Partially Structure-Preserving Signatures: Lower Bounds, Constructions and More

Essam Ghadafi(✉)

University of the West of England, Bristol, UK

Abstract. In this work we first provide a framework for defining a large subset of pairing-based digital signature schemes which we call Partially Structure-Preserving Signature (PSPS) schemes. PSPS schemes are similar in nature to structure-preserving signatures with the exception that in PSPS schemes messages are scalars from $\mathbb{Z}_p$ instead of being group elements. This class encompasses various existing schemes which have a number of desirable features which makes them an ideal building block for many privacy-preserving cryptographic protocols. Such schemes include the widely-used schemes of Camenisch-Lysyanskaya (CRYPTO 2004) and Pointcheval-Sanders (CT-RSA 2016). We then provide various impossibility and lower bound results for variants of this class. Our results include bounds for the signature and verification key sizes as well as lower bounds for achieving strong unforgeability. We also give a generic framework for transforming variants of PSPS schemes into structure-preserving ones. As part of our contribution, we also give a number of optimal PSPS schemes which may be of independent interest. Our results aid in understanding the efficiency of pairing-based signature schemes and show a connection between this class of schemes and structure-preserving ones.

Keywords: Digital signatures · Bilinear groups · Lower bounds · Structure-preserving

1 Introduction

Digital signatures are a fundamental cryptographic primitive which besides being useful in their own right, they are used as an essential building block for various more complex protocols.

The emergence of pairing-based cryptography has been associated with the introduction of many pairing-based digital signature schemes. One of the extensively used pairing-based signature schemes is that of Camenisch and Lysyanskaya (CL) [16]. The scheme has a number of desirable features which makes it an ideal building block for various privacy-preserving protocols, including group signatures, e.g. [10,16], anonymous credentials, e.g. [16], and direct anonymous attestation, e.g. [20]. Notably, the scheme besides having fully and perfectly randomizable signatures, it is compatible with Pedersen-like commitment schemes

K. Sako and N. O. Tippenhauer (Eds.): ACNS 2021, LNCS 12726, pp. 284–312, 2021.
https://doi.org/10.1007/978-3-030-78372-3_11

[41] and thus it is possible to sign committed messages. A recent improvement to the CL scheme is the Pointcheval and Sanders (PS) scheme [42], which besides enjoying better efficiency and preserving all of its desirable features, it yields constant-size signatures regardless of the size of the message. Despite its relatively young age, the PS scheme has been used in the construction of various protocols. A common feature to the structure of both aforementioned schemes is that the signer is generic, and when viewing the signature components as an exponentiation of the respective group generator to a fraction of polynomials, the denominator polynomials are independent of the message. This is to the contrary of other pairing-based schemes, e.g. [11,12,43,45], which even though are based on non-interactive intractability assumptions, they do not enjoy some of the desirable features of the CL and PS schemes, e.g. the randomizability of the signatures, having a generic signer, the ease of being combined with Pedersen-like commitments, and a short verification key.

The dual-form signature framework [27] was used by [18,27] to obtain (less efficient) variants of some existing schemes, e.g. CL and PS schemes, whose security relies on static intractability assumptions.

Structure-Preserrving Signature (SPS) schemes [4] are also pairing-based signature schemes with the extra requirement that the messages, the verification key and the signatures consist of only source group elements. Verification of signatures in those schemes only involves evaluating Pairing-Product Equations (PPEs) and checking group memberships. Such properties make them compatible with widely-used constructs such as ElGamal encryption [21] and Groth-Sahai proofs [35] and hence they render themselves as a tool for designing cryptographic protocols which dispense with relying on random oracles [22] despite the efficiency degradation. SPS schemes have numerous applications, including group signatures, e.g. [4,38], blind signatures, e.g. [4,24], and anonymous credentials, e.g. [15,23].

A numerous number of SPS schemes have been proposed in the 3 different bilinear groups settings. In the most efficient bilinear group setting, i.e. the Type-3 setting (cf. Sect. 2), existing schemes include [4,5,7,19,29,31,34]. Abe et al. [5] proved that a Type-3 signature must contain at least 3 bilateral elements and require at least 2 PPEs for verification. Optimal SPS schemes rely on security proofs in the generic group model [40,44]. Abe et al. [6] proved that the unforgeability of an optimal Type-3 scheme cannot be based on a non-interactive intractability assumption. Ghadafi [31] showed that by restricting the message space to the set of Diffie-Hellman (DH) pairs (cf. Sect. 2) it is possible to circumvent the lower bound and obtain optimal unilateral signatures consisting of 2 elements. Such variants provide some efficiency gains for some protocols, including direct anonymous attestation [13] and attribute-based signatures [39]. Other constructions for this message space include, e.g. [4,28,29,32,33].

Constructions of SPS schemes relying on non-interactive assumptions include [1–3,8,14,26,36–38]. Chase and Kohlweiss [17] gave a transformation which utilizes pairwise-independent hash functions and the Groth-Sahai proof system [35] to obtain structure-preserving signatures based on standard assumptions from some pairing-based signature schemes for scalar messages. Their transformation is rather costly as it yields signatures consisting of tens of group elements.

Motivation and Our Contribution. While structure-preserving signatures and their efficiency are well studied, other types of pairing-based signature schemes still have some open problems pertaining to their feasibility and bounds for their efficiency are still lacking. For instance, it is not currently known whether efficient strongly unforgeable generic-signer schemes with a similar structure to the CL and PS schemes are possible. Moreover, it is not currently known whether the recent efficient PS scheme is optimal or whether it is possible to improve efficiency while preserving all of its desirable features.

SPS schemes might be less desirable than pairing-based schemes for scalar messages for some applications due to the loss in efficiency. This is particularly the case for applications where relying on random oracles is tolerated, applications requiring a stand-alone signature scheme, or applications not requiring proof systems to hide the message.

Towards a better understanding of the efficiency of pairing-based signature schemes for scalar messages, we first define a framework for capturing a large class of such schemes which we refer to as Partially Structure-Preserving Signature (PSPS) schemes[1]. Other than the messages being scalars from $\mathbb{Z}_p$ rather than source group elements, PSPS schemes have similar properties to structure-preserving signatures, including having a generic signer and signatures and verification keys consisting solely of source group elements. We provide different variants of our definition. More precisely, we define Strongly Partially Structure-Preserving (SPSPS) schemes and Linear-Message Strongly Partially Structure-Preserving (LmSPSPS) schemes. The former requires that the PSPS scheme does not involve the message in the denominator of any of the signature components whereas the latter additionally requires that the message is embedded in the signature components in a linear manner. The CL and PS schemes for example fall into the LmSPSPS class.

We provide various lower bounds and impossibility results for LmSPSPS schemes. More precisely, we prove that existentially unforgeable under random-message attacks (EUF-RMA) schemes must have at least 2 elements in the signature and that strongly existentially unforgeable under chosen-message attacks (sEUF-CMA) schemes must have bilateral signatures consisting of at least 3 elements. Also, we prove that optimal schemes, including one-time schemes, cannot have a verification key consisting of fewer than 2 elements. In essence, this proves that the PS scheme and our new LmSPSPS scheme are optimal in every respect. In Table 1 we summarize our lower bound results for the size of the signature and compare them to those for structure-preserving signatures.

We also construct optimal one-time sEUF-CMA LmSPSPS schemes with one-element signatures and a new optimal EUF-CMA LmSPSPS scheme for a vector of messages. We prove the security of the latter using a new interactive intractability assumption which we show holds in the generic group model. The

[1] We remark that such a term was used informally in [31] to refer to SPS schemes where some message components are allowed to be scalar messages.

Table 1. Summary of the lower bounds for $|\sigma|$ in the Type-3 bilinear setting. B stands for bilateral elements whereas U stands for unilateral elements.

Notion	# sign queries	EUF-RMA	sEUF-CMA
LmSPSPS (this work)	1	1	1
	>1	2U	3B
SPS for unilateral messages [5,7]	1	1	1
	>1	3B	3B
SPS for DH pairs [32,33]	1	1	1
	>1	2U	3B

efficiency of our scheme matches that of the PS scheme [42] whose security also relies on an interactive assumption in every respect.

Finally, we show a connection between LmSPSPS schemes and SPS schemes by showing that if a LmSPSPS scheme satisfies an extra requirement which is that the signature and verification key components in either source group are disjoint, which for instance is satisfied by the CL and PS schemes as well as our new scheme, such a scheme automatically yields an analogues SPS scheme where the message space is the set of Diffie-Hellman pairs. The obtained SPS scheme has the same key pair as the original LmSPSPS scheme and is unforgeable in the generic group model. We also show some instantiations of our framework.

Besides being a step closer towards a better understanding of the efficiency of pairing-based signature schemes, our results uncover a link between LmSPSPS and SPS schemes.

Paper Organization. Some preliminary definitions are in Sect. 2. In Sect. 3 we define PSPS schemes. In Sects. 4 and 5 we present our LmSPSPS constructions. In Sect. 6 we give our transformation from LmSPSPS to SPS schemes and provide example instantiations. Finally, in Sect. 7 we give our feasibility results.

Notation. We write $y = A(x; r)$ when algorithm A on input x and randomness r outputs y. We write $y \leftarrow A(x)$ for the process of setting $y = A(x; r)$ where r is sampled at random. We also write $y \leftarrow S$ for sampling y uniformly at random from a set S. A function $\nu(.) : \mathbb{N} \rightarrow \mathbb{R}^+$ is negligible (in n) if for every polynomial $p(.)$ and all sufficiently large values of n, it holds that $\nu(n) < \frac{1}{p(n)}$. By PPT we mean running in probabilistic polynomial time in the relevant security parameter. We use $[k]$ to denote the set $\{1, \ldots, k\}$ and $[i, k]$ to denote the set $\{i, i+1, \ldots, k\}$. For vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}_p^n$ we denote by $\boldsymbol{x}^{\boldsymbol{y}}$ the operation $\prod_{i=1}^{n} x_i^{y_i}$.

2 Preliminaries

In this section we provide some preliminary definitions.

2.1 Bilinear Groups

A bilinear group is a tuple $\mathcal{P} := (\mathbb{G}, \mathbb{H}, \mathbb{T}, p, G, \tilde{H}, e)$ where $\mathbb{G}$, $\mathbb{H}$ and $\mathbb{T}$ are groups of a prime order p, and G and $\tilde{H}$ generate $\mathbb{G}$ and $\mathbb{H}$, respectively. The function e is a non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{H} \longrightarrow \mathbb{T}$. We refer to $\mathbb{G}$ and $\mathbb{H}$ as the source groups whereas we refer to $\mathbb{T}$ as the target group. We will use multiplicative notation for all the groups. To distinguish elements of $\mathbb{H}$ from those of $\mathbb{G}$ we will accent the former with $\tilde{}$. We let $\mathbb{G}^\times := \mathbb{G} \backslash \{1_\mathbb{G}\}$ and $\mathbb{H}^\times := \mathbb{H} \backslash \{1_\mathbb{H}\}$. We limit our attention to the efficient Type-3 setting [25], where $\mathbb{G} \neq \mathbb{H}$ and there is no efficiently computable homomorphism between the source groups in either direction. We assume an algorithm $\mathcal{BG}$ that on input 1^κ, for some security parameter $\kappa \in \mathbb{N}$, outputs a description of a bilinear groups $\mathcal{P}$.

We call a pair $(M, \tilde{N}) \in \mathbb{G} \times \mathbb{H}$ a Diffie-Hellman (DH) pair [4] if it satisfies $e(M, \tilde{H}) = e(G, \tilde{N})$. We denote the set of DH pairs by $\mathcal{DH}$.

2.2 Digital Signatures

A digital signature scheme $\mathcal{DS}$ over a bilinear group $\mathcal{P}$ generated by $\mathcal{BG}$ for a message space $\mathcal{M}$ consists of the following algorithms:

$\mathsf{KeyGen}(\mathcal{P})$ on input $\mathcal{P}$, it outputs a pair of secret/verification keys $(\mathsf{sk}, \mathsf{vk})$.
$\mathsf{Sign}(\mathsf{sk}, m)$ on input sk and a message $m \in \mathcal{M}$, it outputs a signature σ.
$\mathsf{Verify}(\mathsf{vk}, m, \sigma)$ outputs 1 if σ is a valid signature on m w.r.t. vk and 0 otherwise.

Definition 1 (Correctness). *A digital signature scheme $\mathcal{DS}$ over a bilinear group generator $\mathcal{BG}$ is* (perfectly) correct *if for all $\kappa \in \mathbb{N}$*

$$\Pr\left[\begin{array}{l} \mathcal{P} \leftarrow \mathcal{BG}(1^\kappa) \\ (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(\mathcal{P}) \\ m \leftarrow \mathcal{M} \\ \sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m) \end{array} : \mathsf{Verify}(\mathsf{vk}, m, \sigma) = 1 \right] = 1 \cdot$$

Besides the correctness requirement, we require existential unforgeability.

Definition 2 (Existential Unforgeability). *A digital signature scheme $\mathcal{DS}$ over a bilinear group generator $\mathcal{BG}$ is* Existentially-Unforgeable against adaptive Chosen-Message Attack (EUF-CMA) *if for all $\kappa \in \mathbb{N}$ for all PPT adversaries $\mathcal{A}$, the following is negligible (in κ)*

$$\Pr\left[\begin{array}{l} \mathcal{P} \leftarrow \mathcal{BG}(1^\kappa) \\ (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(\mathcal{P}) \\ (\sigma^*, m^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathcal{P}, \mathsf{vk}) \end{array} : \mathsf{Verify}(\mathsf{vk}, m^*, \sigma^*) = 1 \wedge m^* \notin Q_{\mathsf{Sign}} \right],$$

where Q_{Sign} is the set of messages queried to Sign.

Strong Existential Unforgeability against adaptive Chosen-Message Attack (sEUF-CMA) requires that the adversary cannot even output a new signature on a message that was queried to the sign oracle.

A weaker variant of EUF-CMA is *Existential Unforgeability against a Random-Message Attack (EUF-RMA)* in which the sign oracle samples a message uniformly from the message space and returns the message and a signature on it. In one-time signatures, the adversary is restricted to a single signing query.

Sometimes it is desirable that signatures are publicly re-randomizable where there is an algorithm $\mathsf{Randomize}$ that on input (vk, m, σ) outputs a new signature σ' on m which is indistinguishable from a fresh signature on the same message.

Structure-Preserving Signatures. Structure-preserving signatures [4] are signature schemes defined over bilinear groups where the messages, the verification key and signatures are all group elements from either or both source groups, and verifying signatures only involves deciding group membership of the signature components and evaluating pairing-product equations (PPEs) of the form of Eq. (1).

$$\prod_i \prod_j e(A_i, \tilde{B}_j)^{c_{i,j}} = 1_{\mathbb{T}}, \tag{1}$$

where $A_i \in \mathbb{G}$ and $\tilde{B}_j \in \mathbb{H}$ are group elements appearing in $\mathcal{P}, m, \mathsf{vk}, \sigma$, whereas $c_{i,j} \in \mathbb{Z}_p$ are constants.

Generic Signer. We refer to a signer that can only decide group membership, evaluate the bilinear map e, compute the group operations in groups $\mathbb{G}, \mathbb{H}$ and $\mathbb{T}$, and compare group elements as a *generic signer*.

3 Partially Structure-Preserving Signatures

In this section we define a class of prime-order pairing-based digital signature schemes which we call Partially Structure-Preserving Signature (PSPS) schemes. Informally, a PSPS scheme is a pairing-based signature scheme for scalar messages from $\mathbb{Z}_p^n$ for $n \geq 1$ where the signature components and verification key contain only source group elements and the signature components are computed by raising source group elements to fraction of polynomials involving the secret key, the messages and the randomness chosen as part of the signing process. We then define 2 variants of PSPS schemes to capture most of the practical schemes existing in the literature. First, we define Strongly Partially Structure-Preserving Signature (SPSPS) schemes which additionally require that the denominator polynomials used in computing the signature components are independent of the messages to be signed. Then we define a variant of SPSPS which we refer to as Linear-Message Strongly Partially Structure-Preserving Signature (LmSPSPS) schemes which additionally requires that the numerator polynomials are linear in the message to be signed. The latter captures a large class of existing schemes for scalar messages, including variants of the CL and PS schemes.

Definition 3 (Partially Structure-Preserving Signatures). *A digital signature scheme $\mathcal{DS}$ over a bilinear group generator $\mathcal{BG}$ is Partially Structure-Preserving Signature (PSPS) scheme if it satisfies all the following:*

- *$\mathcal{BG}(1^\kappa)$ generates a bilinear group description $\mathcal{P} := (\mathbb{G}, \mathbb{H}, \mathbb{T}, p, G, \tilde{H}, e)$.*
- *The verification key* vk *consists of $\mathcal{P}$ and source group elements $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathbb{G}^{\mu} \times \mathbb{H}^{\mu'}$.*
- *The message space is $\mathcal{M} := \mathbb{Z}_p^n$ for some $n \geq 1$.*
- *A signature on a message $\boldsymbol{m} \in \mathcal{M}$ is of the form $\sigma := (\boldsymbol{S}, \tilde{\boldsymbol{T}}) \in \mathbb{G}^{\nu} \times \mathbb{H}^{\nu'}$ which is computed by a generic signer by sampling a vector $\boldsymbol{r} \in \mathbb{Z}_p^{n'}$ (independently of the message $\boldsymbol{m}$) and computing $S_i := G^{\frac{\alpha_i(\mathsf{sk}, \boldsymbol{m}, \boldsymbol{r})}{\alpha'_i(\mathsf{sk}, \boldsymbol{m}, \boldsymbol{r})}}$ and $\tilde{T}_j := \tilde{H}^{\frac{\beta_j(\mathsf{sk}, \boldsymbol{m}, \boldsymbol{r})}{\beta'_j(\mathsf{sk}, \boldsymbol{m}, \boldsymbol{r})}}$ for some formal multivariate polynomials $\alpha_i, \alpha'_i, \beta_j, \beta'_j \in \mathbb{F}_p[X_1, \ldots, X_{\mu}, Y_1, \ldots, Y_{\mu'}, M_1, \ldots, M_n, R_1, \ldots, R_{n'}]$ of total degree bounded by $d(\kappa)$.*
- *Signature verification involves deciding group membership*[2] *and evaluating a set of pairing-product equations of the following form:*

$$\prod_{i=1}^{\nu} e(S_i, \prod_{j=1}^{\mu'} \tilde{Y}_j)^{\rho_{1,i,j}(\boldsymbol{m})} \prod_{i=1}^{\nu'} e(\prod_{j=1}^{\mu} X_j, \tilde{T}_i)^{\rho_{2,i,j}(\boldsymbol{m})}$$
$$\prod_{i=1}^{\nu} e(S_i, \prod_{j=1}^{\nu'} \tilde{T}_j)^{\rho_{3,i,j}(\boldsymbol{m})} \prod_{i=1}^{\mu} \prod_{j=1}^{\mu'} e(X_i, \tilde{Y}_j)^{\rho_{4,i,j}(\boldsymbol{m})} = Z_\ell, \quad (2)$$

where $\rho_{i,j,k} \in \mathbb{F}_p[M_1, \ldots, M_n]$ are multivariate polynomials of total degree bounded by $d'(\kappa)$ whereas $Z_\ell \in \mathbb{T}$ is a public constant. In the strict sense, one can necessitate that $Z_\ell = 1_{\mathbb{T}}$.

Definition 4 (Strongly Partially Structure-Preserving Signatures). *A digital signature scheme $\mathcal{DS}$ over a bilinear group generator $\mathcal{BG}$ is Strongly Partially Structure-Preserving Signature (SPSPS) if it is partially structure-preserving and it holds that for all $i \in [\nu]$ and for all $j \in [\nu']$, the polynomials α'_i and β'_j are independent of the message.*

Definition 5 (Linear-Message Strongly Partially Structure-Preserving Signatures). *A digital signature scheme $\mathcal{DS}$ over a bilinear group generator $\mathcal{BG}$ is Linear-Message Strongly Partially Structure-Preserving Signature (LmSPSPS) if it is strongly partially structure-preserving and it holds that for all $i \in [\nu]$ and for all $j \in [\nu']$, α_i and β_j are linear in $\boldsymbol{M}$, i.e. for all $k \in [n]$, for all $i \in [\nu]$, for all $j \in [\nu']$, the degree of M_k in α_i and β_j is either* 0 *or* 1 *and for all $\eta, \eta' \in [n]$ neither of the polynomials contain the monomial $M_\eta M_{\eta'}$.*

We now define a subset of PSPS schemes which we call Disjoint Partially Structure-Preserving Signature (DPSPS) schemes. Informally, a DPSPS scheme is a PSPS scheme where the spans of the sets of fraction of formal polynomials corresponding to the verification key and signature components in the source groups are disjoint.

[2] For more generality, we allow membership checks of the forms $S_i \in \mathbb{G}^{\times}$ and $\tilde{T}_j \in \mathbb{H}^{\times}$.

Definition 6 (Disjoint Partially Structure-Preserving Signatures). *Let* $\frac{\gamma_{1,i}(\mathsf{SK})}{\gamma'_{1,i}(\mathsf{SK})}$ *for* $i \in [\mu]$ *and* $\frac{\gamma_{2,j}(\mathsf{SK})}{\gamma'_{2,j}(\mathsf{SK})}$ *for* $j \in [\mu']$ *be the fraction of formal polynomials used to compute the verification key* $\boldsymbol{X} \in \mathbb{G}^{\mu}$ *and* $\boldsymbol{Y} \in \mathbb{H}^{\mu'}$ *(excluding the default source group generators), respectively. We say a signature scheme* $\mathcal{DS}$ *over a bilinear group generator* $\mathcal{BG}$ *is a Disjoint Partially Structure-Preserving Signature (DPSPS) scheme if it is partially structure-preserving and additionally meets the following requirement:*

$$\mathit{Span}\left(\left\{\frac{\gamma_{1,1}(\mathsf{SK})}{\gamma'_{1,1}(\mathsf{SK})},\ldots,\frac{\gamma_{1,\mu}(\mathsf{SK})}{\gamma'_{1,\mu}(\mathsf{SK})},\frac{\alpha_1(\mathsf{SK},\boldsymbol{M},\boldsymbol{R})}{\alpha'_1(\mathsf{SK},\boldsymbol{M},\boldsymbol{R})},\ldots,\frac{\alpha_\nu(\mathsf{SK},\boldsymbol{M},\boldsymbol{R})}{\alpha'_\nu(\mathsf{SK},\boldsymbol{M},\boldsymbol{R})}\right\}\right)$$
$$\cap\ \mathit{Span}\left(\left\{\frac{\gamma_{2,1}(\mathsf{SK})}{\gamma'_{2,i}(\mathsf{SK})},\ldots,\frac{\gamma_{2,\mu'}(\mathsf{SK})}{\gamma'_{2,\mu'}(\mathsf{SK})},\frac{\beta_1(\mathsf{SK},\boldsymbol{M},\boldsymbol{R})}{\beta'_1(\mathsf{SK},\boldsymbol{M},\boldsymbol{R})},\ldots,\frac{\beta_{\nu'}(\mathsf{SK},\boldsymbol{M},\boldsymbol{R})}{\beta'_{\nu'}(\mathsf{SK},\boldsymbol{M},\boldsymbol{R})}\right\}\right)=\{0\}\cdot$$

We call a LmSPSPS scheme a Disjoint LmSPSPS (DLmSPSPS) scheme if it satisfies the above disjointness requirement. Examples of schemes conforming to this requirement include the PS scheme and our new scheme.

We later show that DLmSPSPS schemes yield equivalent structure-preserving signature schemes for DH pairs. In our transformation, the disjointness requirement ensures that a generic adversary against the SPS scheme cannot feed elements obtained from previous queries to the sign oracle back into the sign oracle since they do not have a matching component in the opposite source group, i.e. they do not form DH pairs. This restricts the messages the SPS adversary can query back into her sign oracle to being constant polynomials, i.e. scalars from $\mathbb{Z}_p$, similarly to the generic adversary against the underlying DLmSPSPS scheme.

4 A New Optimal LmSPSPS Scheme

Here we give a new LmSPSPS scheme for signing a vector $\boldsymbol{m} \in \mathbb{Z}_p^n$. The idea of the new scheme is based on the signature scheme underlying the blind signature scheme in [30]. The efficiency of our scheme matches that of the PS scheme in every respect.

Given the description of a Type-3 bilinear group $\mathcal{P}$ output by $\mathcal{BG}(1^\kappa)$, the scheme is as follows:

- $\mathsf{KeyGen}(\mathcal{P})$: Select $x, y_1, \ldots, y_{n-1}, z \leftarrow \mathbb{Z}_p^\times$. Set $\tilde{X} := \tilde{H}^x$, $\tilde{Y}_i := \tilde{H}^{y_i}$ for all $i \in [n-1]$ and $\tilde{Z} := \tilde{H}^z$. Set $\mathsf{sk} := (x, y_1, \ldots, y_{n-1}, z)$ and $\mathsf{vk} := (\tilde{X}, \tilde{Y}_1, \ldots, \tilde{Y}_{n-1}, \tilde{Z}) \in \mathbb{H}^{n+1}$.
- $\mathsf{Sign}(\mathsf{sk}, \boldsymbol{m})$: Select $r \leftarrow \mathbb{Z}_p^\times$ and set $(S_1, S_2) := \left(G^r, G^{\frac{r(x+m_1+\sum_{i=2}^n m_i y_{i-1})}{z}}\right)$.

  The signature is $\sigma := (S_1, S_2) \in \mathbb{G}^{\times 2}$.
- $\mathsf{Verify}(\mathsf{vk}, \boldsymbol{m}, \sigma)$: Return 1 if $S_1 \neq 1_{\mathbb{G}}$ and $e(S_2, \tilde{Z}) = e(S_1, \tilde{X}\tilde{H}^{m_1} \prod_{i=2}^{n} \tilde{Y}_{i-1}^{m_i})$ and 0 otherwise.
- $\mathsf{Randomize}(\mathsf{vk}, \boldsymbol{m}, \sigma)$: Select $r' \leftarrow \mathbb{Z}_p^\times$ and return $\sigma' := \sigma^{r'}$.

4.1 Security of the Scheme

Correctness of the scheme is straightforward and easy to verify. Also, it is easy to verify that the scheme conforms to the requirements of a DLmSPSPS scheme. We now define the following new interactive intractability assumption to which we reduce the unforgeability of the scheme.

Definition 7 *(New PSPS (NPSPS) Assumption). Let $\mathcal{P} = (\mathbb{G}, \mathbb{H}, \mathbb{T}, p, G, \tilde{H}, e)$ be the description of a Type-3 bilinear group generated by $\mathcal{BG}(1^\kappa)$. Let $\tilde{X} := \tilde{H}^x$ and $\tilde{Y} := \tilde{H}^y$ for some $x, y \leftarrow \mathbb{Z}_p^\times$. Let $\widehat{\mathcal{O}}_{\tilde{X},\tilde{Y}}(\cdot)$ be an oracle that when queried on $m \in \mathbb{Z}_p$, selects $r \leftarrow \mathbb{Z}_p^\times$ and returns the pair $(G^r, G^{\frac{r(x+m)}{y}}) \in \mathbb{G}^2$. The NPSPS assumption holds (relative to $\mathcal{BG}$) if for all PPT adversaries $\mathcal{A}$ given $(\mathcal{P}, \tilde{X}, \tilde{Y})$ and unlimited access to $\widehat{\mathcal{O}}_{\tilde{X},\tilde{Y}}(\cdot)$, the probability that $\mathcal{A}$ outputs a new pair $(R^*, R^{*\frac{(x+m^*)}{y}}) \in {\mathbb{G}^\times}^2$ for some $m^* \in \mathbb{Z}_p$ which was not queried to $\widehat{\mathcal{O}}_{\tilde{X},\tilde{Y}}(\cdot)$ is negligible (in κ).*

The following theorem proves that the NPSPS assumption holds in the generic group model.

Theorem 1. *For a generic adversary $\mathcal{A}$ which makes q_G group operation queries, q_P pairing queries and q_O queries to the $\widehat{\mathcal{O}}_{\tilde{X},\tilde{Y}}$ oracle, the probability that $\mathcal{A}$ breaks the NPSPS assumption is $\mathcal{O}(\frac{q_G^2+q_P^2+q_O^2}{p})$ where p if the prime order of the bilinear group.*

Proof. Let q_O be the number of queries to the $\widehat{\mathcal{O}}_{\tilde{X},\tilde{Y}}$ oracle, q_G be the number of group operation queries and q_P be the number of pairing queries the adversary makes in her game. We first prove that no linear combinations of the formal Laurent polynomials in $\mathbb{Z}_p[R_1, \ldots, R_{q_O}, X, Y^{\pm 1}]$ yields a tuple that constitutes a solution for the underlying NPSPS problem.

In the game, we keep 3 different lists $\mathcal{L}_\mathbb{G}$, $\mathcal{L}_\mathbb{H}$ and $\mathcal{L}_\mathbb{T}$ for the Laurent polynomials corresponding to group elements from groups $\mathbb{G}$, $\mathbb{H}$ and $\mathbb{T}$, respectively. At the end of the game, the total number of (non-constant) Laurent polynomials used is $|\mathcal{L}_\mathbb{G}| + |\mathcal{L}_\mathbb{H}| + |\mathcal{L}_\mathbb{T}| \le 2 + q_G + q_P + 2q_O$.

Since both elements in the adversary's output (R^*, S^*) are from $\mathbb{G}$, it follows that r^* and s^* can only be constructed using linear combinations of the Laurent polynomials corresponding to elements from $\mathbb{G}$. Thus, we must have that:

$$r^* = a_r + \sum_{i=1}^{q_O} b_{r,i} r_i + \sum_{i=1}^{q_O} c_{r,i} \left(\frac{r_i x}{y} + \frac{r_i m_i}{y} \right)$$

$$s^* = a_s + \sum_{i=1}^{q_O} b_{s,i} r_i + \sum_{i=1}^{q_O} c_{s,i} \left(\frac{r_i x}{y} + \frac{r_i m_i}{y} \right)$$

For the pair $(R^*, S^*) \in {\mathbb{G}^\times}^2$ to be a valid solution, we must have that:

$$s^* y = r^* x + r^* m^* \tag{3}$$

Thus, we must have:

$$\begin{aligned} a_s y + \sum_{i=1}^{q_O} b_{s,i} r_i y + \sum_{i=1}^{q_O} c_{s,i} \left(r_i x + r_i m_i\right) \\ &= a_r x + \sum_{i=1}^{q_O} b_{r,i} r_i x + \sum_{i=1}^{q_O} c_{r,i} \left(\frac{r_i x^2}{y} + \frac{r_i m_i x}{y}\right) \\ &+ \left(a_r + \sum_{i=1}^{q_O} b_{r,i} r_i + \sum_{i=1}^{q_O} c_{r,i} \left(\frac{r_i x}{y} + \frac{r_i m_i}{y}\right)\right) m^* \end{aligned}$$

There is no term in y or $r_i y$ on the RHS, so we must have $a_s = 0$, $b_{s,i} = 0$ for all $i \in [q_O]$. Thus, we have:

$$\begin{aligned} \sum_{i=1}^{q_O} c_{s,i}(r_i x + r_i m_i) = a_r x + \sum_{i=1}^{q_O} b_{r,i} r_i x + \sum_{i=1}^{q_O} c_{r,i}\left(\frac{r_i x^2}{y} + \frac{r_i m_i x}{y}\right) \\ + \left(a_r + \sum_{i=1}^{q_O} b_{r,i} r_i + \sum_{i=1}^{q_O} c_{r,i}\left(\frac{r_i x}{y} + \frac{r_i m_i}{y}\right)\right) m^* \end{aligned}$$

There is no term $\frac{r_i x^2}{y}$ on the LHS, so we must have that $c_{r,i} = 0$ for all $i \in [q_O]$. Also, no term in x on the LHS, so we must have that $a_r = 0$. Thus, we have:

$$\sum_{i=1}^{q_O} c_{s,i} \left(r_i x + r_i m_i\right) = \sum_{i=1}^{q_O} b_{r,i} r_i x + \sum_{i=1}^{q_O} b_{r,i} r_i m^*$$

The monomial $r_i x$ implies $c_{s,i} = b_{r,i}$ for all $i \in [q_O]$. Since we must have that that $R^* \in \mathbb{G}^\times$, we must have $r^* \neq 0$ and therefore we must have at least a single value of $c_{s,i} = b_{r,i} \neq 0$. The monomial r_i implies $c_{s,i} m_i = b_{r,i} m^*$ which means $m^* = m_i$ for some i. Thus, the pair (R^*, S^*) is not a valid new pair.

Thus far we have proven that the adversary is unable to symbolically produce a valid tuple for a new scalar. What remains is to bound the probability that the simulation fails. The adversary wins if for any two different Laurent polynomials F and F' in any of the 3 lists evaluate to the same value. Note that the only indeterminate in those Laurent polynomials with a negative power is Y. Thus, for any Laurent polynomial F on any of those 3 lists, we can view F as a fraction of polynomials $F = \frac{F_n}{F_d}$ for some polynomials $F_n \in \mathbb{Z}_p[R_1, \ldots, R_{q_O}, X, Y]$ and $F_d \in \mathbb{Z}_p[Y]$. Note that $\mathbb{Z}_p[Y] \subset \mathbb{Z}_p[R_1, \ldots, R_{q_O}, X, Y]$. Thus, the equality check $F(r_1, \ldots, r_O, x, y, y^{-1}) - F'(r_1, \ldots, r_O, x, y, y^{-1}) = 0$ can be substituted by checking whether $F_n(r_1, \ldots, r_O, x, y) F'_d(y) - F'_n(r_1, \ldots, r_O, x, y,) F_d(y) = 0$. It follows that for $F, F' \in \mathcal{L}_{\mathbb{G}}$ we have $\deg(F_n) \leq 2$ and $\deg(F_d) \leq 1$. Thus, the probability that $F_n(r_1, \ldots, r_O, x, y) F'_d(y) - F'_n(r_1, \ldots, r_O, x, y) F_d(y) = 0$ is $\leq \frac{3}{p}$. For $F, F' \in \mathcal{L}_{\mathbb{H}}$, we have $\deg(F_n) \leq 1$ and $\deg(F_d) = 0$. Thus, the probability that $F_n(r_1, \ldots, r_O, x, y) F'_d(y) - F'_n(r_1, \ldots, r_O, x, y) F_d(y) = 0$ is $\leq \frac{1}{p}$. From this it follows that for $F, F' \in \mathcal{L}_{\mathbb{T}}$ the probability that $F_n(r_1, \ldots, r_O, x, y) F'_d(y) - F'_n(r_1, \ldots, r_O, x, y) F_d(y) = 0$ is $\leq \frac{4}{p}$.

Summing over all choices of F and F' in each case we have that the probability ϵ of the simulation failing for this reason is

$$\epsilon \leq \binom{|\mathcal{L}_1|}{2}\frac{3}{p} + \binom{|\mathcal{L}_2|}{2}\frac{1}{p} + \binom{|\mathcal{L}_T|}{2}\frac{4}{p} \leq \frac{2(2+q_G+q_P+2q_O)^2}{p}.$$

Thus, we have that the probability of the simulation failing is $\mathcal{O}(\frac{q_G^2+q_P^2+q_O^2}{p})$. Since by definition we have that q_O, q_G and q_p are all polynomial in κ whereas $\log p \in \Theta(\kappa)$, it follows that the adversary's advantage is negligible. □

The following theorem proves the unforgeability of the scheme.

Theorem 2. *The scheme is EUF-CMA if the NPSPS assumption holds.*

Proof. Let $\mathcal{A}$ be an adversary against the unforgeability of the scheme, we use $\mathcal{A}$ in a blackbox manner to construct an adversary $\mathcal{B}$ against the NPSPS assumption.

Adversary $\mathcal{B}$ gets $(\mathcal{P}, \tilde{X}, \tilde{Y})$ from her game and chooses $y_1, \ldots, y_{n-1}, \alpha_1, \ldots, \alpha_{n-1} \leftarrow \mathbb{Z}_p^\times$ and sets $\tilde{Z} := \tilde{Y}$ and $\tilde{Y}_i := \tilde{Y}^{\alpha_i}\tilde{H}^{y_i}$ for all $i \in [n-1]$. $\mathcal{B}$ initiates $\mathcal{A}$ on $\mathsf{vk} := (\tilde{X}, \tilde{Y}_1, \ldots, \tilde{Y}_{n-1}, \tilde{Z})$. Note that verification key is distributed identically to that of the scheme.

When $\mathcal{A}$ queries the sign oracle on a vector $\boldsymbol{m} \in \mathbb{Z}_p^n$, $\mathcal{B}$ computes $m' := m_1 + \sum_{i=2}^{n} y_{i-1}m_i$ and queries her $\widehat{\mathcal{O}}_{\tilde{X},\tilde{Y}}$ oracle on m' to get a tuple $(S_1, S_2) \in \mathbb{G}$. $\mathcal{B}$ computes $S_2' := S_2 S_1^{\sum_{i=2}^{n} \alpha_{i-1}m_i}$ and returns $\sigma := (S_1, S_2')$ to $\mathcal{A}$ as a signature on $\boldsymbol{m}$. This is a valid signature on $\boldsymbol{m}$ w.r.t vk since:

$$\begin{aligned}
e(S_2', \tilde{Z}) &= e(S_1^{\frac{x+m_1+\sum_{i=2}^{n} y_{i-1}m_i}{z}} S_1^{\sum_{i=2}^{n} \alpha_{i-1}m_i}, \tilde{Z}) \\
&= e(S_1^{x+m_1+\sum_{i=2}^{n} y_{i-1}m_i + z\sum_{i=2}^{n} \alpha_{i-1}m_i}, \tilde{H}) \\
&= e(S_1^{x+m_1+\sum_{i=2}^{n} (y_{i-1}+\alpha_{i-1}z)m_i}, \tilde{H}) \\
&= e(S_1, \tilde{X}\tilde{H}^{m_1+\sum_{i=2}^{n} (y_{i-1}+\alpha_{i-1}z)m_i}, \tilde{H}) \\
&= e(S_1, \tilde{X}\tilde{H}^{m_1}\prod_{i=2}^{n} \tilde{Y}_{i-1}^{m_i}).
\end{aligned}$$

Eventually, when $\mathcal{A}$ halts and outputs her forgery $(\boldsymbol{m}^*, \sigma^*)$, $\mathcal{B}$ computes $m^{*\prime} := m_1^* + \sum_{i=2}^{n} y_{i-1}m_i^*$ and returns $(\sigma^*, m^{*\prime})$ as her output in her game.

It is easy to see that if $\sigma^* = (S_1^*, S_2^*)$ is a signature on the new vector $\boldsymbol{m}^*$ which was not queried to the sign oracle, σ^* is a valid NPSPS tuple on the new scalar $m^{*\prime}$ which $\mathcal{B}$ did not submit to her oracle $\widehat{\mathcal{O}}_{\tilde{X},\tilde{Y}}$.

We need to handle the case where $\boldsymbol{m}^* \notin \{\boldsymbol{m}_i\}_{i=1}^q$ but $m^{*\prime} = m_i'$ for some $i \in [q]$ in which case $\mathcal{A}$ wins her game but $\mathcal{B}$ will not be able to break the NPSPS assumption since the returned tuple is not on a new scalar that was not queried to her oracle. Note that $\mathcal{A}$'s view is independent of the y_i's and hence the probability that this event happens is $\leq \frac{q}{p}$ which is negligible. □

5 A New Optimal One-Time sEUF-CMA LmSPSPS Scheme

Here we give an optimal one-time LmSPSPS scheme for a vector of messages with one-element signatures and a verification key of size $n|\mathbb{G}| + |\mathbb{H}|$. The scheme is optimal in every respect.

Given the description of Type-3 bilinear groups $\mathcal{P}$ output by $\mathcal{BG}(1^\kappa)$, the scheme is as follows:

- $\mathsf{KeyGen}(\mathcal{P})$: Select $x_1, \ldots, x_n, y \leftarrow \mathbb{Z}_p^\times$. Set $\mathsf{sk} := (x_1, \ldots, x_n, y)$, $\mathsf{vk} := (X_1, \ldots, X_n, \tilde{Y}) = (G^{x_1}, \ldots, G^{x_n}, \tilde{H}^y) \in \mathbb{G}^n \times \mathbb{H}$.
- $\mathsf{Sign}(\mathsf{sk}, \boldsymbol{m})$: To sign $\boldsymbol{m} \in \mathbb{Z}_p^n$, compute $\sigma = S := G^{\frac{1+\sum_{i=1}^{n} x_i m_i}{y}}$.
- $\mathsf{Verify}(\mathsf{vk}, \boldsymbol{m}, \sigma = S)$: Return 1 iff $e(S, \tilde{Y}) = e(G \prod_{i=1}^{n} X_i^{m_i}, \tilde{H})$ and 0 otherwise.

Correctness of the scheme is straightforward to verify. We now prove the one-time strong unforgeability of the scheme.

Theorem 3. *The scheme is sEUF-CMA secure in the generic group model.*

Proof. We prove that no linear combinations corresponding to polynomials in the discrete logarithms of the elements the adversary sees correspond to a forgery.

At the start of the game, the only elements in $\mathbb{H}$ the adversary sees are $\tilde{H}$, $\tilde{Y}$, which correspond to the discrete logarithms $1, y$ respectively. Note the sign oracle produces no new elements in $\mathbb{H}$. When queried on a message $\boldsymbol{m}$, the oracle will return a signature $S = G^{\frac{1+\sum_{i=1}^{n} x_i m_i}{y}} \in \mathbb{G}$. The forgery $\sigma^* = S^*$ can only be a linear combination of the group elements from $\mathbb{G}$, i.e. a linear combination of $G, S, X_1, \ldots, X_n$. Thus, we have

$$s^* = \alpha_s + \beta_s \frac{(1 + \sum_{i=1}^{n} m_i x_i)}{y} + \sum_{i=1}^{n} \gamma_{s_i} x_i$$

For the forgery to be accepted, $(s^*, \boldsymbol{m}^*)$ has to satisfy $s^* y = 1 + \sum_{i=1}^{n} m_i^* x_i$. Therefore, we must have

$$\left(\alpha_s + \beta_s \frac{(1 + \sum_{i=1}^{n} m_i x_i)}{y} + \sum_{i=1}^{n} \gamma_{s_i} x_i \right) y = 1 + \sum_{i=1}^{n} m_i^* x_i$$

Therefore, we must have

$$\alpha_s y + \beta_s \left(1 + \sum_{i=1}^{n} m_i x_i\right) + \sum_{i=1}^{n} \gamma_{_{s_i}} x_i y = 1 + \sum_{i=1}^{n} m_i^* x_i$$

There is no terms of the form y or $x_i y$ for any $i \in [n]$ on the RHS, so we must have that $\alpha_s = 0$ and $\gamma_{_{s_i}} = 0$ for all $i \in [n]$. Thus, we have that

$$\beta_s + \sum_{i=1}^{n} \beta_s m_i x_i = 1 + \sum_{i=1}^{n} m_i^* x_i$$

The constant term implies that $\beta_s = 1$. The monomial x_i implies that $\beta_s m_i = m_i^*$ from which it follows that we must have that $m_i^* = m_i$ for all $i \in [n]$ which means the forgery can only be the same signature on $\boldsymbol{m}$ the adversary obtained from the sign oracle.

The probability of the simulation failing is $\leq \frac{3(n+2+q_G+q_P)^2}{2p}$, i.e. $\mathcal{O}(\frac{n^2+q_G^2+q_P^2}{p})$. Since by definition we have that n, q_G and q_p are all polynomial in κ whereas $\log p \in \Theta(\kappa)$, it follows that the adversary's advantage is negligible. □

6 From LmSPSPS Schemes into SPS Schemes

In this section we give a generic framework for transforming any disjoint LmSPSPS scheme into a structure-preserving scheme for the message space $\mathcal{DH}^n$.

Let $\mathcal{P} := (\mathbb{G}, \mathbb{H}, \mathbb{T}, p, G, \tilde{H}, e)$ be the bilinear group description generated by $\mathcal{BG}$. Let $\mathsf{DLmSPSPS} = \left(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, [\mathsf{Randomize}]\right)$ be a (s)EUF-CMA (resp. (s)EUF-RMA) DLmSPSPS scheme. The following transformation yields a (s)EUF-CMA (resp. (s)EUF-RMA) SPS scheme $\mathsf{SPS} = \left(\mathsf{KeyGen}_{\mathsf{SPS}}, \mathsf{Sign}_{\mathsf{SPS}}, \mathsf{Verify}_{\mathsf{SPS}}, [\mathsf{Randomize}_{\mathsf{SPS}}]\right)$.

- $\mathsf{KeyGen}_{\mathsf{SPS}}(\mathcal{P})$: Run $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(\mathcal{P})$. Return $(\mathsf{sk}_{\mathsf{SPS}} := \mathsf{sk}, \mathsf{vk}_{\mathsf{SPS}} := \mathsf{vk})$.
- $\mathsf{Sign}_{\mathsf{SPS}}\left(\mathsf{sk}_{\mathsf{SPS}}, \left((M_1, \tilde{M}_1), \ldots, (M_n, \tilde{M}_n)\right)\right)$:
  - Decompose the PPE equations of DLmSPSPS to the following form:

$$\prod_{i=1}^{\nu} e(S_i, \prod_{j=1}^{\mu'} \tilde{Y}_j)^{a_{i,j,\ell}} \prod_{i=1}^{\nu} \prod_{j=1}^{\mu'} \prod_{k=1}^{n} e(S_i, \tilde{Y}_j)^{a'_{i,j,\ell,k} m_k} \prod_{i=1}^{\nu'} e(\prod_{j=1}^{\mu} X_j, \tilde{T}_i)^{b_{i,j,\ell}}$$

$$\prod_{i=1}^{\nu'} \prod_{j=1}^{\mu} \prod_{k=1}^{n} e(X_j, \tilde{T}_i)^{b'_{i,j,\ell,k} m_k} \prod_{i=1}^{\nu} e(S_i, \prod_{j=1}^{\nu'} \tilde{T}_j)^{c_{i,j,\ell}} \prod_{i=1}^{\nu} \prod_{j=1}^{\nu'} \prod_{k=1}^{n} e(S_i, \tilde{T}_j)^{c'_{i,j,\ell,k} m_k}$$

$$\prod_{i=1}^{\mu} \prod_{j=1}^{\mu'} e(X_i, \tilde{Y}_j)^{d_{i,j,\ell}} \prod_{i=1}^{\mu} \prod_{j=1}^{\mu'} \prod_{k=1}^{n} e(X_i, \tilde{Y}_j)^{d'_{i,j,\ell,k} m_k} = Z_\ell \cdot$$

- ∘ Initialize 2 empty lists E_1 and E_2 of triples representing PPE equations.
- ∘ For each signature component $S_j \in \mathbb{G}$ of $\mathsf{DLmSPSPS}$:
  - ∗ Parse S_j as $G^{\frac{\sum_{i=1}^{q} x^{c_{i,j}} y^{c'_{i,j}} r^{c''_{i,j}} \left(a_{i,j} + \sum_{k=1}^{n} d_{i,j,k} m_k\right)}{\sum_i b_{i,j} x^{e_{i,j}} y^{e'_{i,j}} r^{e''_{i,j}}}}$.
  - ∗ Define the set $\hat{I} \subseteq [q]$ as the subset of indices i where $\exists k \in [n]$ where $d_{i,j,k} \neq 0$ and let $\check{I} := [q] \backslash \hat{I}$. Compute S_j of SPS as:

$$S_j = \left(G^{\sum_{i \in \check{I}} a_{i,j} x^{c_{i,j}} y^{c'_{i,j}} r^{c''_{i,j}}} \prod_{i \in \hat{I}} \prod_{k=1}^{n} M_k^{d_{i,j,k} x^{c_{i,j}} y^{c'_{i,j}} r^{c''_{i,j}}} \right)^{\frac{1}{\sum_i b_{i,j} x^{e_{i,j}} y^{e'_{i,j}} r^{e''_{i,j}}}}.$$

- ∘ For each signature component $\tilde{T}_j \in \mathbb{H}$ of $\mathsf{DLmSPSPS}$:
  - ∗ Parse $\tilde{T}_j$ as $\tilde{H}^{\frac{\sum_{i=1}^{q} x^{c_{i,j}} y^{c'_{i,j}} r^{c''_{i,j}} \left(a_{i,j} + \sum_{k=1}^{n} d_{i,j,k} m_k\right)}{\sum_i b_{i,j} x^{e_{i,j}} y^{e'_{i,j}} r^{e''_{i,j}}}}$.
  - ∗ Define the set $\hat{I} \subseteq [q]$ as the subset of indices i where $\exists k \in [n]$ where $d_{i,j,k} \neq 0$ and let $\check{I} := [q] \backslash \hat{I}$. Compute $\tilde{T}_j$ of SPS as:

$$\tilde{T}_j = \left(\tilde{H}^{\sum_{i \in \check{I}} a_{i,j} x^{c_{i,j}} y^{c'_{i,j}} r^{c''_{i,j}}} \prod_{i \in \hat{I}} \prod_{k=1}^{n} \tilde{M}_k^{d_{i,j,k} x^{c_{i,j}} y^{c'_{i,j}} r^{c''_{i,j}}} \right)^{\frac{1}{\sum_i b_{i,j} x^{e_{i,j}} y^{e'_{i,j}} r^{e''_{i,j}}}}.$$

- ∘ For each PPE verification equation of $\mathsf{DLmSPSPS}$:
  - ∗ For each pairing of the form $e(S_i, \tilde{Y}_j)^{a'_{i,j,\ell,k} m_k}$ where $a'_{i,j,\ell,k} \neq 0$:
    - · If $\tilde{Y}_j \neq \tilde{H}$, append (if it does not already exist) $S'_i = M_k^{s_i}$ to $\boldsymbol{S}$, replace the pairing with $e(S'_i, \tilde{Y}_j)^{a'_{i,j,\ell,k}}$ and append (if it does not already exist) the tuple $(S'_i, S_i, \tilde{M}_k)$ to E_1. Note that S_i is independent of the message m_k so knowledge of the discrete logarithm m_k is not required to compute S'_i.
    - · Otherwise, replace the above pairing with $e(S_i, \tilde{M}_k)^{a'_{i,j,\ell,k}}$.
  - ∗ For each pairing of the form $e(X_i, \tilde{T}_j)^{b'_{i,j,\ell,k} m_k}$ where $b'_{i,j,\ell,k} \neq 0$:
    - · If $X_i \neq G$, append (if it does not already exist) $\tilde{T}'_j = \tilde{M}_k^{t_j}$ to $\boldsymbol{T}$, replace the pairing with $e(X_i, \tilde{T}'_j)^{b'_{i,j,\ell,k}}$ and append (if it does not already exist) the tuple $(\tilde{T}'_j, \tilde{T}_j, M_k)$ to E_2. Note that $\tilde{T}_j$ is independent of the message m_k so knowledge of the discrete logarithm m_k is not required to compute $\tilde{T}'_j$.
    - · Otherwise, replace the pairing with $e(M_k, \tilde{T}_j)^{b'_{i,j,\ell,k}}$.
  - ∗ For each pairing of the form $e(X_i, \tilde{Y}_j)^{d'_{i,j,\ell,k} m_k}$ where $d'_{i,j,\ell,k} \neq 0$:
    - · If $X_i = G$, replace the pairing with $e(M_k, \tilde{Y}_j)^{d'_{i,j,\ell,k}}$.
    - · If $X_i \neq G$ but $Y_j = \tilde{H}$, replace the pairing with $e(X_i, \tilde{M}_k)^{d'_{i,j,\ell,k}}$.
    - · If $X_i \neq G$ and $\tilde{Y}_j \neq \tilde{H}$, append (if it does not already exist) $S_{|\boldsymbol{S}|+1} = M_k^{x_i}$ to $\boldsymbol{S}$, replace the pairing with $e(S_{|\boldsymbol{S}|+1}, \tilde{Y}_j)^{d'_{i,j,\ell,k}}$ and append the tuple $(S_{|\boldsymbol{S}|+1}, X_i, \tilde{M}_k)$ to E_1.

- ∗ For each pairing of the form $e(S_i, \tilde{T}_j)^{c'_{i,j,\ell,k} m_k}$ where $c'_{i,j,\ell,k} \neq 0$: Note that by definition m_k cannot appear in the denominator of S_i or $\tilde{T}_j$. Also, we must have that at least one of the signature components is independent of m_k.
  - · If S_i is independent of m_k, append (if it does not already exist) $S'_i = M_k^{s_i}$ to $\boldsymbol{S}$, replace the pairing with $e(S'_i, \tilde{T}_j)^{c'_{i,j,\ell,k}}$ and append (if it does not already exist) the tuple $(S'_i, S_i, \tilde{M}_k)$ to E_1.
  - · Otherwise, append (if it does not already exist) $\tilde{T}'_j = \tilde{M}_k^{t_j}$ to $\boldsymbol{T}$, replace the pairing with $e(S_i, \tilde{T}'_j)^{c'_{i,j,\ell,k}}$ and append (if it does not already exist) the tuple $(\tilde{T}'_j, \tilde{T}_j, M_k)$ to E_2.

- $\mathsf{Verify}_{\mathsf{SPS}}\Big(\mathsf{vk}_{\mathsf{SPS}}, \big((M_1, \tilde{M}_1), \ldots, (M_n, \tilde{M}_n)\big), \sigma_{\mathsf{SPS}}\Big)$: Return 1 if all the following holds and 0 otherwise:
  - ◦ All modified PPEs of $\mathsf{DLmSPSPS}$ verify correctly.
  - ◦ For each tuple i in E_1, it holds that: $e(\mathsf{E}_1[i][0], \tilde{H}) = e(\mathsf{E}_1[i][1], \mathsf{E}_1[i][2])$.
  - ◦ For each tuple i in E_2, it holds that: $e(G, \mathsf{E}_2[i][0]) = e(\mathsf{E}_2[i][2], \mathsf{E}_2[i][1])$.
  - ◦ All signature group membership required by $\mathsf{DLmSPSPS}$ verify correctly.
  - ◦ $\Big((M_1, \tilde{M}_1), \ldots, (M_n, \tilde{M}_n)\Big) \in \mathcal{DH}^n$.[3]

Efficiency. What determines the added cost in σ_{SPS} compared to σ is distinct pairings of the form $e(S_i, \tilde{Y}_j)^{a'_{i,j,\ell,k} m_k}$ where $a'_{i,j,\ell,k} \neq 0$ and $Y_j \neq \tilde{H}$, $e(X_i, \tilde{T}_j)^{b'_{i,j,\ell,k} m_k}$ where $b'_{i,j,\ell,k} \neq 0$ and $X_i \neq G$, and $e(S_i, \tilde{T}_j)^{c'_{i,j,\ell,k} m_k}$ where $c'_{i,j,\ell,k} \neq 0$ in the verification equations of the $\mathsf{DLmSPSPS}$ scheme. Each distinct pairing of those 3 types adds an extra signature component in $\mathbb{H}$, $\mathbb{G}$, and $\mathbb{G}/\mathbb{H}$ depending on which component is independent of the message, respectively, to σ_{SPS} compared to σ. Also, each distinct pairing of any of those 3 types would add an additional PPE equation involving 2 pairings to the verification overhead of SPS compared to that of $\mathsf{DLmSPSPS}$. Each distinct pairing of the form $e(X_i, \tilde{Y}_j)^{d'_{i,j,\ell,k} m_k}$ where $X_i \neq G$, $\tilde{Y}_j \neq \tilde{H}$ and $d'_{i,j,\ell,k} \neq 0$ incurs an additional signature component in $\mathbb{G}$ and an additional PPE involving 2 pairings. Note that the latter cost is constant for multiple signatures on the same message.

One maintains the same signature size and verification overhead of $\mathsf{DLmSPSPS}$ (modulo the cost for verifying the well-formedness of the messages in SPS) when the verification of $\mathsf{DLmSPSPS}$ does not involve any pairings of the above forms. Also, it is easy to see that if the original scheme yields randomizable signatures, the same applies to the resultant SPS scheme.

6.1 Example Instantiations

We give some example instantiations of our transformation for the sake of illustration. The first example shown in Fig. 1 shows how to transform our new LmSPSPS scheme into an SPS scheme for a vector of messages which also captures the single-message SPS scheme from [31] as a special case. The second

[3] Batch verification techniques, e.g. [9], can speed up this step.

example shown in Fig. 2 shows how to transform the PS scheme [42] into a SPS scheme for a vector of messages which also captures the single-message SPS scheme from [29] as a special case.

```
$\mathsf{KeyGen}(\mathcal{P})$
    $x, y_1, \ldots, y_{n-1}, z \leftarrow \mathbb{Z}_p$.
    $\tilde{X} := \tilde{H}^x, \tilde{Y}_i := \tilde{H}^{y_i}$ for $i \in [n-1]$, $\tilde{Z} := \tilde{H}^z$.
    $\mathsf{sk} := (x, \{y_i\}_{i=1}^{n-1}, z)$. $\mathsf{vk} := (\tilde{X}, \{\tilde{Y}_i\}_{i=1}^{n-1}, \tilde{Z})$.
    Return $(\mathsf{sk}, \mathsf{vk})$.

$\mathsf{Sign}(\mathsf{sk}, \boldsymbol{m})$
    Parse $\mathsf{sk}$ as $(x, \{y_i\}_{i=1}^{n-1}, z)$.
    $r \leftarrow \mathbb{Z}_p^\times, S_1 := G^r, S_2 := G^{\frac{r(x+m_1+\sum_{i=2}^{n} m_i y_{i-1})}{z}}$.
    Return $\sigma := (S_1, S_2)$.

$\mathsf{Verify}(\mathsf{vk}, \boldsymbol{m}, \sigma)$
    Parse $\mathsf{vk}$ as $(\tilde{X}, \{\tilde{Y}_i\}_{i=1}^{n-1}, \tilde{Z})$.
    If $S_1 = 1_{\mathbb{G}} \vee e(S_2, \tilde{Z}) \neq e(S_1, \tilde{X}\tilde{H}^{m_1} \prod_{i=2}^{n} \tilde{Y}_{i-1}^{m_i})$
    Then Return 0 Else Return 1.

$\mathsf{Randomize}(\mathsf{vk}, \boldsymbol{m}, \sigma = (S_1, S_2))$
    $r' \leftarrow \mathbb{Z}_p^\times$, Return $\sigma' := \sigma^{r'}$.
```

```
$\mathsf{KeyGen}_{\mathsf{SPS}}(\mathcal{P})$
    $x, y_1, \ldots, y_{n-1}, z \leftarrow \mathbb{Z}_p$.
    $\tilde{X} := \tilde{H}^x, \tilde{Y}_i := \tilde{H}^{y_i}$ for $i \in [n-1]$, $\tilde{Z} := \tilde{H}^z$.
    $\mathsf{sk}_{\mathsf{SPS}} := (x, \{y_i\}_{i=1}^{n-1}, z)$. $\mathsf{vk}_{\mathsf{SPS}} := (\tilde{X}, \{\tilde{Y}_i\}_{i=1}^{n-1}, \tilde{Z})$.
    Return $(\mathsf{sk}_{\mathsf{SPS}}, \mathsf{vk}_{\mathsf{SPS}})$.

$\mathsf{Sign}_{\mathsf{SPS}}\left(\mathsf{sk}_{\mathsf{SPS}}, \left((M_1, \tilde{M}_1), \ldots, (M_n, \tilde{M}_n)\right)\right)$
    Parse $\mathsf{sk}_{\mathsf{SPS}}$ as $(x, \{y_i\}_{i=1}^{n-1}, z)$.
    $r \leftarrow \mathbb{Z}_p^\times, S_1 := G^r, S'_{1,i} := M_i^r$ for all $i \in [2, n]$,
     $S_2 := \left(G^x M_1 \prod_{i=2}^{n} M_i^{y_{i-1}}\right)^{\frac{r}{z}}$.
    Return $\sigma_{\mathsf{SPS}} := (S_1, S_2, S'_{1,2}, \ldots, S'_{1,n})$.

$\mathsf{Verify}_{\mathsf{SPS}}\left(\mathsf{vk}_{\mathsf{SPS}}, \left((M_1, \tilde{M}_1), \ldots, (M_n, \tilde{M}_n)\right), \sigma_{\mathsf{SPS}}\right)$
    Parse $\mathsf{vk}_{\mathsf{SPS}}$ as $(\tilde{X}, \{\tilde{Y}_i\}_{i=1}^{n-1}, \tilde{Z})$.
    Parse $\sigma_{\mathsf{SPS}}$ as $(S_1, S_2, S'_{1,2}, \ldots, S'_{1,n})$.
    If $S_1 = 1_{\mathbb{G}} \vee e(S'_{1,i}, \tilde{H}) \neq e(S_1, \tilde{M}_i)$ for any $i \in [2, n]$
     $\vee\, e(S_2, \tilde{Z}) \neq e(S_1, \tilde{X}\tilde{M}_1) \prod_{i=2}^{n} e(S'_{1,i}, \tilde{Y}_{i-1})$
    Then Return 0 Else Return 1.

$\mathsf{Randomize}_{\mathsf{SPS}}\left(\mathsf{vk}_{\mathsf{SPS}}, \left((M_1, \tilde{M}_1), \ldots, (M_n, \tilde{M}_n)\right), \sigma_{\mathsf{SPS}}\right)$
    $r' \leftarrow \mathbb{Z}_p^\times$, Return $\sigma'_{\mathsf{SPS}} := \sigma_{\mathsf{SPS}}^{r'}$.
```

Fig. 1. Transforming our new scheme into a SPS scheme for a vector of messages

6.2 Security

Correctness follows from that of DLmSPSPS and the fact that any added PPE to the verification of the SPS scheme will verify. We now prove the following theorem regarding the unforgeability of the obtained SPS scheme.

Theorem 4. *If* DLmSPSPS *is (s)EUF-CMA (resp. (s)EUF-RMA),* SPS *is (s)EUF-CMA (resp. (s)EUF-RMA) in the generic group model.*

Proof. Since DLmSPSPS is unforgeable, it holds that no generic adversary against it can obtain a forgery using linear combinations of the (fraction of) polynomials corresponding to the group elements she sees in the game. We prove that a generic adversary $\mathcal{A}_{\mathsf{SPS}}$ against SPS does not see any additional group elements other than what $\mathcal{A}_{\mathsf{DLmSPSPS}}$ can see in her game and hence it holds that no linear combinations of the (fraction of) polynomials $\mathcal{A}_{\mathsf{SPS}}$ sees leads to a forgery against SPS.

Before the 1st sign query, the group elements $\mathcal{A}_{\mathsf{SPS}}$ sees are the same as those $\mathcal{A}_{\mathsf{DLmSPSPS}}$ can see at the start of her game which include the public key $(\boldsymbol{X}, \tilde{\boldsymbol{Y}}) \in \mathbb{G}^{\mu} \times \mathbb{H}^{\mu'}$. By definition, such a key is disjoint. Now at the first sign query on a valid message vector $\left((M_{1,1}, \tilde{M}_{1,1}), \ldots, (M_{1,n}, \tilde{M}_{1,n})\right)$, it follows that

```
$\mathsf{KeyGen}(\mathcal{P})$
    $x, y_1, \dots, y_n \leftarrow \mathbb{Z}_p$.
    $\tilde{X} := \tilde{H}^x, \tilde{Y}_i := \tilde{H}^{y_i}$ for $i \in [n]$.
    $\mathsf{sk} := (x, \{y_i\}_{i=1}^n), \mathsf{vk} := (\tilde{X}, \{\tilde{Y}_i\}_{i=1}^n)$.
    Return $\big(\mathsf{sk}, \mathsf{vk}\big)$.
$\mathsf{Sign}(\mathsf{sk} = (x, \{y_i\}_{i=1}^n), \boldsymbol{m})$
    $r \leftarrow \mathbb{Z}_p^\times, S_1 := G^r, S_2 := G^{r(x+\sum_{i=1}^{n} m_i y_i)}$.
    Return $\sigma := (S_1, S_2)$.
$\mathsf{Verify}(\mathsf{vk}, \boldsymbol{m}, \sigma)$
    Parse $\mathsf{vk}$ as $(\tilde{X}, \{\tilde{Y}_i\}_{i=1}^n)$.
    Parse $\sigma$ as $(S_1, S_2)$.
    If $S_1 = 1_{\mathbb{G}} \vee e(S_2, \tilde{H}) \neq e(S_1, \tilde{X} \prod_{i=1}^{n} \tilde{Y}_i^{m_i})$
     Then Return 0 Else Return 1.
$\mathsf{Randomize}(\mathsf{vk}, \boldsymbol{m}, \sigma = (S_1, S_2))$
    $r' \leftarrow \mathbb{Z}_p^\times, \sigma' := \sigma^{r'}$.
```

```
$\mathsf{KeyGen}_{\mathsf{SPS}}(\mathcal{P})$
    $x, y_1, \dots, y_n \leftarrow \mathbb{Z}_p$.
    $\tilde{X} := \tilde{H}^x, \tilde{Y}_i := \tilde{H}^{y_i}$ for $i \in [n]$.
    $\mathsf{sk}_{\mathsf{SPS}} := (x, \{y_i\}_{i=1}^n), \mathsf{vk}_{\mathsf{SPS}} := (\tilde{X}, \{\tilde{Y}_i\}_{i=1}^n)$.
    Return $\big(\mathsf{sk}_{\mathsf{SPS}}, \mathsf{vk}_{\mathsf{SPS}}\big)$.
$\mathsf{Sign}_{\mathsf{SPS}}\Big(\mathsf{sk}_{\mathsf{SPS}}, \Big((M_1, \tilde{M}_1), \dots, (M_n, \tilde{M}_n)\Big)\Big)$
    Parse $\mathsf{sk}_{\mathsf{SPS}}$ as $(x, \{y_i\}_{i=1}^n)$.
    $r \leftarrow \mathbb{Z}_p^\times, S_1 := G^r, S'_{1,i} := M_i^r$ for $i \in [n]$,
        $S_2 := (G^x \prod_{i=1}^{n} M_i^{y_i})^r$.
    Return $\sigma_{\mathsf{SPS}} := (S_1, S_2, S'_{1,1}, \dots, S'_{1,n})$.
$\mathsf{Verify}_{\mathsf{SPS}}\Big(\mathsf{vk}_{\mathsf{SPS}}, \Big((M_1, \tilde{M}_1), \dots, (M_n, \tilde{M}_n)\Big), \sigma_{\mathsf{SPS}}\Big)$
    Parse $\sigma_{\mathsf{SPS}}$ as $(S_1, S_2, S'_{1,1}, \dots, S'_{1,n})$.
    Parse $\mathsf{vk}_{\mathsf{SPS}}$ as $(\tilde{X}, \{\tilde{Y}_i\}_{i=1}^n)$.
    If $S_1 = 1_{\mathbb{G}} \vee e(S_2, \tilde{H}) \neq e(S_1, \tilde{X}) \prod_{i=1}^{n} e(S'_{1,i}, \tilde{Y}_i)$
      $\vee\, e(S'_{1,i}, \tilde{H}) \neq e(S_1, \tilde{M}_i)$ for any $i \in [n]$
     Then Return 0 Else Return 1.
$\mathsf{Randomize}_{\mathsf{SPS}}\Big(\mathsf{vk}_{\mathsf{SPS}}, \Big((M_1, \tilde{M}_1), \dots, (M_n, \tilde{M}_n)\Big), \sigma_{\mathsf{SPS}}\Big)$
    $r' \leftarrow \mathbb{Z}_p^\times, \sigma'_{\mathsf{SPS}} := \sigma_{\mathsf{SPS}}^{r'}$.
```

Fig. 2. Transforming the PS scheme [42] into a SPS scheme for a vector of messages

the discrete logarithm $m_{1,i}$ of the message component $(M_{1,i}, \tilde{M}_{1,i})$ for all $i \in [n]$ corresponds to a constant polynomial. The 1st sign query will return a signature of the form $\sigma_1 = (\boldsymbol{S}_1, \tilde{\boldsymbol{T}}_1)$. By definition such a returned signature still conforms to the disjointness requirement and thus such a sign query would not generate any new identical (fractions of polynomials) in groups $\mathbb{G}$ and $\mathbb{H}$ which $\mathcal{A}_{\mathsf{SPS}}$ can feed back as a message into a subsequent sign query, i.e. all subsequent sign queries are on message vectors corresponding to constant polynomials.

We now argue that the additional elements $\mathcal{A}_{\mathsf{SPS}}$ sees could be obtained by $\mathcal{A}_{\mathsf{DLmSPSPS}}$ in her game and hence the former does not have any more advantage over the latter.

- Additional elements of the form $S'_i = M_k^{s_i}$ can be obtained by $\mathcal{A}_{\mathsf{DLmSPSPS}}$ by calling her exponentiation oracle for $\mathbb{G}$ to get $S_i^{m_k}$.
- Additional elements of the form $\tilde{T}'_i = \tilde{M}_k^{t_i}$ can be obtained by $\mathcal{A}_{\mathsf{DLmSPSPS}}$ by calling her exponentiation oracle for $\mathbb{H}$ to get $\tilde{T}_i^{m_k}$.
- Additional elements of the form $S'_j = M_k^{x_i}$ can be obtained by $\mathcal{A}_{\mathsf{DLmSPSPS}}$ by calling her exponentiation oracle for $\mathbb{G}$ to get $X_i^{m_k}$.

Since the SPS forgery must be on a message in $\mathcal{DH}^n$, i.e. the message correspond to a constant polynomials, and $\mathsf{DLmSPSPS}$ is unforgeable, it follows that no linear combinations of the group elements $\mathcal{A}_{\mathsf{SPS}}$ sees in her game leads to a forgery against SPS. □

7 Impossibility Results

In this section we provide some feasibility results for LmSPSPS schemes.

7.1 A Bound on the Number of Signatures for LmSPSPS Schemes

Here we prove, similarly to the case of structure-preserving signatures proven by Abe et al. [7], that a EUF-RMA LmSPSPS scheme must have for each message superpolynomially many potential signatures.

Theorem 5. *An EUF-RMA LmSPSPS scheme (against $q > 1$ sign queries) must have for each message superpolynomially many potential signatures.*

Proof. We can write the j-th signature component of the ℓ-th signing query as:

$$S_j = G^{\frac{\sum_i \boldsymbol{x}^{\boldsymbol{c}_{i,j}} \boldsymbol{y}^{\boldsymbol{c}'_{i,j}} \boldsymbol{r}_\ell^{\boldsymbol{c}''_{i,j}} (a_{i,j} + \sum_{k=1}^{n} d_{i,j,k} m_k)}{\sum_i b_{i,j} \boldsymbol{x}^{\boldsymbol{e}_{i,j}} \boldsymbol{y}^{\boldsymbol{e}'_{i,j}} \boldsymbol{r}_\ell^{\boldsymbol{e}''_{i,j}}}} \quad \text{or} \quad \tilde{T}_j = \tilde{H}^{\frac{\sum_i \boldsymbol{x}^{\boldsymbol{c}_{i,j}} \boldsymbol{y}^{\boldsymbol{c}'_{i,j}} \boldsymbol{r}_\ell^{\boldsymbol{c}''_{i,j}} (a_{i,j} + \sum_{k=1}^{n} d_{i,j,k} m_k)}{\sum_i b_{i,j} \boldsymbol{x}^{\boldsymbol{e}_{i,j}} \boldsymbol{y}^{\boldsymbol{e}'_{i,j}} \boldsymbol{r}_\ell^{\boldsymbol{e}''_{i,j}}}}$$

for some (fixed) $a_{i,j}, b_{i,j}, d_{i,j,k} \in \mathbb{Z}_p$, $\boldsymbol{c}_{i,j}, \boldsymbol{e}_{i,j} \in \mathbb{Z}_p^{\mu}$, $\boldsymbol{c}'_{i,j}, \boldsymbol{e}'_{i,j} \in \mathbb{Z}_p^{\mu'}$, $\boldsymbol{c}''_{i,j}, \boldsymbol{e}''_{i,j} \in \mathbb{Z}_p^{n'}$ which are independent of $\boldsymbol{m}$.

If there are only polynomially many potential signatures for a message vector, there is a polynomial set $\{\boldsymbol{r}_i\}_{i=1}^{\text{poly}(\kappa)}$ from which the randomness vector $\boldsymbol{r}$ is chosen. Thus, with probability $\frac{1}{\text{poly}(\kappa)^2}$ we have that the 2 signatures $\sigma_1 = (\boldsymbol{S}_1, \tilde{\boldsymbol{T}}_1)$ and $\sigma_2 = (\boldsymbol{S}_2, \tilde{\boldsymbol{T}}_2)$ on 2 random messages vectors $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ where $\boldsymbol{m}_1 \neq \boldsymbol{m}_2$, respectively, were produced using the same randomness vector $\boldsymbol{r}_\ell \in \mathbb{Z}_p^{n'}$. Thus, we have that $\sigma^* := \sigma_1^{1-\gamma}\sigma_2^{\gamma}$ is a valid forgery on the new message vector $\boldsymbol{m}^* = (1-\gamma)\boldsymbol{m}_1 + \gamma\boldsymbol{m}_2$ for any $\gamma \in \mathbb{Z}_p^{\times} \backslash \{1\}$ and therefore such a scheme is not EUF-RMA secure. □

7.2 Impossibility of LmSPSPS Schemes with One-Element Signatures

Here we prove that an EUF-RMA (aganist $q > 1$ sign queries) LmSPSPS scheme cannot have one-element signatures. However, as we show in Sect. 5, one-time sEUF-CMA LmSPSPS schemes with one-element signatures are possible.

Theorem 6. *An EUF-RMA LmSPSPS scheme (against $q > 1$ sign queries) must have at least 2 elements in the signature.*

Proof. WLOG, let's assume $\sigma = S \in \mathbb{G}$. The proof for the case where $\sigma = \tilde{T} \in \mathbb{H}$ is similar. Since there is only one unknown in the verification equation, i.e. the signature S, it follows that 1 PPE verification equation is sufficient for such a scheme. Thus, the scheme would have a verification equation of the following form:

$$e(S, \prod_{j=1}^{\mu'} \tilde{Y}_j)^{a_j + \sum_{k=1}^{n} a'_{j,k} m_k} \prod_{i=1}^{\mu} \prod_{j=1}^{\mu'} e(X_i, \tilde{Y}_j)^{d_{i,j} + \sum_{k=1}^{n} d'_{i,j,k} m_k} = Z, \tag{4}$$

where $a_j, a'_{j,k}, d_{i,j}, d'_{i,j,k} \in \mathbb{Z}_p$ and $Z \in \mathbb{T}$ are public constants. By definition, we must have that for all $k \in [n]$ that $a'_{j,k} = 0$.

Given signatures $\sigma_1^* = S_1$ and $\sigma_2^* = S_2$ on random messages $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ where $\boldsymbol{m}_1 \neq \boldsymbol{m}_2$, respectively, we have that $\sigma^* := \sigma_1^{1-\gamma}\sigma_2^{\gamma}$ is a valid forgery on the message $\boldsymbol{m}^* = (1-\gamma)\boldsymbol{m}_1 + \gamma\boldsymbol{m}_2$ for any $\gamma \in \mathbb{Z}_p^{\times}\backslash\{1\}$. Therefore such a scheme is not EUF-RMA secure against an adversary which makes 2 (random-message) sign queries. □

7.3 Lower Bounds for sEUF-CMA LmSPSPS Schemes

The following theorem proves that the signatures of a sEUF-CMA LmSPSPS scheme secure against $q > 1$ sign queries must have bilateral signatures.

Theorem 7. *There is no sEUF-CMA (against $q > 1$ sign queries) LmSPSPS scheme with unilateral signatures.*

Proof. WLOG let's assume that the signature is of the form $\sigma = \boldsymbol{S} \in \mathbb{G}^{\nu}$. The proof for the case where $\sigma = \tilde{\boldsymbol{T}} \in \mathbb{H}^{\nu'}$ is similar. Such a scheme would have a number of PPE verification equations of the following form:

$$\prod_{i=1}^{\nu} e(S_i, \prod_{j=1}^{\mu'} \tilde{Y}_j)^{\rho_{1,i,j,\ell}(\boldsymbol{m})} \prod_{i=1}^{\mu}\prod_{j=1}^{\mu'} e(X_i, \tilde{Y}_j)^{\rho_{2,i,j,\ell}(\boldsymbol{m})} = Z_{\ell}, \quad (5)$$

where $\rho_{1,i,j,\ell}, \rho_{2,i,j,\ell} \in \mathbb{F}_p[M_1, \ldots, M_n]$ are multivariate polynomials and $Z_\ell \in \mathbb{T}$ are some fixed constants.

By definition, the denominator polynomials used in computing the signature components are independent of the message to be signed. Also, since the signature is unilateral, i.e. the signature components only appear on the LHS of the pairings, the numerator polynomials are linear in the randomness vector $\boldsymbol{r}$ whereas the denominator polynomials are independent of the randomness vector. By Theorem 5 such a scheme must have superpolynomially many potential signatures. By querying the sign oracle twice on any message vector $\boldsymbol{m}$ from the message space, with overwhelming probability we obtain 2 distinct signatures $\sigma_1 = \boldsymbol{S}_1$ and $\sigma_2 = \boldsymbol{S}_2$. We have that $\sigma^* = \sigma_1^{1-\gamma}\sigma_2^{\gamma}$ is with overwhelming probability a new signature on $\boldsymbol{m}$ for any $\gamma \in \mathbb{Z}_p^{\times}\backslash\{1\}$. □

The following theorem proves that sEUF-CMA LmSPSPS schemes with 2-element bilateral signatures do not exist. This result holds even without the restriction that the message is linear. This sets a lower bound of 3 bilateral elements for such schemes.

Theorem 8. *There is no sEUF-CMA (against $q > 1$ sign queries) LmSPSPS scheme with 2-element bilateral signatures.*

Proof. The signature is of the form $\sigma = (S, \tilde{T}) \in \mathbb{G} \times \mathbb{H}$ whereas the verification key is of the form $(\boldsymbol{X}, \tilde{\boldsymbol{Y}}) \in \mathbb{G}^{\mu} \times \mathbb{H}^{\mu'}$. As we prove in Lemma 1 in Theorem 9, 1 PPE verification equation is sufficient for a LmSPSPS scheme with 2-element signatures. Therefore, such a scheme would have a PPE verification equation of the following form:

$$e(S, \prod_{j=1}^{\mu'} \tilde{Y}_j)^{a_j} e(\prod_{j=1}^{\mu} X_j, \tilde{T})^{\rho_{2,j}(\boldsymbol{m})} e(S, \tilde{T})^{c} \prod_{i=1}^{\mu} \prod_{j=1}^{\mu'} e(X_i, \tilde{Y}_j)^{\rho_{4,i,j}(\boldsymbol{m})} = Z,$$

where $\rho_{2,j}, \rho_{4,i,j} \in \mathbb{F}_p[M_1, \dots, M_n]$ are multivariate polynomials, and $a_j, c \in \mathbb{Z}_p$, and $Z \in \mathbb{T}$ are some public constants.

If for all $j \in [\mu']$, $a_j = 0$ and $c = 0$, the verification equation is independent of S and hence by Theorem 6 such a scheme is not secure. Thus, we must have that either for some $j \in [\mu']$ that $a_j \neq 0$ or $c \neq 0$, which we consider below:

- Case $a_j \neq 0$ for some $j \in [\mu']$: Given a signature $\sigma = (S, \tilde{T})$ on a random message $\boldsymbol{m}$, we compute a new signature on $\boldsymbol{m}$ as $\sigma^* = (S^*, \tilde{T}^*) := \left(S^{\frac{a_j}{a_j+\gamma c}} \prod_{i=1}^{\mu} X_i^{\frac{-\gamma \rho_{2,i}(\boldsymbol{m})}{a_i+\gamma c}}, \tilde{T}^{\frac{a_j+\gamma c}{a_j}} \prod_{i=1}^{\mu'} Y_i^{\frac{\gamma a_i}{a_j}} \right)$. We have $\sigma^* \neq \sigma$ for any $\gamma \in \mathbb{Z}_p^{\times}$.
- Case $a_j = 0$ for all $j \in [\mu']$ and $c \neq 0$: Given a signature $\sigma = (S, \tilde{T})$ on a random message $\boldsymbol{m}$, we compute a new signature on $\boldsymbol{m}$ as $\sigma^* = (S^*, \tilde{T}^*) := \left(S^{\frac{1}{\gamma}} \prod_{i=1}^{\mu} X_i^{\frac{(1-\gamma)\rho_{2,i}(\boldsymbol{m})}{\gamma c}}, \tilde{T}^{\gamma} \right)$. We have that $\sigma^* \neq \sigma$ for any $\gamma \in \mathbb{Z}_p^{\times} \backslash \{1\}$.

This concludes the proof. □

7.4 Lower Bounds for the Verification Key of Optimal Schemes

We have seen that an optimal (w.r.t. signature size) EUF-RMA LmSP-SPS scheme must have at least 2 elements in the signature. Here we prove that a scheme with ≤ 2 elements in the signature cannot have a verification key consisting of 1 element even for the case when signing single messages, i.e. when $n = 1$. This sets a lower bound of 2 elements in the verification key for even optimal one-time EUF-RMA schemes. Note some of our proofs below assume that the RHS of the PPE equations in Eq. (4) is $Z_\ell = 1_{\mathbb{T}}$.[4]

Theorem 9. *There is no EUF-RMA LmSPSPS scheme (against $q \geq 1$ sign queries) with signatures consisting of ≤ 2 elements and one-element verification key.*

Proof. We start by proving the following lemma regarding the number of PPE verification equations required for schemes with 2-element signatures.

Lemma 1. *One PPE verification equation is sufficient for a LmSPSPS scheme with 2-element signatures.*

[4] Those proofs also hold if the discrete logarithm of Z_ℓ in the case $Z_\ell \neq 1_{\mathbb{T}}$ is known.

Proof. If the scheme has 2 PPE equations, both equations must impose non-trivial constraint on the signature components as otherwise they can be reduced to a single equation. Since each PPE equation must involve at least 1 signature component, we have 3 cases:

- Both equations involve both signature components: This means we have 2 quadratic/linear equations in the discrete logarithm of the signature components. Such an equation system have at most 4 distinct solutions implying that there are at most 4 potential signatures for the message vector which contradicts the proof of Theorem 5.
- One equation involves both components whereas the other equation involves one component: In this case, one equation is quadratic/linear involving both signature components, whereas the remaining equation is linear in one of the components. By substituting the value of the signature component in the linear equation into the other equation we end up with one verification equation that is sufficient for verifying the signature.
- Each verification equation involves a single signature component: Since the other constants (the verification key, the public parameters (if any) and the messages) are fixed, we have that each verification equation is a linear equation in one of the signature components, i.e. each equation is a linear equation in one unknown. Thus, there is exactly 1 potential signature for the message vector which contradicts the proof of Theorem 5. □

The following 4 lemmata complete the proof of the theorem.

Lemma 2. *There is no EUF-RMA SPSPS scheme (against $q \geq 1$ sign queries) with one verification equation and unilateral signatures and a unilateral verification key containing elements from the same source group.*

Proof. Let's consider the case where the signature and the verification key both belong to group $\mathbb{G}$. The proof for the opposite case is similar. The scheme yields a signature $\sigma = (S_1, \ldots, S_\nu) \in \mathbb{G}^\nu$, has a verification key $\mathsf{vk} = (X_1, \ldots, X_\mu) \in \mathbb{G}^\mu$ where WLOG $X_1 = G$, and has a verification equation of the form

$$\prod_{i=1}^{\nu} e(S_i, \tilde{H})^{\rho_{1,i}(m)} \prod_{i=1}^{\mu} e(X_i, \tilde{H})^{\rho_{2,i}(m)} = Z\cdot$$

for some polynomials $\rho_{1,i}$ and $\rho_{2,i}$.

Given a signature $\sigma = (S_1, \ldots, S_\nu)$ on a random message $m \in \mathbb{Z}_p$, we can construct a new forgery $\sigma^* = (S_1^*, \ldots, S_\nu^*)$ on a different message $m^* \neq m$ by fixing some $i \in [\nu]$ and computing let $S_j^* := S_j$ for all $j \in [\nu] \backslash \{i\}$ and $S_i^* := \left(S_i^{\rho_{1,i}(m)} \prod_{j \neq i} S_j^{\rho_{1,j}(m) - \rho_{1,j}(m^*)} \prod_{j=1}^{\mu} X_j^{\rho_{2,i}(m) - \rho_{2,i}(m^*)} \right)^{\frac{1}{\rho_{1,i}(m^*)}}$. It is easy to see that such a forgery is a valid signature on the message m^*. □

Lemma 3. *There is no one-time EUF-RMA LmSPSPS scheme with one verification equation, one-element signatures and one-element verification key.*

Proof. Note here that we assume that $Z = 1_{\mathbb{T}}$. The case where both the signature and verification key lie in the same group follows from Lemma 2. Assume a scheme has a signature $\sigma = S$, a verification key $\mathsf{vk} = \tilde{Y}$ and a verification equation of the following form:

$$e(S, \tilde{H}^{a_1+a_1'm}\tilde{Y}^{a_2+a_2'\ m}) = e(G, \tilde{H}^{b_1+b_1'm}\tilde{Y}^{b_2+b_2'm}).$$

By definition, we must have that $a_1' = a_2' = 0$. Note that we cannot have that $a_1 = a_2 = 0$ as the equation would be independent of the signature, or $b_1' = b_2' = 0$ as the equation would be independent of the message.

Given a signature $\sigma = S$ on a random message m, we construct a forgery on $m^* = \gamma m + \frac{(\gamma-1)(b_1a_2-a_1b_2)}{a_2b_1'-a_1b_2'}$ for any $\gamma \in \mathbb{Z}_p \backslash \{1\}$ as $\sigma* = S^* := G^{\frac{(\gamma-1)(b_1b_2'-b_1'b_2)}{a_2b_1'-a_1b_2'}} S^\gamma$. This is a valid forgery unless $a_2b_1' = a_1b_2'$ which we deal with below:

- Case $a_2b_1' = a_1b_2' \neq 0$ or $b_1' = a_1 = 0$: Given a signature $\sigma = S$ on a random message m, we construct a forgery $\sigma^* = S^* := G^\gamma S_1$ on $m^* = m + \frac{\gamma a_2}{b_2'}$ for any $\gamma \in \mathbb{Z}_p^\times$.
- Case $b_2' = a_2 = 0$: Given a signature $\sigma = S$ on a random message m, we can construct a forgery $\sigma^* = S^* := G^\gamma S_1$ on $m^* = m + \frac{\gamma a_1}{b_1'}$ for any $\gamma \in \mathbb{Z}_p^\times$. □

Lemma 4. *There is no SPSPS scheme with two-element bilateral signatures and one-element verification key that is secure against a key-only attack.*

Proof. Note here that we assume that $Z = 1_{\mathbb{T}}$. The signature is of the form $\sigma = (S, \tilde{T}) \in \mathbb{G} \times \mathbb{H}$ whereas the verification key is either of the form $\tilde{Y} \in \mathbb{H}$ or $X \in \mathbb{G}$. We prove the first case but the proof for the second case is similar. The scheme has a verification equation of the following form:

$$e(S, \tilde{H}^{\rho_1(m)}\tilde{Y}^{\rho_2(m)})e(G, \tilde{T})^{\rho_3(m)}e(S, \tilde{T})^{\rho_4(m)} = e(G, \tilde{H}^{\rho_5(m)}\tilde{Y}^{\rho_6(m)}).$$

for some polynomials ρ_i for $i \in [6]$. Given the verification key, we can construct a forgery on any message $m^* \in \mathbb{Z}_p$ by choosing $\gamma \leftarrow \mathbb{Z}_p^\times$ and computing:

$$\sigma^* = (S^*, \tilde{T}^*) := (G^\gamma, \tilde{H}^{\frac{\rho_5(m^*)-\gamma\rho_1(m^*)}{\rho_3(m^*)+\gamma\rho_4(m^*)}}\tilde{Y}^{\frac{\rho_6(m^*)-\gamma\rho_2(m^*)}{\rho_3(m^*)+\gamma\rho_4(m^*)}}).$$

□

Lemma 5. *There is no one-time EUF-RMA LmSPSPS scheme with two-element unilateral signatures and a verification key consisting of one-element from the opposite source group.*

Proof. Note here that we assume that $Z = 1_{\mathbb{T}}$. Let's consider the case where the signature is of the form $\sigma = (S_1, S_2) \in \mathbb{G}^2$ whereas the verification key is of the form $\tilde{Y} \in \mathbb{H}$. The proof for the opposite case is similar. Such a scheme would have a verification equation of the form

$$\prod_{i=1}^{2} e(S_i, \tilde{H}^{a_{i,1}+a_{i,1}'m}\tilde{Y}^{a_{i,2}+a_{i,2}'m}) = e(G, \tilde{H}^{d_1+d_1'm}\tilde{Y}^{d_2+d_2'm}).$$

By definition, we must have that either $a'_{1,1} = a'_{1,2} = 0$ or $a'_{2,1} = a'_{2,2} = 0$. Let's assume the former case. Note that if $a_{1,1} = a_{1,2} = 0$, the equation is independent of S_1 and hence by Lemma 3 the scheme is not secure against $q \geq 1$ sign queries. Similarly, if $a_{2,1} = a'_{2,1} = a_{2,2} = a'_{2,2} = 0$, the verification equation is independent of S_2 and hence by Lemma 3 it is not secure against $q \geq 1$ sign queries.

Given a signature $\sigma = (S_1, S_2)$ on a random message m, we can construct a new forgery $\sigma = (S_1^*, S_2^*)$ on a new message $m^* \neq m$ by setting $S_1^* := G^{\alpha_{S_1}} S_2^{\beta_{S_1}} S_1$ and $S_2^* := G^{\alpha_{S_2}} S_2^{\beta_{S_2}}$ where

$$\alpha_{S_1} := \frac{\big(d'_2(a_{2,1} + a'_{2,1}m^*) - d'_1(a_{2,2} + a'_{2,2}m^*)\big)(m^* - m)}{a_{1,2}(a_{2,1} + a'_{2,1}m^*) - a_{1,1}(a_{2,2} + a'_{2,2}m^*)}$$

$$\beta_{S_1} := \frac{(a'_{2,1}a_{2,2} - a_{2,1}a'_{2,2})(m^* - m)}{a_{1,2}(a_{2,1} + a'_{2,1}m^*) - a_{1,1}(a_{2,2} + a'_{2,2}m^*)}$$

$$\alpha_{S_2} := \frac{(a_{1,2}d'_1 - a_{1,1}d'_2)(m^* - m)}{a_{1,2}(a_{2,1} + a'_{2,1}m^*) - a_{1,1}(a_{2,2} + a'_{2,2}m^*)}$$

$$\beta_{S_2} := \frac{a_{1,2}(a_{2,1} + a'_{2,1}m) - a_{1,1}(a_{2,2} + a'_{2,2}m)}{a_{1,2}(a_{2,1} + a'_{2,1}m^*) - a_{1,1}(a_{2,2} + a'_{2,2}m^*)}$$

Thus, we can find a forgery unless $a_{1,2}(a_{2,1} + a'_{2,1}m^*) - a_{1,1}(a_{2,2} + a'_{2,2}m^*) = 0$ for all $m^* \in \mathbb{Z}_p$. We have 2 cases to deal with the above as follows:

- Case $a_{1,1} = a_{2,1} = a'_{2,1} = 0$: Note that as stated earlier, if $a_{1,2} = 0$ or $a_{2,2} = a'_{2,2} = 0$, the verification equation is independent of one of the signature components and hence is not secure. We have 2 cases as follows:
  - Case $d_1 \neq 0$: Give a signature $\sigma = (S_1, S_2)$ on a random message $m \in \mathbb{Z}_p$ satisfying $d_1 + d'_1 m \neq 0$, we have that $\sigma^* = (S_1^*, S_2^*)$ where

    $$S_1^* := G^{\frac{-\gamma a_{2,2}(d_1 + d'_1 m^*)}{a_{1,2}d_1}} S_1^{\frac{d_1 + d'_1 m^*}{d_1 + d'_1 m}} \qquad S_2^* := G^{\gamma} S_2$$

    is a valid forgery on any $m^* \neq m$ satisfying $d_1 + d'_1 m^* \neq 0$ for any $\gamma \in \mathbb{Z}_p^{\times}$.
  - Case $d_1 = 0$: Given a signature $\sigma = (S_1, S_2)$ on a random message $m \in \mathbb{Z}_p^{\times}$, we have that $\sigma^* = (S_1^*, S_2^*)$ where

    $$S_1^* := G^{\frac{d_2(m - m^*) - \gamma m(a_{2,2} + a'_{2,2}m^*)}{a_{1,2}m}} S_2^{\frac{a_{2,2}(m^* - m)}{a_{1,2}m}} S_1^{\frac{m^*}{m}} \qquad S_2^* := G^{\gamma} S_2$$

    is a valid forgery on the message $m^* \neq m$ for any $\gamma \in \mathbb{Z}_p^{\times}$.
- Case $a_{2,2}a_{1,1} = a_{1,2}a_{2,1}$, $a'_{2,2}a_{1,1} = a_{1,2}a'_{2,1}$ and $a_{1,1} \neq 0$: If $a_{1,2} = 0$, we have $a_{2,2} = a'_{2,2} = 0$ and hence we cannot have any of the following cases:
  - $d_2 = d'_2 = 0$: Since verification would be independent of the key $\tilde{Y}$.
  - $a_{2,1} = a'_{2,1} = 0$: Since verification would be independent of S_2.
  - $a'_{2,1} = d'_1 = d'_2 = 0$: Since verification would be independent of m.

  We have that $\sigma^* = (S_1^*, S_2^*)$ where

  $$S_1^* := G^{\frac{a_{1,2}\gamma(a_{2,1}d'_1 - a'_{2,1}d_1) + a_{1,1}\left(d_2(a'_{2,1}\gamma - d'_1) + d'_2(d_1 - a_{2,1}\gamma)\right)}{a_{1,1}(a_{1,1}d'_2 - a_{1,2}d'_1)}} \qquad S_2^* := G^{\gamma}$$

is a valid forgery on $m^* := \frac{a_{1,1}d_2 - a_{1,2}d_1}{a_{1,2}d'_1 - a_{1,1}d'_2}$. The forgery is valid unless $a_{1,2}d'_1 - a_{1,1}d'_2 = 0$. We have 2 cases to deal with this as follows:

- Case $a_{1,2}d'_1 = a_{1,1}d'_2 = 0$: Given a signature $\sigma = (S_1, S_2)$ on a random message $m \in \mathbb{Z}_p$, we have that $\sigma^* = (S_1^*, S_2^*)$ where

$$S_1^* := G^{\frac{-\gamma(a_{2,1}+a'_{2,1}m^*)+d'_1(m^*-m)}{a_{1,1}}} S_2^{\frac{a_{2,1}+a'_{2,1}m}{a1,1}} S_1 \qquad S_2^* := G^{\gamma}$$

  is a valid forgery on any $m^* \neq m$ for any $\gamma \in \mathbb{Z}_p^{\times}$.
- Case $a_{1,2}d'_1 = a_{1,1}d'_2 \neq 0$: Given a signature $\sigma = (S_1, S_2)$ on a random message $m \in \mathbb{Z}_p$, we have that $\sigma^* = (S_1^*, S_2^*)$ where

$$S_1^* := G^{\frac{-\gamma a_{1,2}(a_{2,1}+a'_{2,1}m^*)+d'_2 a_{1,1}(m^*-m)}{a_{1,1}a_{1,2}}} S_2^{\frac{a_{2,1}+a'_{2,1}m}{a_{1,1}}} S_1 \qquad S_2^* := G^{\gamma}$$

  is a valid forgery on any $m^* \neq m$ for any $\gamma \in \mathbb{Z}_p^{\times}$.

If it is required that $S_i^* \in \mathbb{G}^{\times}$, we have to additionally handle the case that $d_1 a'_{2,1} = a_{2,1}d'_1$ and $d_2 a'_{2,1} = a_{2,1}d'_2$. Note that we cannot have that $a'_{2,1} = 0$ as otherwise the signature will either be independent of S_2 or m. We have that given a signature $\sigma = (S_1, S_2)$ on a random message $m \in \mathbb{Z}_p^{\times}$, we have that $\sigma^* = (S_1^*, S_2^*) := (S_1^{\gamma}, S_2)$ is a valid forgery on any message $m^* = \frac{a_{2,1}(\gamma-1)+a'_{2,1}\gamma m}{a'_{2,1}}$ for any $\gamma \in \mathbb{Z}_p^{\times} \backslash \{1\}$. □

This concludes the proof. □

We have proved that an (optimal) scheme with two-element unilateral signatures must have at least 2 elements in the verification key besides the default source group generators. An intriguing question is whether, similarly to the one-time EUF-CMA scheme we give in Sect. 5, an EUF-RMA scheme with two-element unilateral signatures and a two-element bilateral verification key exists. We answer this question negatively by proving the following theorem. In essence, this means the PS scheme and our new *LmSPSPS* scheme are optimal w.r.t. the size of the verification key.

Theorem 10. *There is no EUF-RMA (against $q > 1$ sign queries) LmSPSPS scheme with two-element unilateral signatures and a two-element bilateral verification key.*

Proof. Let's consider a scheme with signatures of the form $\sigma = (S_1, S_2) \in \mathbb{G}^2$ whereas the verification key is of the form $(X, \tilde{Y}) \in \mathbb{G} \times \mathbb{H}$. The proof for the opposite case is similar.

Such a scheme has a PPE verification equation of the form

$$\prod_{i=1}^{2} e(S_i, \tilde{H}^{a_{i,1}+a'_{i,1}m}\tilde{Y}^{a_{i,2}+a'_{i,2}m})$$
$$= e(G^{d_{1,1}+d'_{1,1}m}X^{d_{2,1}+d'_{2,1}m}, \tilde{H})e(G^{d_{1,2}+d'_{1,2}m}X^{d_{2,2}+d'_{2,2}m}, \tilde{Y}).$$

By definition, we must have that either $a'_{1,1} = a'_{1,2} = 0$ or $a'_{2,1} = a'_{2,2} = 0$ as otherwise the message features in the denominator polynomial of a signature component. Let's assume WLOG that $a'_{1,1} = a'_{1,2} = 0$ as the other case is similar.

Such a scheme is not secure against an adversary that receives two signatures $\sigma_1 = (S_{1,1}, S_{1,2})$ and $\sigma_2 = (S_{2,1}, S_{2,2})$ on two random distinct messages m_1 and m_2, respectively. We can construct a forgery on a new message $m^* \notin \{m_1, m_2\}$ as follows:

$$\text{Define } A_1 = \begin{bmatrix} a_{2,1} & a_{1,1} \\ a_{2,2} & a_{1,2} \end{bmatrix}, A_2 = \begin{bmatrix} a'_{2,1} & a_{1,1} \\ a'_{2,2} & a_{1,2} \end{bmatrix} \text{ and } A_3 = \begin{bmatrix} a_{2,1} & a'_{2,1} \\ a_{2,2} & a'_{2,2} \end{bmatrix}$$

Let $\alpha := \frac{(|A_1|+|A_2|m_1)(m^*-m_2)}{(|A_1|+|A_2|m^*)(m_1-m_2)}$ and

$$\beta_{s_{1,1}} := \frac{m_2 - m^*}{m_2 - m_1} \qquad \beta_{s_{1,2}} := \frac{m_1 - m^*}{m_1 - m_2}$$

$$\gamma_{s_{1,1}} := \frac{|A_3|(m^* - m_2 + (m_2 - m_1)\alpha)}{|A_2|(m_2 - m_1)} \qquad \gamma_{s_{1,2}} := -\gamma_{s_{1,1}}$$

$$\gamma_{s_{2,1}} := \alpha \qquad \gamma_{s_{2,2}} := -\frac{(|A_1| + |A_2|m_2)(m^* - m_1)}{(|A_1| + |A_2|m^*)(m_1 - m_2)}$$

We have that $\sigma^* = (S_1^* := S_{1,1}^{\beta_{s_{1,1}}} S_{2,1}^{\beta_{s_{1,2}}} S_{1,2}^{\gamma_{s_{1,1}}} S_{2,2}^{\gamma_{s_{1,2}}}, S_2^* := S_{1,2}^{\gamma_{s_{2,1}}} S_{2,2}^{\gamma_{s_{2,2}}})$ is a valid forgery on any message $m^* \in \mathbb{Z}_p \backslash \{m_1, m_2, \frac{-|A_1|}{|A_2|}\}$ satisfying $|A_1| + |A_2|m^* \neq 0$. Thus, we obtain a forgery on a new message unless $|A_2| = 0$ which is dealt with by the following 3 cases:

- Case $a_{1,1} = 0$: We have 2 cases:

  ◦ Case $a_{1,2} = 0$: The verification equation is independent of the signature component S_1 and hence is not secure.

  ◦ Case $a'_{2,1} = 0$: Given signatures $\sigma_1 = (S_{1,1}, S_{1,2})$ and $\sigma_2 = (S_{2,1}, S_{2,2})$ on random messages m_1 and m_2, respectively, we have that $\sigma^* = (S_1^*, S_2^*)$ where

  $$S_1^* := S_{1,1}^{\gamma} S_{2,1}^{1-\gamma} S_{1,2}^{-\frac{a'_{2,2}(\gamma^2-\gamma)(m_1-m_2)}{a_{1,2}}} S_{2,2}^{\frac{a'_{2,2}(\gamma^2-\gamma)(m_1-m_2)}{a_{1,2}}} \qquad S_2^* := S_{1,2}^{\gamma} S_{2,2}^{1-\gamma}$$

  is a valid forgery on $m^* = \gamma m_1 + (1 - \gamma)m_2$ for any $\gamma \in \mathbb{Z}_p^{\times} \backslash \{1\}$.

- Case $a'_{2,2} = 0$ and $a_{1,1} \neq 0$: Given signatures $\sigma_1 = (S_{1,1}, S_{1,2})$ and $\sigma_2 = (S_{2,1}, S_{2,2})$ on two random messages m_1 and m_2, respectively, we compute

  $$S_1^* := S_{1,1}^{\gamma} S_{2,1}^{1-\gamma} S_{1,2}^{-\frac{a'_{2,1}(\gamma^2-\gamma)(m_1-m_2)}{a_{1,1}}} S_{2,2}^{\frac{a'_{2,1}(\gamma^2-\gamma)(m_1-m_2)}{a_{1,1}}} \qquad S_2^* := S_{1,2}^{\gamma} S_{2,2}^{1-\gamma}$$

  We have that $\sigma^* = (S_1^*, S_2^*)$ is a valid forgery on $m^* = \gamma m_1 + (1 - \gamma)m_2$ for any $\gamma \in \mathbb{Z}_p^{\times} \backslash \{1\}$.

- Case $a'_{2,2}a_{1,1} = a_{1,2}a'_{2,1} \neq 0$: Given signatures $\sigma_1 = (S_{1,1}, S_{1,2})$ and $\sigma_2 = (S_{2,1}, S_{2,2})$ on two distinct random messages m_1 and m_2, respectively, we compute

$$S_1^* := S_{1,1}^{\frac{m_2-m^*}{m_2-m_1}} S_{2,1}^{\frac{m_1-m^*}{m_1-m_2}} S_{1,2}^{-\frac{a'_{2,1}(m^*-m_1)(m^*-m_2)}{a_{1,1}(m_1-m_2)}} S_{2,2}^{\frac{a'_{2,1}(m^*-m_1)(m^*-m_2)}{a_{1,1}(m_1-m_2)}}$$
$$S_2^* := S_{1,2}^{\frac{m_2-m^*}{m_2-m_1}} S_{2,2}^{\frac{m_1-m^*}{m_1-m_2}}$$

We have that $\sigma^* = (S_1^*, S_2^*)$ is a valid forgery on any new message $m^* \in \mathbb{Z}_p \backslash \{m_1, m_2\}$.

This concludes the proof. □

References

1. Abe, M., Ambrona, M., Ohkubo, M., Tibouchi, M.: Lower bounds on structure-preserving signatures for bilateral messages. In: Catalano, D., De Prisco, R. (eds.) SCN 2018. LNCS, vol. 11035, pp. 3–22. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98113-0_1
2. Abe, M., Chase, M., David, B., Kohlweiss, M., Nishimaki, R., Ohkubo, M.: Constant-size structure-preserving signatures: generic constructions and simple assumptions. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 4–24. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34961-4_3
3. Abe, M., David, B., Kohlweiss, M., Nishimaki, R., Ohkubo, M.: Tagged one-time signatures: tight security and optimal tag size. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 312–331. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_20
4. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 209–236. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_12
5. Abe, M., Groth, J., Haralambiev, K., Ohkubo, M.: Optimal structure-preserving signatures in asymmetric bilinear groups. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 649–666. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_37
6. Abe, M., Groth, J., Ohkubo, M.: Separating short structure-preserving signatures from non-interactive assumptions. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 628–646. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_34
7. Abe, M., Groth, J., Ohkubo, M., Tibouchi, M.: Unified, minimal and selectively randomizable structure-preserving signatures. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 688–712. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_29
8. Abe, M., Hofheinz, D., Nishimaki, R., Ohkubo, M., Pan, J.: Compact structure-preserving signatures with almost tight security. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10402, pp. 548–580. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_19

9. Bellare, M., Garay, J.A., Rabin, T.: Fast batch verification for modular exponentiation and digital signatures. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 236–250. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054130
10. Bichsel, P., Camenisch, J., Neven, G., Smart, N.P., Warinschi, B.: Get shorty via group signatures without encryption. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 381–398. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15317-4_24
11. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_4
12. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. J. Cryptol. **17**, 297–319 (2004). https://doi.org/10.1007/s00145-004-0314-9
13. Brickell, E., Camenisch, J., Chen, L.: Direct anonymous attestation. In: ACM CCS. ACM (2004)
14. Camenisch, J., Dubovitskaya, M., Haralambiev, K.: Efficient structure-preserving signature scheme from standard assumptions. In: Visconti, I., De Prisco, R. (eds.) SCN 2012. LNCS, vol. 7485, pp. 76–94. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32928-9_5
15. Camenisch, J., Dubovitskaya, M., Haralambiev, K., Kohlweiss, M.: Composable and modular anonymous credentials: definitions and practical constructions. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9453, pp. 262–288. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48800-3_11
16. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28628-8_4
17. Chase, M., Kohlweiss, M.: A new hash-and-sign approach and structure-preserving signatures from DLIN. In: Visconti, I., De Prisco, R. (eds.) SCN 2012. LNCS, vol. 7485, pp. 131–148. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32928-9_8
18. Chatterjee, S., Kabaleeshwaran, R.: Rerandomizable signatures under standard assumption. In: Hao, F., Ruj, S., Sen Gupta, S. (eds.) INDOCRYPT 2019. LNCS, vol. 11898, pp. 45–67. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35423-7_3
19. Chatterjee, S., Menezes, A.: Type 2 structure-preserving signature schemes revisited. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 286–310. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_13
20. Chen, L., Page, D., Smart, N.P.: On the design and implementation of an efficient DAA scheme. In: Gollmann, D., Lanet, J.-L., Iguchi-Cartigny, J. (eds.) CARDIS 2010. LNCS, vol. 6035, pp. 223–237. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12510-2_16
21. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory **31**(4), 469–472 (1985)
22. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12
23. Fuchsbauer, G.: Commuting signatures and verifiable encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 224–245. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_14

24. Fuchsbauer, G., Hanser, C., Slamanig, D.: Practical round-optimal blind signatures in the standard model. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 233–253. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_12
25. Galbraith, S., Paterson, K., Smart, N.P.: Pairings for cryptographers. Discrete Appl. Math. **156**, 3113–3121 (2008)
26. Gay, R., Hofheinz, D., Kohl, L., Pan, J.: More efficient (almost) tightly secure structure-preserving signatures. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10821, pp. 230–258. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78375-8_8
27. Gerbush, M., Lewko, A., O'Neill, A., Waters, B.: Dual form signatures: an approach for proving security from static assumptions. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 25–42. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34961-4_4
28. Ghadafi, E.: Formalizing group blind signatures and practical constructions without random oracles. In: Boyd, C., Simpson, L. (eds.) ACISP 2013. LNCS, vol. 7959, pp. 330–346. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39059-3_23
29. Ghadafi, E.: Short structure-preserving signatures. In: Sako, K. (ed.) CT-RSA 2016. LNCS, vol. 9610, pp. 305–321. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29485-8_18
30. Ghadafi, E.: Efficient round-optimal blind signatures in the standard model. In: Kiayias, A. (ed.) FC 2017. LNCS, vol. 10322, pp. 455–473. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70972-7_26
31. Ghadafi, E.: More efficient structure-preserving signatures - or: bypassing the type-III lower bounds. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) ESORICS 2017. LNCS, vol. 10493, pp. 43–61. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66399-9_3
32. Ghadafi, E.: How low can you go? Short structure-preserving signatures for Diffie-Hellman vectors. In: O'Neill, M. (ed.) IMACC 2017. LNCS, vol. 10655, pp. 185–204. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71045-7_10
33. Ghadafi, E.: Further lower bounds for structure-preserving signatures in asymmetric bilinear groups. In: Buchmann, J., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 2019. LNCS, vol. 11627, pp. 409–428. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23696-0_21
34. Groth, J.: Efficient fully structure-preserving signatures for large messages. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 239–259. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_11
35. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. SIAM J. Comput. **41**(5), 1193–1232 (2012)
36. Jutla, C.S., Roy, A.: Improved structure preserving signatures under standard bilinear assumptions. In: Fehr, S. (ed.) PKC 2017. LNCS, vol. 10175, pp. 183–209. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54388-7_7
37. Kiltz, E., Pan, J., Wee, H.: Structure-preserving signatures from standard assumptions, revisited. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 275–295. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_14
38. Libert, B., Peters, T., Yung, M.: Short group signatures via structure-preserving signatures: standard model security from simple assumptions. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 296–316. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_15

39. Maji, H.K., Prabhakaran, M., Rosulek, M.: Attribute-based signatures. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 376–392. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19074-2_24
40. Maurer, U.: Abstract models of computation in cryptography. In: Smart, N.P. (ed.) Cryptography and Coding 2005. LNCS, vol. 3796, pp. 1–12. Springer, Heidelberg (2005). https://doi.org/10.1007/11586821_1
41. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_9
42. Pointcheval, D., Sanders, O.: Short randomizable signatures. In: Sako, K. (ed.) CT-RSA 2016. LNCS, vol. 9610, pp. 111–126. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29485-8_7
43. Pointcheval, D., Sanders, O.: Reassessing security of randomizable signatures. In: Smart, N.P. (ed.) CT-RSA 2018. LNCS, vol. 10808, pp. 319–338. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76953-0_17
44. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-69053-0_18
45. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_7