^{\mathrm{GPS}_3}()$ to obtain a random base element h, which it memorizes and returns to $\mathcal{A}$. When $\mathcal{A}$ queries its signing oracle $\mathcal{O}_{\mathsf{Sign}}(\cdot)$ on (id, M_1, M_2), $\mathcal{B}$ looks up $h = \mathsf{H}(id)$ and queries its oracle $\mathcal{O}_1^{\mathrm{GPS}_3}(\cdot)$ on (h, M_1, M_2) to receive $h^x M_1^y$. Finally, $\mathcal{B}$ returns the signature $\sigma = (h, h^x M_1^y)$ to $\mathcal{A}$. $\mathcal{B}$ correctly simulates the EUF-CiMA game, and the success probability of $\mathcal{A}$ and $\mathcal{B}$ is the same.

The attack above would violate the condition $(h, \star) \notin \mathcal{Q}_1$ in Line 3 of the $\mathcal{O}_1^{\mathrm{GPS}_3}(\cdot)$ oracle in Figure 2.

The full proof can be found in Appendix B.2.

4.4 Our Indexed Multi-Message SPS

Next, we extend our IM-SPS construction to an indexed *multi*-message SPS construction IMM-SPS, which allows vectors of indexed messages to be signed, and prove its EUF-CiMA security. Extending the message space to allow vectors of any length is desirable for applications in which several attributes may be signed. The number of pairings required for verification scales linearly with the length of the message vectors, but signatures remain constant sized (2 group elements).

We first generalize the notion of an indexed message space to the multi-message setting. In Figure 8, we present the encoding function $\mathsf{MiDH}^{\mathsf{H}}(id, \vec{m})$ of a multi-message variant of the indexed Diffie-Hellman message space that maps, for any $\ell > 1$, ℓ-scalar message vectors $\vec{m} = (m_1, \ldots, m_\ell) \in \mathbb{Z}_p^\ell$ to 2ℓ-source group message vectors $(\vec{M}_1, \vec{M}_2) = ((M_{11}, \ldots, M_{1\ell}), (M_{21}, \ldots, M_{2\ell})) \in \mathbb{G}_1^\ell \times \mathbb{G}_2^\ell$ based on a given index id.

Definition 9 (Indexed Diffie-Hellman Multi-Message Space). *Given an asymmetric bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$, an index set $\mathcal{I}$, and a random oracle $\mathsf{H} : \mathcal{I} \to \mathbb{G}_1$, $\mathcal{M}^{\mathsf{H}}_{\mathsf{MiDH}}$ is an indexed Diffie-Hellman (DH) message space if $\mathcal{M}^{\mathsf{H}}_{\mathsf{MiDH}} \subset \{(id, \tilde{M}) \mid id \in \mathcal{I}, \vec{m} \in \mathbb{Z}_p^\ell, \tilde{M} = \mathsf{MiDH}^{\mathsf{H}}(id, \vec{m})\}$ and the following index uniqueness property holds: for all $(id, \tilde{M}) \in \mathcal{M}^{\mathsf{H}}_{\mathsf{MiDH}}$, $(id', \tilde{M}') \in \mathcal{M}^{\mathsf{H}}_{\mathsf{MiDH}}$, $id = id' \Rightarrow \tilde{M} = \tilde{M}'$.*

We define the equivalence class for each multi-message $\tilde{M} = (\vec{M}_1, \vec{M}_2) \in \tilde{\mathcal{M}}^{\mathsf{H}}_{\mathsf{MiDH}}$ as $\mathsf{EQ}_{\mathsf{MiDH}}(\vec{M}_1, \vec{M}_2) = \{(\vec{M}_1^r, \vec{M}_2) \mid \exists\, r \in \mathbb{Z}_p\}$.

$\mathsf{MiDH}^{\mathsf{H}}(id, \vec{m})$	$\mathsf{H}(id)$
$h \leftarrow \mathsf{H}(id)$	**if** $\mathcal{Q}_{\mathsf{H}}[id] = \perp$:
for $j \in [1, \ell]$:	$\quad \mathcal{Q}_{\mathsf{H}}[id] \leftarrow\!\$\; \mathbb{G}_1$
$\quad M_{1j} \leftarrow h^{m_j}, M_{2j} \leftarrow \hat{g}^{m_j}$	**return** $\mathcal{Q}_{\mathsf{H}}[id]$
return $\left(id, (\vec{M}_1, \vec{M}_2)\right)$	

Fig. 8: Encoding function of indexed Diffie-Hellman multi-message space in the ROM.

This generalization of the indexed Diffie-Hellman message space leads us to an indexed multi-message SPS, described in Figure 9.

Theorem 3. *The indexed multi-message SPS scheme* $\mathsf{IMM\text{-}SPS}$ *(Figure 9) is correct and* $\mathsf{EUF\text{-}CiMA}$ *secure (Definition 8) under the GPS_3 assumption (Definition 6) in the random oracle model.*

The proof can be found in Appendix B.3.

$\mathsf{Setup}(1^\kappa)$

$(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$
$\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g})$
return pp

$\mathsf{KGen}(\mathsf{pp}, \ell)$

Parse (pp, ℓ)
$x, y_1, \ldots, y_\ell \leftarrow\$ \mathbb{Z}_p^*$
$\vec{\mathsf{sk}} \leftarrow (\mathsf{sk}_0, \ldots, \mathsf{sk}_\ell) = (x, y_1, \ldots, y_\ell)$
$\vec{\mathsf{vk}} \leftarrow (\mathsf{vk}_0, \ldots, \mathsf{vk}_\ell) = (\hat{g}^x, \hat{g}^{y_1}, \ldots, \hat{g}^{y_\ell})$
return $(\vec{\mathsf{sk}}, \vec{\mathsf{vk}})$

$\mathsf{Sign}(\mathsf{pp}, \vec{\mathsf{sk}}, (id, \vec{M}_1, \vec{M}_2))$

Parse $(\mathsf{pp}, \vec{\mathsf{sk}})$
$h \leftarrow\$ \mathsf{H}(id)$
if $\exists\, j \in [1, \ell] \mid$
  $e(h, M_{2j}) \neq e(M_{1j}, \hat{g})$:
  return $\perp$
else : $s = h^{\mathsf{sk}_0} \prod_{j=1}^{\ell} M_{1j}^{\mathsf{sk}_j}$
  return $\sigma \leftarrow (h, s)$

$\mathsf{Verify}(\mathsf{pp}, \vec{\mathsf{vk}}, (\vec{M}_1, \vec{M}_2), \sigma)$

// does not invoke H
Parse $(\mathsf{pp}, \vec{\mathsf{vk}}, \sigma)$
return $\Big(h \neq 1_{\mathbb{G}_1} \;\wedge\; M_{1j} \neq 1_{\mathbb{G}_1} \;\wedge$
$e(h, M_{2j}) = e(M_{1j}, \hat{g})\big\}_{j \in [1,\ell]} \;\wedge$
$e(h, \mathsf{vk}_0) \prod_{j=1}^{\ell} e(M_{1j}, \mathsf{vk}_j) = e(s, \hat{g})\Big)$

Fig. 9: Our Indexed Multi-Message SPS Construction $\mathsf{IMM\text{-}SPS}$.

5 Threshold Structure-Preserving Signatures

In this section, we define the syntax and security notions of non-interactive (n, t)-Threshold Structure-Preserving Signatures (TSPS) for indexed message spaces and then propose an efficient instantiation for an indexed Diffie-Hellman multi-message space. Generally, in an (n, t)-TSPS, the signing key is distributed among n parties and the generation of any signature requires the cooperation of a subset of parties of size at least t. Our syntax assumes a centralized key generation algorithm for distributing the signing key, but can be replaced with a decentralized key generation protocol (DKG).

5.1 Definition and Security Requirements

Definition 10 (Threshold Structure-Preserving Signature). *For a given security parameter κ and bilinear group $\mathcal{BG}$, an (n, t)-TSPS over indexed message space $\mathcal{M}$ consists of a tuple* $(\mathsf{Setup}, \mathsf{KGen}, \mathsf{ParSign}, \mathsf{ParVerify}, \mathsf{Reconst}, \mathsf{Verify})$ *of PPT algorithms defined as follows:*

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$*: The setup algorithm takes the security parameter 1^κ as input and returns the public parameters* pp *as output.*
- $(\vec{\mathsf{sk}}, \vec{\mathsf{vk}}, \mathsf{vk}) \leftarrow \mathsf{KGen}(\mathsf{pp}, \ell, n, t)$*: The probabilistic key generation algorithm takes the public parameters* pp *and length ℓ along with two integers $t, n \in$*

$\mathsf{poly}(1^\kappa)$ *such that* $1 \le t \le n$ *as inputs. It returns two vectors of size* n *of signing/verification keys* $\vec{\mathsf{sk}} = (\mathsf{sk}_1, \dots, \mathsf{sk}_n)$ *and* $\vec{\mathsf{vk}} = (\mathsf{vk}_1, \dots, \mathsf{vk}_n)$ *such that each party* P_i *for* $i \in [n]$ *receives a pair* $(\mathsf{sk}_i, \mathsf{vk}_i)$ *along with the global verification key* vk*.*

- $\sigma_i \leftarrow \mathsf{ParSign}(\mathsf{pp}, \mathsf{sk}_i, M)$*: The partial signing algorithm takes the public parameters* pp*, a secret signing key* sk_i*, and a message* $M \in \mathcal{M}$ *as inputs and returns the partial signature* σ_i *as output.*
- $0/1 \leftarrow \mathsf{ParVerify}(\mathsf{pp}, \mathsf{vk}_i, \tilde{M}, \sigma_i)$*: The partial verification algorithm is a deterministic algorithm that takes a verification key* vk_i*, message* $\tilde{M} \in \tilde{\mathcal{M}}$*, and partial signature* σ_i *as inputs. If* σ_i *is a valid partial signature, it returns* 1*; else, it returns* 0*. We refer to well-formed partial signatures as those that pass this verification.*
- $\sigma \leftarrow \mathsf{Reconst}(\mathsf{pp}, \{i, \sigma_i\}_{i \in \mathcal{T}})$*: The reconstruction algorithm takes public parameters* pp *and a set of well-formed partial signatures* $\{i, \sigma_i\}$ *over subset* $\mathcal{T} \subseteq \{1, \dots, n\}$ *as inputs. It outputs an aggregated signature* σ *if* $|\mathcal{T}| \ge t$*; else, it returns* $\perp$*.*
- $0/1 \leftarrow \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \tilde{M}, \sigma)$*: This deterministic algorithm takes the verification key* vk*, a message* $\tilde{M} \in \tilde{\mathcal{M}}$*, and an aggregated signature* σ *as inputs. It outputs either* 1 *(accept) or* 0 *(reject).*

Two main security properties for our TSPS are correctness and threshold existential unforgeability against chosen indexed message attack (Threshold $\mathsf{EUF\text{-}CiMA}$), defined as follows.

Definition 11 (Correctness). *An* (n,t)*-TSPS scheme is called correct if we have:*

$$\Pr[\forall\ \mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa), (\vec{\mathsf{sk}}, \vec{\mathsf{vk}}, \mathsf{vk}) \leftarrow \mathsf{KGen}(\mathsf{pp}, \ell, n, t), M \in \mathcal{M}, |\mathcal{T}| \ge t :$$
$$\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \tilde{M}, \mathsf{Reconst}\left(\mathsf{pp}, \{\mathsf{ParSign}(\mathsf{pp}, \mathsf{sk}_i, M)\}_{i \in \mathcal{T}}\right)) = 1] \ge 1 - \nu(\kappa)\ .$$

We next define the notion of threshold unforgeability of non-interactive TSPS schemes. The Threshold $\mathsf{EUF\text{-}CiMA}$ game is defined formally in Figure 10. Given a set of player indices $\mathcal{P} = \{1, \dots, n\}$, we assume that the adversary can corrupt up to $t-1$ signers, which we denote by the set $\mathcal{C}$. We assume the set of honest signers $\mathcal{H} = \mathcal{P} \setminus \mathcal{C}$ to be of size at least one.

In the unforgeability game, the challenger generates public parameters pp and returns them to the adversary. The adversary chooses the set of corrupted participants $\mathcal{C}$. The challenger then runs KGen to derive the global verification key vk, the individual verification keys $\{\mathsf{vk}_i\}_{i \in \{n\}}$, and the secret signing shares $\{\mathsf{sk}_i\}_{i \in \{n\}}$. It returns vk, $\{\mathsf{vk}_i\}_{i \in \{n\}}$, and the set of corrupt signing shares $\{\mathsf{sk}_j\}_{j \in \mathcal{C}}$ to the adversary. We assume the adversary maintains state before and after KGen.

After key generation has concluded, the adversary can request partial signatures on messages of its choosing from honest signers by querying oracle $\mathcal{O}_{\mathsf{PSign}}(\cdot)$.

The adversary wins if it can produce a valid forgery $(\tilde{M}^*, \sigma^*)$ with respect to the global verification key vk representing the set of n signers, on a message $\tilde{M}^*$ for which no equivalent $\tilde{M}^{*\prime}$ has been previously queried to $\mathcal{O}_{\mathsf{PSign}}(\cdot)$.

$\mathbf{G}_{\mathcal{A}}^{\text{T-EUF-CiMA}}(1^\kappa)$	$\mathcal{O}_{\mathsf{PSign}}(k, id, \tilde{M})$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$	$\mathbf{if}\ k \notin \mathcal{H}\ \vee\ (k, id, \star) \in \mathcal{Q}_{\mathsf{S}}\ \vee$
$\mathcal{C} \leftarrow\$\ \mathcal{A}^{\mathsf{H}}(\mathsf{pp})$	$\quad (\star, id, \tilde{M}') \in \mathcal{Q}_{\mathsf{S}}, \tilde{M}' \neq \tilde{M} :$
$\mathbf{if}\ \mathcal{C} \notin [1,n]\ \vee\ \lvert\mathcal{C}\rvert > t-1 :$	$\quad \mathbf{return}\ \perp$
$\quad \mathbf{return}\ \perp$	$\mathbf{else}\ :$
$\mathbf{else}\ :$	$\quad \sigma_k \leftarrow \mathsf{ParSign}\left(\mathsf{pp}, \mathsf{sk}_k, (id, \tilde{M})\right)$
$\quad \mathcal{H} \leftarrow [1,n] \setminus \mathcal{C}$	$\quad \mathcal{Q}_{\mathsf{S}} \leftarrow \mathcal{Q}_{\mathsf{S}} \cup \{(k, id, \tilde{M})\}$
$\quad \left(\vec{\mathsf{sk}}, \vec{\mathsf{vk}}, \mathsf{vk}\right) \leftarrow \mathsf{KGen}\left(\mathsf{pp}, \ell, n, t\right)$	$\quad \mathcal{Q}_{\mathsf{EQ}} \leftarrow \mathcal{Q}_{\mathsf{EQ}} \cup \{\mathsf{EQ}(\tilde{M})\}$
$\quad \left(\tilde{M}^*, \sigma^*\right) \leftarrow\$\ \mathcal{A}^{\mathsf{H}, \mathcal{O}_{\mathsf{PSign}}}(\{\vec{\mathsf{sk}}\}_{i \in \mathcal{C}}, \vec{\mathsf{vk}}, \mathsf{vk})$	$\quad \mathbf{return}\ \sigma_k$
$\quad \mathbf{return}\ \Big(\tilde{M}^* \notin \mathcal{Q}_{\mathsf{EQ}}\ \wedge$	
$\qquad\qquad \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \tilde{M}^*, \sigma^*)\Big)$	

Fig. 10: Game $\mathbf{G}_{\mathcal{A}}^{\text{T-EUF-CiMA}}(1^\kappa)$.

Definition 12 (Threshold EUF-CiMA). *A non-interactive (n,t)-TSPS scheme over indexed message space $\mathcal{M}$ is Threshold EUF-CiMA secure if for all PPT adversaries $\mathcal{A}$ playing game $\mathbf{G}^{\textsf{T-EUF-CiMA}}$ (Figure 10), there exists a negligible function ν such that*

$$Adv_{\mathcal{A}}^{\textsf{T-EUF-CiMA}}(\kappa) = \Pr\left[\mathbf{G}_{\mathcal{A}}^{\textsf{T-EUF-CiMA}}(1^\kappa) = 1\right] \leq \nu(\kappa) \ .$$

5.2 Our Indexed Multi-Message TSPS

In Figure 11, we present the proposed (n,t)-TSPS scheme TSPS over indexed Diffie-Hellman multi-message space $\mathcal{M}^{\mathsf{H}}_{\mathsf{MiDH}}$, as defined in Figure 8.

5.3 Security of TSPS

Theorem 4. *The indexed multi-message (n,t)-threshold SPS scheme* TSPS *is correct and Threshold EUF-CiMA secure (Definition 12) in the random oracle model under the EUF-CiMA security of* $\mathsf{IMM\text{-}SPS}$ *(Theorem 3).*

Proof. **Correctness.** First, we show that the reconstruction algorithm for a set of valid partial signatures $\{i, \sigma_i\}_{i \in \mathcal{T}}, |\mathcal{T}| \geq t$, on a message $M = (id, \vec{M}_1, \vec{M}_2)$ results in a valid aggregated signature $\sigma = (h, s)$. Indeed,

$$s = \prod_{i \in \mathcal{T}} s_i^{\lambda_i} = \prod_{i \in \mathcal{T}} \left(h^{\mathsf{sk}_{i0}} \prod_{j=1}^{\ell} M_{1j}^{\mathsf{sk}_{ij}}\right)^{\lambda_i} = h^{\sum_{i \in \mathcal{T}} \mathsf{sk}_{i0} \lambda_i} \prod_{j=1}^{\ell} M_{1j}^{\sum_{i \in \mathcal{T}} \mathsf{sk}_{ij} \lambda_i} = h^{\mathsf{sk}_0} \prod_{j=1}^{\ell} M_{1j}^{\mathsf{sk}_j} \ ,$$

where λ_i is the Lagrange coefficient for party i with respect to the signing set $\mathcal{T}$. Next, we show that verification holds for the above aggregated signature

$\mathsf{Setup}(1^\kappa)$

$(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$
$\mathsf{pp} \leftarrow (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g})$
return pp

$\mathsf{KGen}(\mathsf{pp}, \ell, n, t)$

Parse $(\mathsf{pp}, \ell, n, t)$
$x, y_1, ..., y_\ell \leftarrow\$\, \mathbb{Z}_p^*$
$\vec{x} \leftarrow\$\, \mathsf{Share}(x, p, n, t)$
for $j \in [1, \ell]$:
  $\vec{y}_j \leftarrow\$\, \mathsf{Share}(y_j, p, n, t)$
for $i \in [1, n]$:
  $\mathsf{vk}_i \leftarrow (\mathsf{vk}_{i0}, ..., \mathsf{vk}_{i\ell}) = (\hat{g}^{x_i}, \hat{g}^{y_{i1}}, ..., \hat{g}^{y_{i\ell}})$
  $\mathsf{sk}_i \leftarrow (\mathsf{sk}_{i0}, ..., \mathsf{sk}_{i\ell}) = (x_i, y_{i1}, ..., y_{i\ell})$
$\vec{\mathsf{sk}} \leftarrow (\mathsf{sk}_1, ..., \mathsf{sk}_n)$
$\vec{\mathsf{vk}} \leftarrow (\mathsf{vk}_1, ..., \mathsf{vk}_n)$
$\mathsf{vk} \leftarrow (\mathsf{vk}_{00}, ..., \mathsf{vk}_{0\ell}) = (\hat{g}^{x}, \hat{g}^{y_1}, ..., \hat{g}^{y_\ell})$
return $(\vec{\mathsf{sk}}, \vec{\mathsf{vk}}, \mathsf{vk})$

$\mathsf{ParSign}(\mathsf{pp}, \mathsf{sk}_i, (id, \vec{M}_1, \vec{M}_2))$

Parse $(\mathsf{pp}, \mathsf{sk}_i)$
$h \leftarrow\$\, \mathsf{H}(id)$
if $\exists\, j \in [1, \ell] : e(h, M_{2j}) \neq e(M_{1j}, \hat{g})$
  return $\perp$
else : $s_i = h^{\mathsf{sk}_{i0}} \prod_{j=1}^{\ell} {M_{1j}}^{\mathsf{sk}_{ij}}$
  return $\sigma_i \leftarrow (h, s_i)$

$\mathsf{ParVerify}(\mathsf{pp}, \mathsf{vk}_i, (\vec{M}_1, \vec{M}_2), \sigma_i)$

// does not invoke H
Parse $(\mathsf{pp}, \mathsf{vk}_i, \sigma_i)$
return $\big(h \neq 1_{\mathbb{G}_1} \;\wedge\; \big\{M_{1j} \neq 1_{\mathbb{G}_1} \;\wedge$
$e(h, M_{2j}) = e(M_{1j}, \hat{g})\big\}_{j \in [1,\ell]} \;\wedge$
$e(h, \mathsf{vk}_{i0}) \prod_{j=1}^{\ell} e(M_{1j}, \mathsf{vk}_{ij}) = e(s_i, \hat{g})\big)$

$\mathsf{Reconst}(\mathsf{pp}, \{i, \sigma_i\}_{i \in \mathcal{T}})$

Parse pp and $\sigma_i = (h, s_i)$
if $|\mathcal{T}| < t$:
  return $\perp$
else :
  return $\sigma \leftarrow (h, s) = (h, \prod_{i \in \mathcal{T}} s_i^{\lambda_i})$
  // λ_i are the Lagrange coefficients

$\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, (\vec{M}_1, \vec{M}_2), \sigma)$

// does not invoke H
Parse $(\mathsf{pp}, \mathsf{vk}, \sigma)$
return $\big(h \neq 1_{\mathbb{G}_1} \;\wedge\; \big\{M_{1j} \neq 1_{\mathbb{G}_1} \;\wedge$
$e(h, M_{2j}) = e(M_{1j}, \hat{g})\big\}_{j \in [1,\ell]} \;\wedge$
$e(h, \mathsf{vk}_{00}) \prod_{j=1}^{\ell} e(M_{1j}, \mathsf{vk}_{0j}) = e(s, \hat{g})\big)$

Fig. 11: Our Threshold SPS Construction TSPS.

σ on message $\tilde{M} = (\vec{M}_1, \vec{M}_2)$. Indeed, $\forall\ j \in [1, \ell] : e(h, M_{2j}) = e(h, \hat{g}^{m_j}) = e(h^{m_j}, \hat{g}) = e(M_{1j}, \hat{g})$ and

$$e(h, \mathsf{vk}_0) \prod_{j=1}^{\ell} e(M_{1j}, \mathsf{vk}_j) = e(h, \hat{g}^x) \prod_{j=1}^{\ell} e(M_{1j}, \hat{g}^{y_j}) = e\left(h^x \prod_{j=1}^{\ell} M_{1j}^{y_j}, \hat{g}\right) = e(s, \hat{g}) \ .$$

Need for Uniqueness. Note that the hypothetical attack described after Theorem 2 also works with partial signing oracles. Assume an (n, t)-TSPS with $n > 2t$, and suppose there were no uniqueness requirement for the message space and that the redundant check in Line 2 of the $\mathcal{O}_{\mathsf{PSign}}(\cdot)$ oracle of Figure 10 were not present. Then, a forger could obtain $2t$ partial signatures to reconstruct signatures $s = h^x M_1^y$, $s' = h^x {M'_1}^y$ and compute a forgery $s^* = s^2/s' = h^x(M_1^2/M'_1)^y$ that is a valid signature on fresh message M_1^2/M'_1 .

Threshold EUF-CiMA. Our proof is similar to that of threshold BLS in [Bol03].

We wish to show that if there exists a PPT adversary $\mathcal{A}$ that breaks the Threshold EUF-CiMA security (Figure 10) of TSPS (Figure 11) with non-negligible probability, then we can construct a PPT adversary $\mathcal{B}$ that breaks the EUF-CiMA security (Figure 5) of the underlying IMM-SPS scheme (Figure 6) with non-negligible probability.

Suppose there exists such a PPT adversary $\mathcal{A}$. Then, running $\mathcal{A}$ as a subroutine, we construct a reduction $\mathcal{B}$ breaking the EUF-CiMA security of IMM-SPS as follows.

The reduction $\mathcal{B}$ is responsible for simulating oracle responses for queries to $\mathcal{O}_{\mathsf{PSign}}(\cdot)$ and H. Let $\mathcal{Q}_{\mathsf{H}}$ be the set of H queries id and their responses. $\mathcal{B}$ may program the random oracle H. Let $\mathcal{Q}_{\mathsf{S}}$ be the set of $\mathcal{O}_{\mathsf{PSign}}(\cdot)$ queries $(k, id, \tilde{M})$ and $\mathcal{Q}_{\mathsf{EQ}}$ the set of equivalence classes of messages $\tilde{M}$. $\mathcal{B}$ initializes $\mathcal{Q}_{\mathsf{H}}, \mathcal{Q}_{\mathsf{S}}, \mathcal{Q}_{\mathsf{EQ}}$ to the empty set.

Initialization. $\mathcal{B}$ takes as input public parameters $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g})$ and an IMM-SPS verification key vk'. In the EUF-CiMA game, $\mathcal{B}$ has access to oracles $\mathcal{O}'_{\mathsf{Sign}}(\cdot)$ and H'. $\mathcal{B}$ uses $\mathsf{vk}' = (\mathsf{vk}'_{00}, \mathsf{vk}'_{01}, \ldots, \mathsf{vk}'_{0\ell})$ as the TSPS verification key $\mathsf{vk} = (\mathsf{vk}_{00}, \mathsf{vk}_{01}, \ldots, \mathsf{vk}_{0\ell})$.

Simulating Key Generation. $\mathcal{B}$ simulates the key generation algorithm as follows.

- $\mathcal{B}$ defines the pair of secret/verification keys of the corrupted parties $P_i, i \in \mathcal{C}$, as follows. Assume without loss of generality that $|\mathcal{C}| = t - 1$. For all $i \in \mathcal{C}$, $\mathcal{B}$ samples random values $x_{i0}, y_{i1}, \ldots, y_{i\ell} \leftarrow\!\!\$\, \mathbb{Z}_p^{\ell+1}$ and defines party P_i's secret key as $\mathsf{sk}_i \leftarrow (\mathsf{sk}_{i0}, \mathsf{sk}_{i1}, \ldots, \mathsf{sk}_{i\ell}) = (x_{i0}, y_{i1}, \ldots, y_{i\ell})$ the corresponding verification key as $\mathsf{vk}_i \leftarrow (\mathsf{vk}_{i0}, \mathsf{vk}_{i1}, \ldots, \mathsf{vk}_{i\ell}) = (\hat{g}^{x_{i0}}, \hat{g}^{y_{i1}}, \ldots, \hat{g}^{y_{i\ell}})$.
- To generate the verification key of the honest parties $P_k, k \in \mathcal{H}, \mathcal{H} = [1, n] \setminus \mathcal{C}$, $\mathcal{B}$ proceeds as follows:

1. For all $i \in \tilde{\mathcal{T}} = \mathcal{C} \cup \{0\}$, it computes the Lagrange polynomials evaluated at point k:
$$\tilde{\lambda}_{ki} = L_i^{\tilde{\mathcal{T}}}(k) = \prod_{j \in \tilde{\mathcal{T}} j \neq i} \frac{(j-k)}{(j-i)} \ . \tag{2}$$
2. It takes the verification keys of corrupted parties $\{\mathsf{vk}_i\}_{i \in \mathcal{C}}$ and the global verification key vk and computes
$$\begin{aligned}\mathsf{vk}_k &= (\mathsf{vk}_{k0}, \mathsf{vk}_{k1}, \ldots, \mathsf{vk}_{k\ell}) \\ &= \left(\mathsf{vk}_{00}^{\tilde{\lambda}_{k0}} \prod_{i \in \mathcal{C}} \mathsf{vk}_{i0}^{\tilde{\lambda}_{ki}}, \mathsf{vk}_{01}^{\tilde{\lambda}_{k0}} \prod_{i \in \mathcal{C}} \mathsf{vk}_{i1}^{\tilde{\lambda}_{ki}}, \ldots, \mathsf{vk}_{0\ell}^{\tilde{\lambda}_{k0}} \prod_{i \in \mathcal{C}} \mathsf{vk}_{i\ell}^{\tilde{\lambda}_{ki}}\right) \ .\end{aligned}$$

$\mathcal{B}$ returns the global verification key vk, $\vec{\mathsf{vk}} = (\mathsf{vk}_1, \ldots, \mathsf{vk}_n)$, and secret keys $\left\{\mathsf{sk}_j\right\}_{j \in \mathcal{C}}$ to $\mathcal{A}$.

Simulating Random Oracle $\mathsf{H}(id)$**:** When $\mathcal{A}$ queries H on index id, if $\mathcal{Q}_{\mathsf{H}}[id] = \perp$, then $\mathcal{B}$ queries $\mathsf{H}'(id)$, receives a base element h, and sets $\mathcal{Q}_{\mathsf{H}}[id] \leftarrow h$. $\mathcal{B}$ returns $\mathcal{Q}_{\mathsf{H}}[id]$ to $\mathcal{A}$.

Simulating Signing Oracle $\mathcal{O}_{\mathsf{PSign}}(k, id, \tilde{M})$**:** When $\mathcal{A}$ queries $\mathcal{O}_{\mathsf{PSign}}(\cdot)$ on $(k, id, \tilde{M})$ for honest party identifier $k \in \mathcal{H}$ and message $\tilde{M} = (\vec{M}_1, \vec{M}_2)$, if $k \notin \mathcal{H}$ or $(k, id, \star) \in \mathcal{Q}_{\mathsf{S}}$ or $(\star, id, \tilde{M}') \in \mathcal{Q}_{\mathsf{S}}, \tilde{M}' \neq \tilde{M}$, $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ does the following:

1. $\mathcal{B}$ looks up $h = \mathcal{Q}_{\mathsf{H}}[id]$, queries $\mathcal{O}'_{\mathsf{Sign}}(id, \vec{M}_1, \vec{M}_2)$, and receives the signature $\sigma_0 = (h, s_0)$ as output.
2. For all $i \in \mathcal{C}$, $\mathcal{B}$ computes the partial signatures $\sigma_i = (h, s_i) = \left(h, h^{\mathsf{sk}_{i0}} \prod_{j=1}^{\ell} M_{1j}^{\mathsf{sk}_{ij}}\right)$, as it knows the secret keys of corrupted parties.
3. For all $i \in \tilde{\mathcal{T}} = \mathcal{C} \cup \{0\}$, $\mathcal{B}$ computes Lagrange coefficients $\tilde{\lambda}_{ki}$ as in Equation (2).
4. $\mathcal{B}$ updates $\mathcal{Q}_{\mathsf{S}} \leftarrow \mathcal{Q}_{\mathsf{S}} \cup \{(k, id, \tilde{M})\}$ and $\mathcal{Q}_{\mathsf{EQ}} \leftarrow \mathcal{Q}_{\mathsf{EQ}} \cup \{\mathsf{EQ}(\tilde{M})\}$.
5. $\mathcal{B}$ computes $(h, s_k) = \left(h, s_0^{\tilde{\lambda}_{k0}} \prod_{i \in \mathcal{C}} s_i^{\tilde{\lambda}_{ki}}\right)$ and returns $\sigma_k = (h, s_k)$ to $\mathcal{A}$.

Output. At the end of the game, $\mathcal{A}$ produces a valid forgery $\sigma^* = (h^*, s^*)$ on message $\tilde{M}^* = (\vec{M}_1^*, \vec{M}_2^*)$, and $\mathcal{B}$ returns $(\tilde{M}^*, \sigma^*)$ as its forgery.

$\mathcal{B}$ correctly simulates key generation and $\mathcal{A}$'s hash and signing queries. Since $\mathcal{A}$'s forgery satisfies $\tilde{M}^* \notin \mathcal{Q}_{\mathsf{EQ}}$ and $\mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, \tilde{M}^*, \sigma^*) = 1$, $\mathcal{B}$'s winning conditions are also satisfied and
$$Adv_{\mathsf{TSPS},\mathcal{A}}^{\mathsf{T\text{-}EUF\text{-}CiMA}}(\kappa) \leq Adv_{\mathsf{IMM\text{-}SPS},\mathcal{B}}^{\mathsf{EUF\text{-}CiMA}}(\kappa) \ .$$

□

6 Conclusion and Open Problems

In this work, we introduce the notion of a threshold structure-preserving signature (TSPS) and present an efficient fully non-interactive TSPS construction. We prove that the proposed TSPS is secure under a new variant of the generalized Pointcheval-Sanders (PS) assumption in the random oracle model.

While we use a message indexing method in order to construct a non-interactive scheme, a TSPS without indexing is an interesting open problem. Moreover, it is interesting to construct schemes that rely on weaker assumptions and avoid the use of the random oracle model. When it comes to the security model, the following two challenging problems remain open: obtaining security under adaptive corruptions more tightly than via a guessing argument from static corruptions, and achieving the strongest notion possible for fully non-interactive schemes (TS-UF-1) [BCK+22].

In general we believe this work can open a new line of research for structure-preserving multi-party protocols, such as threshold structure-preserving encryption.

Acknowledgments. We would like to thank Behzad Abdolmaleki, Daniele Cozzo and Hyoseung Kim for their valuable comments. Elizabeth Crites was supported by Input Output through their funding of the Blockchain Technology Lab at the University of Edinburgh. The work of Markulf Kohlweiss was done in part while visiting imec-COSIC. Mahdi Sedaghat and Bart Preneel were supported in part by the Research Council KU Leuven C1 on Security and Privacy for Cyber-Physical Systems and the Internet of Things with contract number C16/15/058 and by CyberSecurity Research Flanders with reference number VR20192203. Daniel Slamanig was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 871473 (Kraken) and No. 861696 (Labyrinth) and by the Austrian Science Fund (FWF) and netidee SCIENCE under grant agreement P31621-N38 (Profet).

References

AAOT18. Masayuki Abe, Miguel Ambrona, Miyako Ohkubo, and Mehdi Tibouchi. Lower bounds on structure-preserving signatures for bilateral messages. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 3–22. Springer, Heidelberg, September 2018.

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.

ACHO11. Masayuki Abe, Sherman S. M. Chow, Kristiyan Haralambiev, and Miyako Ohkubo. Double-trapdoor anonymous tags for traceable signatures. In Javier Lopez and Gene Tsudik, editors, *ACNS 11*, volume 6715 of *LNCS*, pages 183–200. Springer, Heidelberg, June 2011.

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.

AJO^{+}19. Masayuki Abe, Charanjit S. Jutla, Miyako Ohkubo, Jiaxin Pan, Arnab Roy, and Yuyu Wang. Shorter QA-NIZK and SPS with tighter security. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 669–699. Springer, Heidelberg, December 2019.

ALP12. Nuttapong Attrapadung, Benoît Libert, and Thomas Peters. Computing on authenticated data: New privacy definitions and constructions. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 367–385. Springer, Heidelberg, December 2012.

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.

BCF^{+}11. Olivier Blazy, Sébastien Canard, Georg Fuchsbauer, Aline Gouget, Hervé Sibert, and Jacques Traoré. Achieving optimal anonymity in transferable e-cash with a judge. In Abderrahmane Nitaj and David Pointcheval, editors, *AFRICACRYPT 11*, volume 6737 of *LNCS*, pages 206–223. Springer, Heidelberg, July 2011.

BCK^{+}22. Mihir Bellare, Elizabeth C. Crites, Chelsea Komlo, Mary Maller, Stefano Tessaro, and Chenzhi Zhu. Better than advertised security for non-interactive threshold signatures. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part IV*, volume 13510 of *Lecture Notes in Computer Science*, pages 517–550. Springer, 2022.

BDV^{+}20. Luís TAN Brandão, Michael Davidson, Apostol Vassilev, et al. Nist roadmap toward criteria for threshold schemes for cryptographic primitives. In *National Institute of Standards and Technology Internal or Interagency Report 8214A*, 2020.

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.

BFL20. Balthazar Bauer, Georg Fuchsbauer, and Julian Loss. A classification of computational assumptions in the algebraic group model. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 121–151. Springer, Heidelberg, August 2020.

BL22. Renas Bacho and Julian Loss. On the Adaptive Security of the Threshold BLS Signature Scheme. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, September 2022.

Bla79. G. R. Blakley. Safeguarding cryptographic keys. *Proceedings of AFIPS 1979 National Computer Conference*, 48:313–317, 1979.

BLS04. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, September 2004.

Bol03. Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer, Heidelberg, January 2003.

CDHK15. Jan Camenisch, Maria Dubovitskaya, Kristiyan Haralambiev, and Markulf Kohlweiss. Composable and modular anonymous credentials: Definitions and practical constructions. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 262–288. Springer, Heidelberg, November / December 2015.

CDL$^+$20. Jan Camenisch, Manu Drijvers, Anja Lehmann, Gregory Neven, and Patrick Towa. Short threshold dynamic group signatures. In Clemente Galdi and Vladimir Kolesnikov, editors, *SCN 20*, volume 12238 of *LNCS*, pages 401–423. Springer, Heidelberg, September 2020.

CDN01. Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. Multiparty computation from threshold homomorphic encryption. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 280–299. Springer, Heidelberg, May 2001.

CFSY96. Ronald Cramer, Matthew K. Franklin, Berry Schoenmakers, and Moti Yung. Multi-autority secret-ballot elections with linear work. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 72–83. Springer, Heidelberg, May 1996.

CGG$^+$20. Ran Canetti, Rosario Gennaro, Steven Goldfeder, Nikolaos Makriyannis, and Udi Peled. UC non-interactive, proactive, threshold ECDSA with identifiable aborts. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1769–1787. ACM Press, November 2020.

CGJ$^+$99. Ran Canetti, Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Adaptive security for threshold cryptosystems. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 98–115. Springer, Heidelberg, August 1999.

CGS97. Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 103–118. Springer, Heidelberg, May 1997.

CH20. Geoffroy Couteau and Dominik Hartmann. Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 768–798. Springer, Heidelberg, August 2020.

CL19. Elizabeth C. Crites and Anna Lysyanskaya. Delegatable anonymous credentials from mercurial signatures. In Mitsuru Matsui, editor, *CT-RSA 2019*, volume 11405 of *LNCS*, pages 535–555. Springer, Heidelberg, March 2019.

DDFY94. Alfredo De Santis, Yvo Desmedt, Yair Frankel, and Moti Yung. How to share a function securely. In *26th ACM STOC*, pages 522–533. ACM Press, May 1994.

DEF+19. Manu Drijvers, Kasra Edalatnejad, Bryan Ford, Eike Kiltz, Julian Loss, Gregory Neven, and Igors Stepanovs. On the security of two-round multi-signatures. In *2019 IEEE Symposium on Security and Privacy*, pages 1084–1101. IEEE Computer Society Press, May 2019.

Des90. Yvo Desmedt. Making conditionally secure cryptosystems unconditionally abuse-free in a general context. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 6–16. Springer, Heidelberg, August 1990.

DF90. Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 307–315. Springer, Heidelberg, August 1990.

DK01. Ivan Damgård and Maciej Koprowski. Practical threshold RSA signatures without a trusted dealer. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 152–165. Springer, Heidelberg, May 2001.

DKLs19. Jack Doerner, Yashvanth Kondi, Eysa Lee, and abhi shelat. Threshold ECDSA from ECDSA assumptions: The multiparty case. In *2019 IEEE Symposium on Security and Privacy*, pages 1051–1066. IEEE Computer Society Press, May 2019.

DN03. Ivan Damgård and Jesper Buus Nielsen. Universally composable efficient multiparty computation from threshold homomorphic encryption. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 247–264. Springer, Heidelberg, August 2003.

EGK14. Ali El Kaafarani, Essam Ghadafi, and Dalia Khader. Decentralized traceable attribute-based signatures. In Josh Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 327–348. Springer, Heidelberg, February 2014.

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.

FHS19. Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. *Journal of Cryptology*, 32(2):498–546, April 2019.

FKL18. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018.

Fuc09. Georg Fuchsbauer. Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. Cryptology ePrint Archive, Report 2009/320, 2009. https://eprint.iacr.org/2009/320.

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.

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.

Gha17. Essam Ghadafi. More efficient structure-preserving signatures - or: Bypassing the type-III lower bounds. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *ESORICS 2017, Part II*, volume 10493 of *LNCS*, pages 43–61. Springer, Heidelberg, September 2017.

GHKP18. Romain Gay, Dennis Hofheinz, Lisa Kohl, and Jiaxin Pan. More efficient (almost) tightly secure structure-preserving signatures. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 230–258. Springer, Heidelberg, April / May 2018.

GJM+21. Kobi Gurkan, Philipp Jovanovic, Mary Maller, Sarah Meiklejohn, Gilad Stern, and Alin Tomescu. Aggregatable distributed key generation. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 147–176. Springer, Heidelberg, October 2021.

GMR88. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.

GPS08. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008. Applications of Algebra to Cryptography.

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.

HJ12. Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607. Springer, Heidelberg, August 2012.

JR17. Charanjit S. Jutla and Arnab Roy. Improved structure preserving signatures under standard bilinear assumptions. In Serge Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 183–209. Springer, Heidelberg, March 2017.

KLAP20. Hyoseung Kim, Youngkyung Lee, Michel Abdalla, and Jong Hwan Park. Practical dynamic group signature with efficient concurrent joins and batch verifications. Cryptology ePrint Archive, Report 2020/921, 2020. https://eprint.iacr.org/2020/921.

KMOS21. Yashvanth Kondi, Bernardo Magri, Claudio Orlandi, and Omer Shlomovits. Refresh When You Wake Up: Proactive Threshold Wallets with Offline Devices. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 608–625, 2021.

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.

KSAP22. Hyoseung Kim, Olivier Sanders, Michel Abdalla, and Jong Hwan Park. Practical Dynamic Group Signatures Without Knowledge Extractors. *Designs, Codes and Cryptography*, Oct 2022.

Lin17. Yehuda Lindell. Fast secure two-party ECDSA signing. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 613–644. Springer, Heidelberg, August 2017.

LJY14. Benoît Libert, Marc Joye, and Moti Yung. Born and raised distributively: fully distributed non-interactive adaptively-secure threshold signatures with short shares. In Magnús M. Halldórsson and Shlomi Dolev, editors, *33rd ACM PODC*, pages 303–312. ACM, July 2014.

LJY16. Benoît Libert, Marc Joye, and Moti Yung. Born and Raised Distributively: Fully Distributed Non-Interactive Adaptively-Secure Threshold Signatures with short shares. *Theor. Comput. Sci.*, 645:1–24, 2016.

LPJY13. Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Linearly homomorphic structure-preserving signatures and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 289–307. Springer, Heidelberg, August 2013.

LPY15. Benoît Libert, Thomas Peters, and Moti Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 296–316. Springer, Heidelberg, August 2015.

MRV99. Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *40th FOCS*, pages 120–130. IEEE Computer Society Press, October 1999.

MTT19. Taiga Mizuide, Atsushi Takayasu, and Tsuyoshi Takagi. Tight reductions for Diffie-Hellman variants in the algebraic group model. In Mitsuru Matsui, editor, *CT-RSA 2019*, volume 11405 of *LNCS*, pages 169–188. Springer, Heidelberg, March 2019.

PS16. David Pointcheval and Olivier Sanders. Short randomizable signatures. In Kazue Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 111–126. Springer, Heidelberg, February / March 2016.

SAB+19. Alberto Sonnino, Mustafa Al-Bassam, Shehar Bano, Sarah Meiklejohn, and George Danezis. Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers. In *NDSS 2019*. The Internet Society, February 2019.

SG98. Victor Shoup and Rosario Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 1–16. Springer, Heidelberg, May / June 1998.

Sha79. Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.

Sho00. Victor Shoup. Practical threshold signatures. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 207–220. Springer, Heidelberg, May 2000.

TBA+22. Alin Tomescu, Adithya Bhat, Benny Applebaum, Ittai Abraham, Guy Gueta, Benny Pinkas, and Avishay Yanai. UTT: Decentralized Ecash with Accountable Privacy. *IACR Cryptol. ePrint Arch.*, page 452, 2022.

TS21. Dmytro Tymokhanov and Omer Shlomovits. Alpha-Rays: Key Extraction Attacks on Threshold ECDSA Implementations. Cryptology ePrint Archive, Report 2021/1621, 2021. https://ia.cr/2021/1621.

A Additional Definitions

A.1 Digital Signatures

Definition 13 (Digital Signature). *A digital signature scheme over message space $\mathcal{M}$ is a tuple of the following polynomial-time algorithms:*

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$*: Setup is a probabilistic algorithm which takes as input the security parameter 1^κ and outputs the set of public parameters* pp*.*
- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}(\mathsf{pp})$*: Key generation is a probabilistic algorithm which takes as input* pp *and outputs a pair of signing/verification keys* $(\mathsf{sk}, \mathsf{vk})$*.*
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, m)$*: The signing algorithm takes as input* pp*, a secret signing key* sk*, and a message $m \in \mathcal{M}$, and outputs a signature σ.*
- $0/1 \leftarrow \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, m, \sigma)$*: Verification is a deterministic algorithm which takes as input* pp*, a public verification key* vk*, a message $m \in \mathcal{M}$, and a purported signature σ, and outputs either 0 (reject) or 1 (accept).*

The primary security requirements for a digital signature scheme are *correctness* and *existential unforgeability against chosen message attack* (EUF-CMA).

Definition 14 (Correctness). *A digital signature is correct if we have:*

$$\Pr\begin{bmatrix}\forall\ \mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa), (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}(\mathsf{pp}), m \in \mathcal{M} : \\ \mathsf{Verify}\left(\mathsf{pp}, \mathsf{vk}, m, \mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, m)\right) = 1\end{bmatrix} \geq 1 - \nu(\kappa)\ .$$

Definition 15 (Existential Unforgeability under Chosen Message Attack (EUF-CMA) [GMR88]). *A digital signature scheme over message space $\mathcal{M}$ is EUF-CMA secure if for all PPT adversaries $\mathcal{A}$ playing game $\mathbf{G}^{\textsf{EUF-CMA}}$ (Figure 12), there exists a negligible function ν such that*

$$Adv_{\mathcal{A}}^{\textsf{EUF-CMA}}(\kappa) = \Pr\left[\mathbf{G}_{\mathcal{A}}^{\textsf{EUF-CMA}}(1^\kappa) = 1\right] \leq \nu(\kappa)\ .$$

$\mathbf{G}_{\mathcal{A}}^{\text{EUF-CMA}}(1^\kappa)$	$\mathcal{O}_{\mathsf{Sign}}(m)$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa)$	$\sigma \leftarrow \mathsf{Sign}\,(\mathsf{pp}, \mathsf{sk}, m)$
$(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}(\mathsf{pp})$	$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
$(m^*, \sigma^*) \leftarrow\!\!\$\ \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}}}(\mathsf{pp}, \mathsf{vk})$	$\mathbf{return}\ \sigma$
$\mathbf{return}\ (m^* \notin \mathcal{Q}\ \wedge\ \mathsf{Verify}(\mathsf{pp}, \mathsf{vk}, m^*, \sigma^*))$	

Fig. 12: The EUF-CMA security game.

A.2 Generalized PS Assumptions

Definition 16 (Generalized PS Assumption [KLAP20]). *Let the advantage of an adversary $\mathcal{A}$ against the GPS game $\mathbf{G}^{GPS}$, as defined in Figure 13, be as follows:*

$$Adv_{\mathcal{A}}^{GPS}(\kappa) = \Pr\left[\mathbf{G}_{\mathcal{A}}^{GPS} = 1\right] \ .$$

The GPS assumption holds if for all PPT adversaries $\mathcal{A}$, there exists a negligible function ν such that $Adv_{\mathcal{A}}^{GPS}(\kappa) < \nu(\kappa)$.

$\mathbf{G}^{\mathrm{GPS}}(1^\kappa)$

$\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$
$x, y \leftarrow\!\!\$\ \mathbb{Z}_p^*$
$(m^*, h^*, s^*) \leftarrow \mathcal{A}^{\mathcal{O}_0^{\mathrm{GPS}}, \mathcal{O}_1^{\mathrm{GPS}}}(\mathsf{pp}, \hat{g}^x, \hat{g}^y)$
return (1) $h^* \neq 1_{\mathbb{G}_1} \wedge\ m^* \neq 0 \ \wedge$
(2) $s^* = {h^*}^{x + m^* y} \ \wedge$
(3) $(\star, m^*) \notin \mathcal{Q}_1$.

$\mathcal{O}_0^{\mathrm{GPS}}()$

$h \leftarrow\!\!\$\ \mathbb{G}_1$
$\mathcal{Q}_0 \leftarrow \mathcal{Q}_0 \cup \{h\}$
return h

$\mathcal{O}_1^{\mathrm{GPS}}(h, m) //\ m \in \mathbb{Z}_p$

if $(h \notin \mathcal{Q}_0 \ \vee\ (h, \star) \notin \mathcal{Q}_1)$:
  return $\perp$
$s \leftarrow h^x M_1^y$
$\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{(h, m)\}$
return (h^{x+my})

Fig. 13: Game defining the GPS assumption.

Definition 17 (GPS_2 **Assumption [KSAP22]).** *Let the advantage of an adversary $\mathcal{A}$ against the GPS_2 game $\mathbf{G}^{GPS_2}$, as defined in Figure 14, be as follows:*

$$Adv_{\mathcal{A}}^{GPS_2}(\kappa) = \Pr\left[\mathbf{G}_{\mathcal{A}}^{GPS_2} = 1\right] \ .$$

The GPS_2 assumption holds if for all PPT adversaries $\mathcal{A}$, there exists a negligible function ν such that $Adv_{\mathcal{A}}^{GPS_2}(\kappa) < \nu(\kappa)$.

A.3 $\mathbf{GPS_3}$ assumption in the AGM

Recall that an adversary is said to be *algebraic* if for every group element $h \in \mathbb{G} = \langle g \rangle$ that it outputs, it is required to output a representation

$\mathbf{G}^{\mathrm{GPS}_2}(1^\kappa)$

$\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$

$x, y \leftarrow\!\$\, \mathbb{Z}_p$

$(M_1^*, M_2^*, h^*, s^*) \leftarrow \mathcal{A}^{\mathcal{O}_0^{\mathrm{GPS}_2}, \mathcal{O}_1^{\mathrm{GPS}_2}}(\mathsf{pp}, \hat{g}^x, \hat{g}^y)$

return (1) $h^* \neq 1_{\mathbb{G}_1} \wedge\ M^* \neq 1_{\mathbb{G}_1} \wedge$

(2) $s^* = h^{*^x} (M_1^*)^{*^y} \wedge$

(3) $\mathsf{dlog}_{h^*}(M_1^*) = \mathsf{dlog}_g(M_2^*) \wedge$

(4) $(\star, M^*) \notin \mathcal{Q}_1$.

$\mathcal{O}_0^{\mathrm{GPS}_2}()$

$r \leftarrow\!\$\, \mathbb{Z}_p^*$

$\mathcal{Q}_0 \leftarrow \mathcal{Q}_0 \cup \{g^r\}$

return g^r

$\mathcal{O}_1^{\mathrm{GPS}_2}(h, M_1, M_2)$ $//M_1, M_2 \in \mathbb{G}_1$

if $(h \notin \mathcal{Q}_0\ \vee\ \mathsf{dlog}_h(M_1) \neq \mathsf{dlog}_g(M_2))$:

  return $\perp$

if $(h, \star) \in \mathcal{Q}_1$:

  return $\perp$

$\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{(h, M_1)\}$

return $(h^x M_1^y)$

Fig. 14: Game defining the GPS_2 assumption.

$\vec{h} = (\eta_0, \eta_1, \eta_2, \dots)$ such that $h = g^{\eta_0} \prod g_i{}^{\eta_i}$, where $g, g_1, g_2, \dots \in \mathbb{G}$ are group elements that the adversary has seen thus far. The original definition of the algebraic group model (AGM) [FKL18] only captures regular cyclic groups $\mathbb{G} = \langle g \rangle$. Mizuide et al. [MTT19] extend this definition to include symmetric pairing groups ($\mathbb{G}_1 = \mathbb{G}_2$), such that the adversary is also allowed to output target group elements (in $\mathbb{G}_T$) and their representations. Recently, Couteau and Hartmann [CH20] defined the Algebraic Asymmetric Bilinear Group Model, which extends the AGM definition for asymmetric pairings by allowing the adversary to output multiple elements from all three groups.

Definition 18 (Algebraic Asymmetric Bilinear Group Model [CH20]). *For a given asymmetric bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g})$, an adversary $\mathcal{A}_{alg}$ who takes the vectors $\vec{\zeta}_1 = (g_1, \dots, g_n) \in \mathbb{G}_1^n$, $\vec{\zeta}_2 = (g'_1, \dots, g'_{n'}) \in \mathbb{G}_2^{n'}$ and $\vec{\zeta}_T = (g_1^T, \dots, g_\ell^T) \in \mathbb{G}_T^\ell$ is called algebraic in an asymmetric bilinear group if it always outputs:*

$$H = \left(h_1, \dots, h_m, h'_1, \dots, h'_{m'}, h_1^T, \dots, h_{\ell'}^T\right) \in \mathbb{G}_1^{n'} \times \mathbb{G}_2^{m'} \times \mathbb{G}_T^{\ell'} \ ,$$

along with a representation vector of size $n \cdot m + n' \cdot m' + \ell'(n \cdot n' + \ell)$, as follows:

$$\vec{H} = \left((\alpha_{ij})_{\substack{i\in[1,m]\\ j\in[1,n]}}, (\beta_{ij})_{\substack{i\in[1,m']\\ j\in[1,n']}}, (\gamma_{ijk})_{\substack{i\in[1,\ell']\\ j\in[1,n]\\ k\in[1,n']}}, (\gamma'_{ij})_{\substack{i\in[1,\ell']\\ j\in[1,\ell]}} \right) \in \mathbb{Z}_p \ ,$$

such that, $h_i = \prod_{j=1}^{n} g_j^{\alpha_{ij}}$ *for* $i \in [1, m]$, $h'_i = \prod_{j=1}^{n'} (g'_j)^{\beta_{ij}}$ *for* $i \in [1, m']$ *and* $h_i^T = \prod_{j=1}^{n} \sum_{k=1}^{n'} e(h_j, h'_k)^{\gamma_{ijk}} \cdot \prod_{j=1}^{\ell} (h_i^T)^{\gamma'_{ij}}$ *for* $i \in [1, \ell']$*. We denote the outputs and their representations as* $(H; \vec{H}) \leftarrow\$ \mathcal{A}_{alg}(\vec{\zeta}_1, \vec{\zeta}_2, \vec{\zeta}_T)$*.*

With regard to the representations that the algebraic adversary $\mathcal{A}_{alg}$ outputs, we provide some additional notation. Let $\mathcal{A}_{alg}$ take the vectors of group elements $(g_1, \ldots, g_n) \in \mathbb{G}_1^n$, $(g'_1, \ldots, g'_m) \in \mathbb{G}_2^{n'}$ and $(g_{T1}, \ldots, g_{T\ell}) \in \mathbb{G}_T^{\ell}$ as inputs. By Definition 18, when $\mathcal{A}_{alg}$ outputs the group elements $\big(h_1, \ldots, h_{n'}, h'_1, \ldots, h'_{m'}, h_1^T, \ldots, h_{\ell'}^T\big)$, with $h_i \in \mathbb{G}_1$, $h'_i \in \mathbb{G}_2$, $h_i^T \in \mathbb{G}_T$, for each element $h_i \in \mathbb{G}_1$ (and similarly for the other groups), $\mathcal{A}_{alg}$ must also output the corresponding representation $\vec{h}_i = (\eta_{i1}, \ldots, \eta_{in}) \in \mathbb{Z}_p^n$, such that $h_i = \prod_{j=1}^{n} g_j^{\eta_{ij}}$. The representation element $\eta_{ij} \in \mathbb{Z}_p$ base $g_j \in \mathbb{G}_1$ for $j \in [1, n]$ is denoted by $\vec{h}_i[g_j]$.

Assume that each $\mathsf{dlog}_g(g_j)$ for any $g_j \in \mathbb{G}_1$ (and similarly for the other groups) can be represented as the evaluation on $\vec{x} = (x_1, \ldots, x_k)$ of a k-variant polynomial $\mathbf{P}_j$ from the ring $\mathbb{Z}_p[\vec{\mathbf{X}}]$, $\vec{\mathbf{X}} = (\mathbf{X}_1, \ldots, \mathbf{X}_k)$, i.e., $\mathsf{dlog}_g(g_j) = \mathbf{P}_j(\vec{x})$. $\mathbf{P}_j(\vec{x}) = 0$ means that the polynomial evaluates to 0 at point $\vec{x}$, while $\mathbf{P}_j(\vec{\mathbf{X}}) \equiv 0$ means that $\mathbf{P}_j(\vec{\mathbf{X}})$ is the zero polynomial.

Then we can define the polynomial $\mathbf{P}_{\vec{h}_i}(\vec{\mathbf{X}}) = \sum_j \mathbf{P}_j(\vec{\mathbf{X}}) \cdot \vec{h}_i[g_j]$ that evaluates on $\vec{x} = (x_1, \ldots, x_k)$ to $\mathsf{dlog}_g(h_i)$:

$$\mathsf{dlog}_g(h_i) = \sum_{j=1}^{n} \Big(\mathsf{dlog}_g(g_j) \cdot \vec{h}_i[g_j] \Big) = \sum_j \mathbf{P}_j(\vec{x}) \cdot \vec{h}_i[g_j] = \mathbf{P}_{\vec{h}_i}(\vec{x}) \ .$$

Similar to [KSAP22], we define the GPS_3 assumption in the AGM (Figure 15). Compared to the GPS_3 game in Figure 2, there are three main differences:

1. The first difference is the use of an extractor Ext, defined as a deterministic polynomial algorithm in the second oracle $\mathcal{O}_1^{\mathrm{GPS}_3}(\cdot)$. For the j^{th} query, $\mathsf{Ext}(.)$ takes as input three source group elements $h_j, M_{j1}, M_{j2} \in \mathbb{G}_1^2 \times \mathbb{G}_2$ along with their representations $\vec{h}_j, \vec{M}_{j1}, \vec{M}_{j2}$. It then returns a scalar $m_j \in \mathbb{Z}_p$ such that $M_{j1} = h_j^{m_j}$ and $M_{j2} = \hat{g}^{m_j}$, or it returns $\perp$ whenever the extraction fails. Ext succeeds in extracting the scalar m_j because, under the conditions shown in lines 1 and 2 of $\mathsf{Ext}(.)$ in Figure 15, if the extraction fails, then the $(2, 1)$-DL problem is no longer hard (Claim 2). With Ext, the oracle $\mathcal{O}_1^{\mathrm{GPS}_3}(\cdot)$ can provide the appropriate responses to $\mathcal{A}_{alg}$'s queries.

$\mathbf{G}^{\mathrm{GPS}_3}_{\mathcal{A}_{alg}}(1^\kappa)$

$$\begin{array}{l}
\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa) \\
x \leftarrow\!\!\$\, \mathbb{Z}_p^* \sim \mathbf{X}, y \leftarrow\!\!\$\, \mathbb{Z}_p^* \sim \mathbf{Y} \\
\Big((M_1^*, M_2^*, h^*, s^*), \boxed{(\vec{M_1^*}, \vec{M_2^*}, \vec{h^*}, \vec{s^*})}\Big) \leftarrow \mathcal{A}_{alg}^{\mathcal{O}_0^{\mathrm{GPS}_3}, \mathcal{O}_1^{\mathrm{GPS}_3}}(\mathsf{pp}, \hat{g}^x, \hat{g}^y) \\
\text{// arrows denote representation vectors} \\
\textbf{if} \quad (1)\ M_1^*, h^* \neq 1_{\mathbb{G}_1} \ \wedge \ M_2^* \neq 1_{\mathbb{G}_2} \ \wedge \\
\qquad (2) \boxed{\mathbf{P}_{\vec{s^*}}(\vec{\mathbf{X}}) - \Big(\mathbf{X}\mathbf{P}_{\vec{h^*}}(\vec{\mathbf{X}}) + \mathbf{Y}\mathbf{P}_{\vec{M_1^*}}(\vec{\mathbf{X}})\Big) \equiv 0} \wedge \\
\qquad (3) \boxed{\mathbf{P}_{\vec{M_2^*}}(\vec{\mathbf{X}})\mathbf{P}_{\vec{h^*}}(\vec{\mathbf{X}}) - \mathbf{P}_{\vec{M_1^*}}(\vec{\mathbf{X}}) \equiv 0} \wedge \\
\qquad (4)\ (\star, M_2^*) \notin \mathcal{Q}_1 : \\
\qquad \textbf{return } 1 \\
\textbf{else} \ : \\
\qquad \textbf{return } 0
\end{array}$$

$\mathcal{O}_0^{\mathrm{GPS}_3}()$ $//j^{th}$ query

$$\begin{array}{l}
r_j \leftarrow\!\!\$\, \mathbb{Z}_p^* \sim \mathbf{R}_j \\
h_j \leftarrow g^{r_j} \\
\mathcal{Q}_0 \leftarrow \mathcal{Q}_0 \cup \{h_j\} \\
\textbf{return } h_j
\end{array}$$

$\mathsf{Ext}\Big((M_{j1}; \vec{M}_{j1}), (M_{j2}; \vec{M}_{j2})\Big)$

$$\begin{array}{l}
\textbf{if} \ \Big(\mathbf{P}_{\vec{M}_{j1}}(\vec{\mathbf{X}}) - \mathbf{R}_j \mathbf{P}_{\vec{M}_{j2}}(\vec{\mathbf{X}})\Big) \not\equiv 0 : \\
\qquad \textbf{return } \perp \\
\textbf{else} \ : \\
\qquad \textbf{return } \vec{M}_{j2}[\hat{g}]
\end{array}$$

$\mathcal{O}_1^{\mathrm{GPS}_3}((h_j, M_{j1}, M_{j2}), \boxed{(\vec{h}_j, \vec{M}_{j1}, \vec{M}_{j2})})$ $//$ j^{th} query

$$\begin{array}{l}
\textbf{if} \quad h_j \notin \mathcal{Q}_0 \ \vee \ (h_j, \star) \in \mathcal{Q}_1 : \\
\qquad \textbf{return } \perp \\
\textbf{if} \quad e(M_{j1}, \hat{g}) \neq e(h_j, M_{j2}) : \\
\qquad \textbf{return } \perp \\
\textbf{else} \ : \\
\qquad \boxed{m_j \leftarrow \mathsf{Ext}\Big((M_{j1}; \vec{M}_{j1}), (M_{j2}; \vec{M}_{j2})\Big)} \\
\qquad s_j \leftarrow h_j^x M_{j1}^y \boxed{= h_j^{x + m_j y}} \\
\qquad \mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{(h_j, M_{j2})\} \\
\qquad \textbf{return } s_j
\end{array}$$

Fig. 15: Game defining the GPS_3 assumption in the AGM. The GPS_3 assumption (Figure 2) includes all but the dashed boxes, with different winning conditions (2) and (3).

2. The second difference is that the second condition in the GPS_3 game in Figure 2, namely $s^* = {h^*}^x {M_1^*}^y$, can be written as $\mathsf{dlog}_g(s^*) = x\mathsf{dlog}_g(h^*) + y\mathsf{dlog}_g(M_1^*)$ and validated by checking whether the polynomial $\mathbf{P}_{\vec{s}^*}(\vec{\mathbf{X}}) - \left(\mathbf{X}\mathbf{P}_{\vec{h}^*}(\vec{\mathbf{X}}) + \mathbf{Y}\mathbf{P}_{\vec{M}_1^*}(\vec{\mathbf{X}})\right)$ is the zero polynomial or not.
3. The third difference is that the third condition in the GPS_3 game in Figure 2 can be written as $\mathsf{dlog}_{\hat{g}}(M_2^*) = \mathsf{dlog}_{h^*}(M_1^*) = \frac{\mathsf{dlog}_g(M_1^*)}{\mathsf{dlog}_g(h^*)}$ and validated by checking $\mathbf{P}_{\vec{M}_2^*}(\vec{\mathbf{X}}) = \mathsf{dlog}_{\hat{g}}(M_2^*) = \frac{\mathsf{dlog}_g(M_1^*)}{\mathsf{dlog}_g(h^*)} = \frac{\mathbf{P}_{\vec{M}_1^*}(\vec{\mathbf{X}})}{\mathbf{P}_{\vec{h}^*}(\vec{\mathbf{X}})}$, i.e., whether $\mathbf{P}_{\vec{h}^*}(\vec{\mathbf{X}})\mathbf{P}_{\vec{M}_2^*}(\vec{\mathbf{X}}) - \mathbf{P}_{\vec{M}_1^*}(\vec{\mathbf{X}})$ is the zero polynomial or not.

Definition 19 (GPS_3 Assumption in the AGM). *Let the advantage of an adversary $\mathcal{A}$ against the GPS_3 game $\mathbf{G}^{GPS_3}_{\mathcal{A}_{alg}}$, as defined in Figure 15, be as follows:*

$$Adv^{GPS_3}_{\mathcal{A}_{alg}}(\kappa) = \Pr\left[\mathbf{G}^{GPS_3}_{\mathcal{A}_{alg}} = 1\right] \ .$$

The GPS_3 assumption holds if for all algebraic adversaries $\mathcal{A}$, there exists a negligible function ν such that $Adv^{GPS_3}_{\mathcal{A}_{alg}}(\kappa) < \nu(\kappa)$.

B Proofs

B.1 Proof of Theorem 1

Proof. We wish to show that if there exists an algbraic adversary $\mathcal{A}_{alg}$ that breaks the GPS_3 assumption (Figure 15) with non-negligible probability, then we can construct an algebraic adversary $\mathcal{B}_{alg}$ that breaks the $(2,1)$-DL assumption (Definition 4) with non-negligible probability.

Suppose there exists such an adversary $\mathcal{A}_{alg}$. Then, running $\mathcal{A}_{alg}$ as a subroutine, we construct a reduction $\mathcal{B}_{alg}$ breaking the $(2,1)$-DL assumption as follows.

The reduction $\mathcal{B}_{alg}$ is responsible for simulating oracle responses for queries to $\mathcal{O}_0^{\mathrm{GPS}_3}()$ and $\mathcal{O}_1^{\mathrm{GPS}_3}(\cdot)$. Let $\mathcal{Q}_0, \mathcal{Q}_1$ be the set of $\mathcal{O}_0^{\mathrm{GPS}_3}()$ and $\mathcal{O}_1^{\mathrm{GPS}_3}(\cdot)$ queries, and q_0 and q_1 the maximum number of queries to each of them, respectively. Without loss of generality, we assume $\mathcal{A}_{alg}$ always queries $\mathcal{O}_0^{\mathrm{GPS}_3}()$ to receive h_j prior to querying $\mathcal{O}_1^{\mathrm{GPS}_3}\left((h_j;\vec{h}_j), (M_{j1};\vec{M}_{j1}), (M_{j2};\vec{M}_{j2})\right)$ for some $h_j, \vec{M}_{j1}, \vec{M}_{j2}$. $\mathcal{B}_{alg}$ initializes $\mathcal{Q}_0, \mathcal{Q}_1$ to the empty set.

Initialization. $\mathcal{B}_{alg}$ takes as input the group description $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$ and a $(2,1)$-DL challenge $(Z_1, Z_1', Z_2) = (g^z, g^{z^2}, \hat{g}^z) \in \mathbb{G}_1^2 \times \mathbb{G}_2$. $\mathcal{B}_{alg}$ simulates the GPS_3 instance $(X,Y) \leftarrow (\hat{g}^x, \hat{g}^y) = (Z_2^{a_0}\hat{g}^{b_0}, Z_2^{a_0'}\hat{g}^{b_0'}) \in \mathbb{G}_2^2$ by implicitly setting $x \leftarrow a_0 z + b_0$ and $y \leftarrow a_0' z + b_0'$, where $a_0, b_0, a_0', b_0' \leftarrow\$\, \mathbb{Z}_p$. $\mathcal{B}_{alg}$ runs $\mathcal{A}_{alg}(\mathsf{pp}, (X,Y))$.

Simulating Oracle Queries. $\mathcal{B}_{alg}$ simulates the defined oracles as follows:

– **Oracle** $\mathcal{O}_0^{\mathrm{GPS}_3}()$: To simulate the i^{th} query s.t. $i \in [1, q_0]$, $\mathcal{B}_{alg}$ samples $a_i, b_i \leftarrow\$ \mathbb{Z}_p$ and assigns $h_i \leftarrow Z_1^{a_i} g^{b_i}$, which implicitly sets $r_i \leftarrow a_i z + b_i$.
– **Oracle** $\mathcal{O}_1^{\mathrm{GPS}_3}\left((h_j; \vec{h}_j), (M_{j1}; \vec{M}_{j1}), (M_{j2}; \vec{M}_{j2})\right)$: To simulate the j^{th} query for $j \geq 2$, we assume that $\mathcal{B}_{alg}$ has successfully simulated the $j-1$ previous queries to this oracle. Thus, the algebraic adversary $\mathcal{A}_{alg}$ has access to $\{s_\ell = g^{r_\ell(x+m_\ell y)}\}_{\ell=1}^{j-1}$, where $r_\ell = \mathsf{dlog}_g(h_\ell)$ and m_ℓ is the scalar message extracted by $\mathsf{Ext}(.)$ of Figure 15.
$\mathcal{A}_{alg}$ makes the j^{th} query to the oracle $\mathcal{O}_1^{\mathrm{GPS}_3}(.)$ by providing the tuple $\left((h_j; \vec{h}_j), (M_{j1}; \vec{M}_{j1}), (M_{j2}; \vec{M}_{j2})\right)$. Note that $\vec{h}_j = r_j$, i.e., $P_{\vec{h}_j}(\vec{\mathbf{X}}) = \mathbf{R}_j$. $\vec{M}_{j1}$ and $\vec{M}_{j2}$ determine the following two polynomials:

$$\mathbf{P}_{\vec{M}_{j1}}(\vec{\mathbf{X}}) = \vec{M}_{j1}[g] + \sum_{\ell=1}^{|\mathcal{Q}_0|} \mathbf{R}_\ell \vec{M}_{j1}[h_\ell] + \sum_{\ell=1}^{j-1} \mathbf{R}_\ell(\mathbf{X} + m_\ell \mathbf{Y}) \vec{M}_{j1}[s_\ell] \ ,$$
$$\mathbf{P}_{\vec{M}_{j2}}(\vec{\mathbf{X}}) = \vec{M}_{j2}[\hat{g}] + \mathbf{X}\vec{M}_{j2}[X] + \mathbf{Y}\vec{M}_{j2}[Y] \ ,$$

where $|\mathcal{Q}_0|$ is the number of $\mathcal{O}_0^{\mathrm{GPS}_3}()$ queries made thus far, and $\vec{\mathbf{X}} = (\mathbf{X}, \mathbf{Y}, \mathbf{R}_1, \ldots, \mathbf{R}_{|\mathcal{Q}_0|})$. Representation coefficients for these polynomials are well defined, as $\mathcal{A}_{alg}$ has access to the answers received from the oracles, $\{h_\ell = g^{r_\ell}\}_{\ell=1}^{|\mathcal{Q}_0|}$ and $\{s_\ell = g^{r_\ell(x+m_\ell y)}\}_{\ell=1}^{j-1}$, of the first source group elements and the GPS_3 instances of the second source group elements. As shown in Figure 15, the oracle $\mathcal{O}_1^{\mathrm{GPS}_3}(.)$ does not fail if $e(h_j, M_{j2}) = e(M_{j1}, \hat{g})$, which implies that the polynomial $\mathbf{P}_j(\vec{\mathbf{X}}) = \mathbf{P}_{\vec{M}_j 1}(\vec{\mathbf{X}}) - \mathbf{R}_j \mathbf{P}_{\vec{M}_j 2}(\vec{\mathbf{X}})$ should vanish at least on point $\vec{x} = (x, y, r_1, \ldots, r_{|\mathcal{Q}_0|})$. We define the event E as $\mathbf{P}_j(\vec{\mathbf{X}}) \equiv 0$. There are two possible cases: 1) If the event E holds, i.e., $\mathbf{P}_j(\vec{\mathbf{X}}) \equiv 0$, then the defined extractor can successfully extract the scalar messages m_j for $j \in [1, q_1]$, or 2) if the event does not hold, $\neg E$, i.e., $\mathbf{P}_j(\vec{\mathbf{X}}) \not\equiv 0$, then we can define an algebraic adversary $\mathcal{D}_{alg}$ that can solve the $(2,1)$-DL problem with a non-negligible advantage. We formally discuss these two cases in the following claims.

Claim 1 *If $\mathbf{P}_j(\vec{\mathbf{X}}) \equiv 0$, then the extractor can successfully extract the scalar messages m_j.*

Proof. Similar to the proof of [KSAP22, Theorem 2, Claim 1], the condition $\mathbf{P}_j(\vec{\mathbf{X}}) \equiv 0$ implies the equality $\mathbf{P}_{\vec{M}_j 1}(\vec{\mathbf{X}}) = \mathbf{R}_j \mathbf{P}_{\vec{M}_j 2}(\vec{\mathbf{X}})$ must hold. Thus, based on the received representations from $\mathcal{A}_{alg}$, we can write:

$$\vec{M}_{j1}[g] + \left(\sum_{\ell=1}^{q_0 \setminus \{j\}} \mathbf{R}_\ell \vec{M}_{j1}[h_\ell]\right) + \mathbf{R}_j \vec{M}_{j1}[h_j] + \sum_{\ell=1}^{j-1} \mathbf{R}_\ell \vec{M}_{j1}[s_\ell](\mathbf{X} + m_\ell \mathbf{Y})$$
$$= \mathbf{R}_j \left(\vec{M}_{j2}[\hat{g}] + \mathbf{X}\vec{M}_{j2}[X] + \mathbf{Y}\vec{M}_{j2}[Y]\right) \ .$$

This implies:

$$\begin{aligned}
&\vec{M}_{j2}[\hat{g}] = \vec{M}_{j1}[h_j] \text{ (Due to } \mathbf{R}_j) \ , \\
&\vec{M}_{j1}[g] = 0 \text{ (Due to } \mathbf{R}_j) \ , \\
&\vec{M}_{j2}[X] = 0 \text{ (Due to } \mathbf{R}_j\mathbf{X}) \ , \\
&\vec{M}_{j2}[Y] = 0 \text{ (Due to } \mathbf{R}_j\mathbf{Y}) \ , \\
&\left\{\vec{M}_{j1}[h_\ell] = 0\right\}_{\forall\ \ell \in [1,|\mathcal{Q}_0|], \ell \neq j} \text{ (Due to } \mathbf{R}_\ell) \ , \\
&\left\{\vec{M}_{j1}[s_\ell] = 0\right\}_{\forall\ \ell \in [1,j-1]} \text{ (Due to } \mathbf{R}_\ell\mathbf{X}) \ .
\end{aligned}$$

Finally, based on the above equations, we can write $M_{j2} = \hat{g}^{\vec{M}_{j2}[\hat{g}]}$ and $M_{j1} = h_j^{\vec{M}_{j1}[h_j]}$, and therefore the extractor returns $m_j \leftarrow \vec{M}_{j2}[\hat{g}] = \vec{M}_{j1}[h_j]$ as the scalar message. □

In this case, $\mathcal{B}_{alg}$ responds to the j^{th} query to $\mathcal{O}_1^{\mathrm{GPS}_3}(.)$ by computing:

$$\begin{aligned}
s_j &= (Z_1')^{a_j(a_0+a_0'm_j)} Z_1^{a_jb_0+b_ja_0+m_j(a_jb_0'+b_ja_0')} g^{a_j(a_0+a_0'm_j)} \\
&= g^{(a_j+zb_j)(a_0+zb_0)+m_j(a_j+zb_j)(a_0'+zb_j')} \\
&= g^{(a_j+zb_j)(a_0+zb_0)} g^{m_j(a_j+zb_j)(a_0'+zb_j')} \\
&= h_j^x M_{j1}^y \ .
\end{aligned}$$

Claim 2 *If $\mathbf{P}_j(\vec{\mathbf{X}}) \not\equiv 0$, i.e., the extractor fails, then the $(2,1)$-DL problem is not hard.*

Proof. Based on the fact that $x = a_0z + b_0$, $y = a_0'z + b_0'$ and $\{r_\ell = a_\ell z + b_\ell\}_{\ell=1}^{q_0}$, we can convert the variables $\mathbf{X}$, $\mathbf{Y}$ and $\{\mathbf{R}_\ell\}_{\ell=1}^{q_0}$ to $\mathbf{A_0Z} + \mathbf{B_0}$, $\mathbf{A_0'Z} + \mathbf{B_0'}$ and $\{\mathbf{A}_\ell\mathbf{Z} + \mathbf{B}_\ell\}_{\ell=1}^{q_0}$ and define a univariate polynomial $\mathbf{G}_j^*(\vec{\mathbf{Z}})$ from the polynomial $\mathbf{P}_j^*(\vec{\mathbf{X}})$. If $\mathbf{G}_j^*(\vec{\mathbf{Z}}) \not\equiv 0$, then the equality $\mathbf{G}_j^*(\vec{z}) = \mathbf{P}_j^*(\vec{x})$ implies that $\mathbf{G}_j^*(\vec{\mathbf{Z}})$ has at least one root like $\vec{z}$. Similar to the analysis in the proof of [KSAP22, Theorem 2, Claim 2], by the Schwartz-Zippel lemma, $\Pr[\mathbf{G}_j^*(\vec{\mathbf{Z}}) \equiv 0] \leq \Pr[a_3 = 0] \leq 3/p$, where a_3 is the leading coefficient of $\mathbf{G}_j^*(\vec{\mathbf{Z}})$. In the case of $\mathbf{G}_j^*(\vec{\mathbf{Z}}) \not\equiv 0$, there exists a vector $\vec{z}$ as a root for this polynomial that can be the solution of the $(2,1)$-DL problem. Thus, we can write:

$$\begin{aligned}
&\Pr[\neg E] \leq \Pr\left[\mathbf{P}_j^*(\vec{\mathbf{X}}) \not\equiv 0 \wedge \mathbf{G}_j^*(\vec{\mathbf{Z}}) \not\equiv 0\right] + \Pr\left[\mathbf{G}_j^*(\vec{\mathbf{Z}}) \equiv 0\right] \leq \\
&Adv_{\mathcal{D}}^{(2,1)\text{-DL}}(\kappa) + 3/p \ .
\end{aligned}$$

Thus, if the extractor fails, then we can define an algebraic algorithm $\mathcal{D}_{alg}$ that can solve the $(2,1)$-DL problem with a non-negligible advantage. □

$\mathcal{B}_{alg}$ fails to answer $\mathcal{A}_{alg}$'s query to $\mathcal{O}_1^{\mathrm{GPS}_3}(\cdot)$ in this case because it does not know the scalar message m_j and cannot return a valid s_j, but the probability of this occurring is negligible $(3/p)$.

We can conclude that $\mathcal{B}_{alg}$ successfully simulates the defined oracles for the adversary $\mathcal{A}_{alg}$ as long as the $(2,1)$-DL problem is hard.

Output. $\mathcal{A}_{alg}$ finally outputs $\Big((h^*;\vec{h}^*),(M_1^*;\vec{M}_1^*),(M_2^*;\vec{M}_2^*),(s^*;\vec{s^*})\Big)$ based on the received responses from the oracles and public parameters. From the received representations, we can write:

$$\mathbf{P}_{\vec{M}_1^*}(\vec{\mathbf{X}}) = \vec{M}_1^*[g] + \sum_{\ell=1}^{q_0} \mathbf{R}_\ell \vec{M}_1^*[h_\ell] + \sum_{\ell=1}^{q_1} \mathbf{R}_\ell \vec{M}_1^*[s_\ell](\mathbf{X}+m_\ell\mathbf{Y}) \ ,$$
$$\mathbf{P}_{\vec{M}_2^*}(\vec{\mathbf{X}}) = \vec{M}_2^*[\hat{g}] + \mathbf{X}\vec{M}_2^*[X] + \mathbf{Y}\vec{M}_2^*[Y] \ ,$$
$$\mathbf{P}_{\vec{h}^*}(\vec{\mathbf{X}}) = \vec{h}^*[g] + \sum_{\ell=1}^{q_0} \mathbf{R}_\ell \vec{h}^*[h_\ell] + \sum_{\ell=1}^{q_1} \mathbf{R}_\ell \vec{h}^*[s_\ell](\mathbf{X}+m_\ell\mathbf{Y}) \ ,$$
$$\mathbf{P}_{\vec{s^*}}(\vec{\mathbf{X}}) = \vec{s^*}[g] + \sum_{\ell=1}^{q_0} \mathbf{R}_\ell \vec{s^*}[h_\ell] + \sum_{\ell=1}^{q_1} \mathbf{R}_\ell \vec{s^*}[s_\ell](\mathbf{X}+m_\ell\mathbf{Y}) \ .$$

As discussed in Appendix A.3, the assumption in Definition 19 is identical to the assumption in Definition 6, except the second and third conditions are defined by polynomial evaluations. Next, we describe three events to cover all possible scenarios.

1. Event E_1: All the conditions described in Figure 2 are fulfilled and the extractor $\mathsf{Ext}(.)$ does not fail.
2. Event E_2: The polynomial $\mathbf{P}_2^*(\vec{\mathbf{X}}) = \mathbf{P}_{\vec{s^*}}(\vec{\mathbf{X}}) - \Big(\mathbf{X}\mathbf{P}_{\vec{h}^*}(\vec{\mathbf{X}}) + \mathbf{Y}P^*_{\vec{M}_1^*}(\vec{\mathbf{X}})\Big)$ is the zero polynomial.
3. Event E_3: The polynomial $\mathbf{P}_3^*(\vec{\mathbf{X}}) = \mathbf{P}_{\vec{M}_2^*}(\vec{\mathbf{X}})\mathbf{P}_{\vec{h}^*}(\vec{\mathbf{X}}) - \mathbf{P}_{\vec{M}_1^*}(\vec{\mathbf{X}})$ is the zero polynomial.

Claim 3 $\Pr[E_1 \wedge E_2 \wedge E_3] = 0$.

Proof. Similar to the proof of [KSAP22, Theorem 2, Claim 3], we suppose all three conditions occur and arrive at a contradiction. From the second condition, $\mathbf{P}_2^*(\vec{\mathbf{X}}) \equiv 0$, we can deduce that the degree of both polynomials $\mathbf{P}_{\vec{h}^*}(\vec{\mathbf{X}}) = \sum_{\ell=1}^{q_1} \mu_\ell \mathbf{R}_\ell$ and $\mathbf{P}_{\vec{M}_1^*}(\vec{\mathbf{X}}) = \sum_{\ell=1}^{q_1} \mathbf{R}_\ell \vec{M}_1^*[h_\ell]$ should be equal to 1, and we can write $\mathbf{P}_{\vec{s^*}}(\vec{\mathbf{X}}) = \sum_{\ell=1}^{q_1} \mu_\ell(\mathbf{X}+m_\ell\mathbf{Y})\mathbf{R}_\ell$, where $\mu_\ell \leftarrow \vec{h}^*[h_\ell] = \vec{s^*}[s_\ell]$. Additionally, from the third condition, $\mathbf{P}_3^*(\vec{\mathbf{X}}) \equiv 0$, we can deduce that the degree of polynomial $\mathbf{P}_{\vec{M}_2^*}(\vec{\mathbf{X}})$ should be equal to zero, as we have $\mathbf{P}_{\vec{M}_2^*}(\vec{\mathbf{X}})\mathbf{P}_{\vec{h}^*}(\vec{\mathbf{X}}) = \mathbf{P}_{\vec{M}_1^*}(\vec{\mathbf{X}})$. More precisely, polynomials $\mathbf{P}_{\vec{h}^*}(\vec{\mathbf{X}})$ and $\mathbf{P}_{\vec{M}_1^*}(\vec{\mathbf{X}})$ have degree 1, and to fulfil the third condition, the polynomial $\mathbf{P}_{\vec{M}_2^*}(\vec{\mathbf{X}})$ should be constant. Thus, we can write $\mathbf{P}_{\vec{M}_2^*}(\vec{\mathbf{X}}) = \vec{M}_2^*[\hat{g}]$ and denote by m^*, and according the first condition in Figure 15, $m^* \neq 0$. Thus, we can reform the polynomial $\mathbf{P}_2^*(\vec{\mathbf{X}})$ as $\mathbf{P}_2^*(\vec{\mathbf{X}}) = \mathbf{P}_{\vec{s}^*}(\vec{\mathbf{X}}) - \mathbf{P}_{\vec{h}^*}(\vec{\mathbf{X}})\,(\mathbf{X} + m^*\mathbf{Y})$.

Putting everything together, we have:

$$\begin{aligned}\mathbf{P}_2^*(\vec{\mathbf{X}}) &= \sum_{\ell=1}^{q_1} \mu_\ell\,(\mathbf{X} + m_\ell \mathbf{Y})\,\mathbf{R}_\ell - \sum_{\ell=1}^{q_1} \mu_\ell\,(\mathbf{X} + m^*\mathbf{Y})\,\mathbf{R}_\ell \\ &= \sum_{\ell=1}^{q_1} \mu_\ell\,(m_\ell - m^*)\,\mathbf{Y}\mathbf{R}_\ell \ .\end{aligned}$$

As we have shown, the polynomial $\mathbf{P}_{\vec{h}^*}(\vec{\mathbf{X}})$ has degree 1, so there exists at least one non-zero μ_ℓ for some $\ell \in [1, q_1]$. Thus, to have $\mathbf{P}_2^*(\vec{\mathbf{X}}) \equiv 0$, we must have $m_\ell = m^*$ for some $\ell \in [1, q_1]$. As it was shown before that $M_{2\ell} = \hat{g}^{m_\ell}$ and $M_2^* = \hat{g}^{m^*}$, the last condition of $(\star, M_2^*) \notin \mathcal{Q}_1$ cannot be fulfilled, as $(\star, M_{2\ell})$ is already recorded in $\mathcal{Q}_1$.

Thus, we can conclude that all three events cannot occur simultaneously, i.e., $\Pr[E_1 \wedge E_2 \wedge E_3] = 0$. □

Claim 4 $\Pr[E_1 \wedge \neg E_2] + \Pr[E_1 \wedge E_2 \wedge \neg E_3] \leq Adv_{\mathcal{D}_{alg}}^{(2,1)\text{-}DL}(\kappa) + 7/p$.

Proof. Similar to the proof of [KSAP22, Theorem 2, Claim.4], if the event E_2 does not occur, i.e., $\mathbf{P}_2^*(\vec{\mathbf{X}}) \not\equiv 0$, then for random values $a, b \leftarrow\$\, \mathbb{Z}_p$, we can form a polynomial $\mathbf{G}_2^*(\vec{\mathbf{Z}})$ by changing the variables of $\mathbf{P}_2^*(\vec{\mathbf{X}})$ to $\vec{\mathbf{Z}} \leftarrow \mathbf{A}\mathbf{Z} + \mathbf{B}$. As discussed in the proof of Claim 2, one of the roots of the univariate polynomial $\mathbf{G}_2^*(\vec{\mathbf{Z}})$ should be a valid solution for the $(2, 1)$-DL assumption. Moreover, by the Schwartz-Zippel lemma, the probability of the event that the polynomial $\mathbf{G}_2^*(\vec{\mathbf{Z}})$ is the zero polynomial is bounded by $3/p$. We can similarly show that if the third condition does not hold, i.e., $\mathbf{P}_3^*(\vec{\mathbf{X}}) \not\equiv 0$, then we can define the non-zero univariate polynomial $\mathbf{G}_3^*(\vec{\mathbf{Z}})$ that enables a valid solution for the $(2, 1)$-DL problem. Similarly, by the Schwartz-Zippel lemma, the probability of $\mathbf{G}_3^*(\vec{\mathbf{Z}}) \equiv 0$ is at most $4/p$. Thus, we can define an efficient algorithm against the hardness of $(2, 1)$-DL assumption, $\mathcal{D}_{alg}$, that fails with probability $7/p$, which completes the claim. □

Thus,

$$\begin{aligned}\Pr[E_1] &= \Pr[E_1 \wedge E_2 \wedge E_3] + \Pr[E_1 \wedge \neg E_2] + \Pr[E_1 \wedge E_2 \wedge \neg E_3] \\ &= \Pr[E_1 \wedge \neg E_2] + \Pr[E_1 \wedge E_2 \wedge \neg E_3] \leq Adv_{\mathcal{D}_{alg}}^{(2,1)\text{-}DL}(\kappa) + 7/p \ ,\end{aligned}$$

and

$$\begin{aligned}Adv_{\mathcal{A}_{alg}}^{\mathrm{GPS}_3}(\kappa) &= \Pr[\mathbf{G}_{\mathcal{A}_{alg}}^{\mathrm{GPS}_3} = 1 \wedge \neg E] + \Pr[\mathbf{G}_{\mathcal{A}_{alg}}^{\mathrm{GPS}_3} = 1 \wedge E] \\ &\leq Adv_{\mathcal{D}_{alg}}^{(2,1)\text{-DL}}(\kappa) + 3/p + Adv_{\mathcal{D}_{alg}}^{(2,1)\text{-DL}}(\kappa) + 7/p \\ &\leq 2Adv_{\mathcal{D}_{alg}}^{(2,1)\text{-DL}}(\kappa) + 10/p \ .\end{aligned}$$

□

B.2 Proof of Theorem 2

Proof. **Correctness.** If $e(h, M_2) = e(h, \hat{g}^m) = e(h^m, \hat{g}) = e(M_1, \hat{g})$, then for $\sigma = (h, s) = (h, h^{x+my})$, we have $e(h, \mathsf{vk}_1)e(M_1, \mathsf{vk}_2) = e(h, \hat{g}^x)e(h^m, \hat{g}^y) = e(h, \hat{g})^{x+my} = e(h^{x+my}, \hat{g}) = e(h^x M_1^y, \hat{g}) = e(s, \hat{g})$.

EUF-CiMA **Security.** We wish to show that if there exists a PPT adversary $\mathcal{A}$ that breaks the EUF-CiMA security (Figure 5) of the indexed message SPS scheme IM-SPS (Figure 6) with non-negligible probability, then we can construct a PPT adversary $\mathcal{A}$' that breaks the GPS_3 assumption (Figure 13) with non-negligible probability.

Suppose there exists such a PPT adversary $\mathcal{A}$. Then, we construct a PPT adversary $\mathcal{A}$' as a reduction $\mathcal{B}$ running $\mathcal{A}$ as a subroutine. We construct the reduction $\mathcal{B}$ for breaking the GPS_3 assumption as follows.

The reduction $\mathcal{B}$ is responsible for simulating oracle responses for queries to $\mathcal{O}_{\mathsf{Sign}}(\cdot)$ and H. Let $\mathcal{Q}_{\mathsf{H}}$ be the set of H queries and their responses. $\mathcal{B}$ may program the random oracle H. Let $\mathcal{Q}_{\mathsf{S}}$ be the set of messages $(id, \tilde{M})$ that have been queried in $\mathcal{O}_{\mathsf{Sign}}(\cdot)$ and $\mathcal{Q}_{\mathsf{EQ}}$ the set of equivalence classes of messages $\tilde{M}$. $\mathcal{B}$ initializes $\mathcal{Q}_{\mathsf{H}}, \mathcal{Q}_{\mathsf{S}}, \mathcal{Q}_{\mathsf{EQ}}$ to the empty set.

Initialization. $\mathcal{B}$ takes as input the group description $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$ and a GPS_3 challenge $(\hat{g}^x, \hat{g}^y)$. As in game $\mathbf{G}_{\mathcal{B}}^{\mathrm{GPS}_3}(1^\kappa)$, $\mathcal{B}$ has access to oracles $\mathcal{O}_0^{\mathrm{GPS}_3}()$ and $\mathcal{O}_1^{\mathrm{GPS}_3}(\cdot)$. $\mathcal{B}$ sets the IM-SPS verification key $\mathsf{vk} \leftarrow (\mathsf{vk}_1, \mathsf{vk}_2) = (\hat{g}^x, \hat{g}^y)$ and runs $\mathcal{A}(\mathsf{pp}, \mathsf{vk})$.

Simulating Oracle Queries. $\mathcal{B}$ simulates $\mathcal{A}$'s oracle queries as follows:

- **Random Oracle** $\mathsf{H}(id_k)$: On the k^{th} query to this oracle, $\mathcal{A}$ queries on an index id_k. If $\mathcal{Q}_{\mathsf{H}}[id_k] = \perp$, $\mathcal{B}$ queries $\mathcal{O}_0^{\mathrm{GPS}_3}()$ and receives a base element h_k. It then sets $\mathcal{Q}_{\mathsf{H}}[id_k] \leftarrow h_k$ and returns $\mathcal{Q}_{\mathsf{H}}[id_k]$ to $\mathcal{A}$.
- **Signing Oracle** $\mathcal{O}_{\mathsf{Sign}}(id_k, M_{1k}, M_{2k})$: On the k^{th} query to this oracle, $\mathcal{A}$ queries on an indexed DH message $(id_k, M_{k1}, M_{k2}) \in \mathcal{M}_{\mathsf{iDH}}^{\mathsf{H}}$. If $(id_k, \star) \in \mathcal{Q}_{\mathsf{S}}$, $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ looks up $h_k = \mathcal{Q}_{\mathsf{H}}[id_k]$, queries its oracle $\mathcal{O}_1^{\mathrm{GPS}_3}(h_k, M_{1k}, M_{2k})$, and receives $h_k^x M_{1k}^y$. $\mathcal{B}$ updates the set of queried messages $\mathcal{Q}_{\mathsf{S}} = \mathcal{Q}_{\mathsf{S}} \cup \{(id_k, M_{1k}, M_{2k})\}$ and the set of equivalence classes $\mathcal{Q}_{\mathsf{EQ}} \leftarrow \mathcal{Q}_{\mathsf{EQ}} \cup \{\mathsf{EQ}(M_{k1}, M_{k2})\}$ and returns the signature $\sigma_k = (h_k, h_k^x M_{1k}^y)$ to $\mathcal{A}$.

Output. At the end of the game, $\mathcal{A}$ produces a valid forgery $(\tilde{M}^*, \sigma^*) = (M_1^*, M_2^*, (h^*, s^*))$ and $\mathcal{B}$ outputs (M_1^*, M_2^*, h^*, s^*).

$\mathcal{B}$ correctly simulates the EUF-CiMA game. Since $\mathcal{A}$'s forgery satisfies $\tilde{M}^* \notin \mathcal{Q}_{\mathsf{S}}^{\mathsf{EQ}}$ and $\mathsf{Verify}(\mathsf{p}, \mathsf{vk}, \tilde{M}^*, \sigma^*) = 1$, $\mathcal{B}$'s winning conditions are also satisfied and

$$Adv_{\mathsf{IM\text{-}SPS},\mathcal{A}}^{\mathsf{EUF\text{-}CiMA}}(\kappa) = Adv_{\mathcal{B}}^{\mathrm{GPS}_3}(\kappa) \leq \nu(\kappa) \ .$$

□

B.3 Proof of Theorem 3

Proof. **Correctness.** If If $e(h, M_{2j}) = e(h, \hat{g}^{m_j}) = e(h^{m_j}, \hat{g}) = e(M_{1j}, \hat{g})$ for all $j \in [1, \ell]$, then for $\sigma = (h, s) = (h, h^{x+\sum_{j=1}^{\ell} m_j y_j})$ we have $e(h, \mathsf{vk}_0) \prod_{j=1}^{\ell} e(M_{1j}, \mathsf{vk}_j) = e(h, \hat{g}^x) \prod_{j=1}^{\ell} e(h^{m_j}, \hat{g}^{y_j}) = e(h, \hat{g})^{x+\sum_{j=1}^{\ell} m_j y_j} = e(h^{x+\sum_{j=1}^{\ell} m_j y_j}, \hat{g}) = e(h^x \prod_{j=1}^{\ell} M_{1j}^{y_j}, \hat{g}) = e(s, \hat{g})$.

EUF-CiMA **Security.** To prove this theorem, we use a technique for compressing multi-messages into (single) messages from [PS16, Theorem 7].

We wish to show that if there exists a PPT adversary $\mathcal{A}$ that breaks the EUF-CiMA security (Figure 5) of the indexed multi-message SPS scheme IMM-SPS (Figure 9) with non-negligible probability, then we can construct a PPT adversary $\mathcal{A}$' that breaks the GPS$_3$ assumption (Figure 13) with non-negligible probability.

Suppose there exists such a PPT adversary $\mathcal{A}$. Then, we construct a PPT adversary $\mathcal{A}$' as a reduction $\mathcal{B}$ running $\mathcal{A}$ as a subroutine. We construct the reduction $\mathcal{B}$ for breaking the GPS$_3$ assumption as follows.

The reduction $\mathcal{B}$ is responsible for simulating oracle responses for queries to $\mathcal{O}_{\mathsf{Sign}}(\cdot)$ and H. Let $\mathcal{Q}_{\mathsf{H}}$ be the set of H queries and their responses. $\mathcal{B}$ may program the random oracle H. Let $\mathcal{Q}_{\mathsf{S}}$ be the set of $\mathcal{O}_{\mathsf{Sign}}(\cdot)$ queries $(id, \tilde{M})$ and $\mathcal{Q}_{\mathsf{EQ}}$ the set of equivalence classes of messages $\tilde{M}$. $\mathcal{B}$ initializes $\mathcal{Q}_{\mathsf{H}}, \mathcal{Q}_{\mathsf{S}}, \mathcal{Q}_{\mathsf{EQ}}$ to the empty set.

Initialization. $\mathcal{B}$ takes as input the group description $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, g, \hat{g}) \leftarrow \mathcal{BG}(1^\kappa)$ and a GPS$_3$ challenge $(\hat{g}^x, \hat{g}^y)$. As in game $\mathbf{G}_{\mathcal{B}}^{\mathrm{GPS}_3}(1^\kappa)$, $\mathcal{B}$ has access to oracles $\mathcal{O}_0^{\mathrm{GPS}_3}()$ and $\mathcal{O}_1^{\mathrm{GPS}_3}(\cdot)$. For all $j \in [1, \ell]$, $\mathcal{B}$ samples $\alpha_j, \beta_j \leftarrow\$\, \mathbb{Z}_p$ and sets $\mathsf{vk}_j \leftarrow \hat{g}^{y_j} = (\hat{g}^y)^{\alpha_j} \hat{g}^{\beta_j}$. $\mathcal{B}$ sets the IMM-SPS verification key $\vec{\mathsf{vk}} \leftarrow (\mathsf{vk}_0, \mathsf{vk}_1, \ldots, \mathsf{vk}_\ell) = (\hat{g}^x, \hat{g}^{y_1}, \ldots, \hat{g}^{y_\ell})$ and runs $\mathcal{A}(\mathsf{pp}, \mathsf{vk})$.

Simulating Oracle Queries. $\mathcal{B}$ simulates $\mathcal{A}$'s oracle queries as follows:

- **Random Oracle** $\mathsf{H}(id_k)$: On the k^{th} query to this oracle, $\mathcal{A}$ queries on an index id_k. If $\mathcal{Q}_{\mathsf{H}}[id_k] = \perp$, $\mathcal{B}$ queries $\mathcal{O}_0^{\mathrm{GPS}_3}()$ and receives a base element h_k. It then sets $\mathcal{Q}_{\mathsf{H}}[id_k] \leftarrow h_k$ and returns $\mathcal{Q}_{\mathsf{H}}[id_k]$ to $\mathcal{A}$.
- **Signing Oracle** $\mathcal{O}_{\mathsf{Sign}}(id_k, \vec{M}_{k1}, \vec{M}_{k2})$: On the k^{th} query to this oracle, $\mathcal{A}$ queries on an indexed DH message $(id_k, \vec{M}_{k1}, \vec{M}_{k2}) \in \mathcal{M}_{\mathsf{iDH}}^{\mathsf{H}}$. If $(id_k, \star) \in \mathcal{Q}_{\mathsf{S}}$, $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ looks up $h_k = \mathcal{Q}_{\mathsf{H}}[id_k]$, computes $M'_{k1} = \prod_{j=1}^{\ell} M_{k1j}^{\alpha_j}$ and $M'_{k2} = \prod_{j=1}^{\ell} M_{k2j}^{\alpha_j}$, queries $\mathcal{O}_1^{\mathrm{GPS}_3}(h_k, M'_{k1}, M'_{k2})$, and receives $s'_k = h_k^x (M'_{k1})^y$.

$\mathcal{B}$ computes $\sigma_k = (h_k, s_k) = \left(h_k, s_k^{'} \prod_{j=1}^{\ell} M_{k1j}^{\beta_j}\right)$, which is a valid signature on $(id_k, \vec{M}_{k1}, \vec{M}_{k2})$. Indeed,

$$\begin{aligned} e(s_k, \hat{g}) &= e\left(s_k^{'} \prod_{j=1}^{\ell} M_{k1j}^{\beta_j}, \hat{g}\right) = e\left(s_k^{'}, \hat{g}\right) \cdot e\left(\prod_{j=1}^{\ell} M_{k1j}^{\beta_j}, \hat{g}\right) \\ &= e\left(h_k^x \left(\prod_{j=1}^{\ell} M_{k1j}^{\alpha_j}\right)^y, \hat{g}\right) \cdot e\left(\prod_{j=1}^{\ell} M_{k1j}^{\beta_j}, \hat{g}\right) \\ &= e\left(h_k^x, \hat{g}\right) \cdot e\left(\prod_{j=1}^{\ell} M_{k1j}^{\alpha_j y + \beta_j}, \hat{g}\right) = e\left(h_k, \mathsf{vk}_0\right) \cdot \prod_{j=1}^{\ell} e\left(M_{k1j}, \mathsf{vk}_j\right) \ . \end{aligned}$$

$\mathcal{B}$ updates the set of queried messages $\mathcal{Q}_{\mathsf{S}} \leftarrow \mathcal{Q}_{\mathsf{S}} \cup \left\{(id_k, \vec{M}_{k1}, \vec{M}_{k2})\right\}$ and the set of equivalence classes $\mathcal{Q}_{\mathsf{EQ}} \leftarrow \mathcal{Q}_{\mathsf{EQ}} \cup \left\{\mathsf{EQ}(\vec{M}_{k1}, \vec{M}_{k2})\right\}$ and returns σ_k to $\mathcal{A}$.

Output. At the end of the game, $\mathcal{A}$ returns a valid forged signature $\sigma^* = (h^*, s^*)$ on $(\vec{M}_1^*, \vec{M}_2^*)$ satisfying $(\vec{M}_1^*, \vec{M}_2^*) \notin \mathcal{Q}_{\mathsf{EQ}}$, $h^* \neq 1_{\mathbb{G}_1}$, $e(h^*, \mathsf{vk}_0) \prod_{j=1}^{\ell} e(M_{1j}^*, \mathsf{vk}_j) = e(s^*, \hat{g})$, and for all $j \in [1, \ell]$, $M_{1j}^* \neq 1_{\mathbb{G}_1}$ and $e(h^*, M_{2j}^*) = e(M_{1j}^*, \hat{g})$. If there exists a queried message $(\vec{M}_{i1}, \vec{M}_{i2}) \in \mathcal{Q}_{\mathsf{EQ}}$ such that $\prod_{j=1}^{\ell} (M_{2j}^*)^{\alpha_j} = \prod_{j=1}^{\ell} (M_{i2j}^*)^{\alpha_j}$, then $\mathcal{B}$ aborts. Else, $\mathcal{B}$ returns $(M_1^{*'}, M_2^{*'}, h^*, s^{*'})$, where $s^{*'} = s^* \prod_{j=1}^{\ell} \left(M_{1j}^*\right)^{-\beta_j}$ and $(M_1^{*'}, M_2^{*'}) = \left(\prod_{j=1}^{\ell} \left(M_{1j}^*\right)^{\alpha_j}, \prod_{j=1}^{\ell} \left(M_{2j}^*\right)^{\alpha_j}\right)$.

$\mathcal{B}$ correctly simulates the $\mathsf{EUF\text{-}CiMA}$ game. Because $\mathcal{A}$'s forgery satisfies $e(s^*, \hat{g}) = e(h^*, \mathsf{vk}_0) \prod_{j=1}^{\ell} e(M_{1j}^*, \mathsf{vk}_j)$, $\mathcal{B}$'s output satisfies:

$$\begin{aligned} e(s^{*'}, \hat{g}) &= e\left(s^* \prod_{j=1}^{\ell} \left(M_{1j}^*\right)^{-\beta_j}, \hat{g}\right) = e\left(s^*, \hat{g}\right) \cdot e\left(\prod_{j=1}^{\ell} (M_{1j}^*)^{-\beta_j}, \hat{g}\right) \\ &= e\left(h^*, \hat{g}^x\right) \prod_{j=1}^{\ell} e(M_{1j}^*, \hat{g}^{\alpha_j y + \beta_j}) \cdot e\left(\prod_{j=1}^{\ell} (M_{1j}^*)^{-\beta_j}, \hat{g}\right) \\ &= e\left(h^*, \hat{g}^x\right) e\left(\prod_{j=1}^{\ell} (M_{1j}^*)^{\alpha_j y + \beta_j}, \hat{g}\right) \cdot e\left(\prod_{j=1}^{\ell} (M_{1j}^*)^{-\beta_j}, \hat{g}\right) \\ &= e\left(h^*, \hat{g}^x\right) e\left(\prod_{j=1}^{\ell} (M_{1j}^*)^{\alpha_j y}, \hat{g}\right) = e\left(h^*, \hat{g}^x\right) \cdot e\left(M_1', \hat{g}^y\right) \ . \end{aligned}$$

$\mathcal{B}$ fails to provide a valid output if there exists a queried message $(\vec{M}_{i1}, \vec{M}_{i2}) \in \mathcal{Q}_{\mathsf{EQ}}$ such that $\prod_{j=1}^{\ell} (M_{2j}^*)^{\alpha_j} = \prod_{j=1}^{\ell} (M_{i2j})^{\alpha_j}$. Thus, we must demonstrate that the probability of this event, denoted by E, is negligible, i.e., $\Pr[E] \leq \nu(\kappa)$.

For the given instance $(\hat{g}^x, \hat{g}^y)$, suppose the reduction $\mathcal{B}$ instead initializes the verification keys for $\mathsf{IMM\text{-}SPS}$ by sampling $\mu_j \leftarrow\!\$\, \mathbb{Z}_p$ for all $j \in [1, \ell]$ and setting $\hat{g}^{\alpha'_j} \leftarrow \hat{g}^{\alpha_j - \mu_j}$ and $\hat{g}^{\beta'_j} \leftarrow \hat{g}^{\beta_j}(\hat{g}^y)^{\mu_j}$. Then,

$$\hat{g}^{\alpha'_j y}\hat{g}^{\beta'_j} = \hat{g}^{y(\alpha_j - \mu_j)}\hat{g}^{\beta_j + y\mu_j} = \hat{g}^{y\alpha_j}\hat{g}^{-y\mu_j}\hat{g}^{\beta_j}\hat{g}^{y\mu_j} = \hat{g}^{\alpha_j y + \beta_j} = \mathsf{vk}_j \ .$$

Therefore, the issued verification keys are independent of the μ_j's and do not disclose any information about the α_j's. Moreover, due to the fact that the queried signatures are also based on the random oracle and public keys, the view of adversary is completely independent of the α_j's and we can write $\Pr[E] \leq q_S/p$, where q_S is the number of queries to the signing oracle made by the adversary $\mathcal{A}$. Thus,

$$Adv^{\mathsf{EUF\text{-}CiMA}}_{\mathsf{IMM\text{-}SPS},\mathcal{A}}(\kappa) \leq Adv^{\mathrm{GPS}_3}_{\mathcal{B}}(\kappa) + q_S/p \ .$$

$\square$