More Efficient Structure-Preserving Signatures - Or: Bypassing the Type-III Lower Bounds

Essam Ghadafi*

University College London, UK
e.ghadafi@ucl.ac.uk

Abstract. Structure-preserving signatures are an important cryptographic primitive that is useful for the design of modular cryptographic protocols. It has been proven that structure-preserving signatures (in the most efficient Type-III bilinear group setting) have a lower bound of 3 group elements in the signature (which must include elements from both source groups) and require at least 2 pairing-product equations for verification. In this paper, we show that such lower bounds can be circumvented. In particular, we define the notion of Unilateral Structure-Preserving Signatures on Diffie-Hellman pairs (USPSDH) which are structure-preserving signatures in the efficient Type-III bilinear group setting with the message space being the set of Diffie-Hellman pairs, in the terminology of Abe et al. (Crypto 2010). The signatures in these schemes are elements of one of the source groups, i.e. unilateral, whereas the verification key elements' are from the other source group. We construct a number of new structure-preserving signature schemes which bypass the Type-III lower bounds and hence they are much more efficient than all existing structure-preserving signature schemes. We also prove optimality of our constructions by proving lower bounds and giving some impossibility results. Our contribution can be summarized as follows:

- We construct two optimal randomizable CMA-secure schemes with signatures consisting of only 2 group elements from the first short source group and therefore our signatures are at least half the size of the best existing structure-preserving scheme for unilateral messages in the (most efficient) Type-III setting. Verifying signatures in our schemes requires, besides checking the well-formedness of the message, the evaluation of a single Pairing-Product Equation (PPE) and requires a fewer pairing evaluations than all existing structure-preserving signature schemes in the Type-III setting. Our first scheme has a feature that permits controlled randomizability (combined unforgeability) where the signer can restrict some messages such that signatures on those cannot be re-randomized which might be useful for some applications.
- We construct optimal strongly unforgeable CMA-secure one-time schemes with signatures consisting of 1 group element, and which can also sign a vector of messages while maintaining the same signature size.
- We give a one-time strongly unforgeable CMA-secure structure-preserving scheme that signs unilateral messages, i.e. messages in one of the source groups, whose efficiency matches the best existing optimal one-time scheme in every respect.
- We investigate some lower bounds and prove some impossibility results regarding this variant of structure-preserving signatures.
- We give an optimal (with signatures consisting of 2 group elements and verification requiring 1 pairing-product equation) fully randomizable CMA-secure partially structure-preserving scheme that simultaneously signs a Diffie-Hellman pair and a vector in $\mathbb{Z}_p^k$.
- As an example application of one of our schemes, we obtain efficient instantiations of randomizable weakly blind signatures which do not rely on random oracles. The latter is a building block that is used, for instance, in constructing Direct Anonymous Attestation (DAA) protocols, which are protocols deployed in practice.

Our results offer value along two fronts: On the practical side, our constructions are more efficient than existing ones and thus could lead to more efficient instantiations of many cryptographic protocols. On the theoretical side, our results serve as a proof that many of the lower bounds for the Type-III setting can be circumvented.

Keywords. Structure-Preserving, Digital Signatures, Type-III Bilinear Groups, Lower Bounds.

*The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 307937 and EPSRC grant EP/J009520/1.

1 Introduction

Structure-Preserving Signatures (SPS) [3] are pairing-based digital signature schemes whose messages, verification key and signatures are all group elements and signature verification involves evaluating Pairing-Product Equations (PPE). Such schemes compose nicely with existing popular tools such as Groth-Sahai proofs [34] and ElGamal encryption scheme [23] and hence they are a useful tool for the design of cryptographic protocols which do not rely on heuristic assumptions such as random oracles [24]. They have numerous applications which include group signatures, e.g. [3, 37, 38], blind signatures, e.g. [3, 26], tightly secure encryption schemes, e.g. [35, 2], malleable signatures, e.g. [9], anonymous credentials, e.g. [26], network coding, e.g. [9], and oblivious transfer, e.g. [31].

Related Work. The term "structure-preserving signature" was put forward by Abe et al. [3] but earlier schemes conforming to the definition were given by Groth [32] and Green and Hohenberger [31]. The notion has received a significant amount of attention from the cryptographic community and many results regarding proving lower bounds for the design of such schemes as well as new schemes meeting those lower bounds have been published in the literature. Abe et al. [3] constructed structure-preserving signature schemes based on non-interactive intractability assumptions which work in the different bilinear group settings. Abe et al. [4] showed that a signature of a structure-preserving scheme in the Type-III bilinear group setting (cf. Section 2.1) must have at least 3 group elements and require at least 2 pairing-product equations to be verified. They also proved that the signature must contain elements from both source groups which rules out the existence of unilateral signatures (i.e. signatures whose all components are elements of one of the source groups). They gave optimal constructions and proved their security in the generic group model [42, 40]. Abe et al. [5] proved that it is impossible to base the security of a scheme with signatures consisting of 3 group elements in the Type-III setting on non-interactive intractability assumptions. In essence, their result proves that in the Type-III setting, the only way to meet the 3 group element lower bound is to either employ interactive intractability assumptions or resort to direct proofs in the generic group model. Ghadafi [28] gave a structure-preserving variant of the Camenisch-Lysyanskaya signature scheme [17] in the Type-III setting that is based on an interactive assumption. Abe et al. [7] gave a scheme in the Type-II setting (where there is an efficiently computable isomorphism from the second source group to the first) with signatures consisting of only 2 group elements. Chatterjee and Menezes [20] revisited the work of [7] and showed that Type-III constructions outperform their Type-II counterparts. They also gave constructions in Type-III setting meeting the 3 group element lower bound. Barthe et al. [10] gave optimal constructions of structure-preserving signatures in Type-II setting. Constructions relying on standard assumptions (such as DLIN and DDH) were given by [18, 1, 16, 2, 36, 38]. It is well known that schemes based on standard assumptions are much less efficient than their counterparts relying on non-standard assumptions or those proven directly in the generic group model. Recently, Ghadafi [29] gave a randomizable scheme with signatures consisting of 3 elements from the first source group which can also be regarded as a unilateral structure-preserving signature scheme on Diffie-Hellman pairs. Verification in his scheme requires, besides checking the well-formedness of the message, the evaluation of 2 pairing-product equations. Abe et al. [8] and Groth [33] recently gave fully structure-preserving schemes where even the secret key consists of only group elements.

Our Contribution. After defining unilateral structure-preserving signatures on Diffie-Hellman pairs, our contribution can be summarized as follows:-

- We construct two new randomizable structure-preserving signature schemes that are existentially unforgeable against a chosen message attack. Our schemes yield signatures consisting of only two group elements from the first short source group and hence our signatures are at least half the size of the shortest existing Type-III structure-preserving signature scheme. Our schemes also outperform the very recent scheme in [29]. Verifying signatures in our schemes requires, besides checking the well-formedness of the message, the evaluation of a single pairing-product equation. The total number of pairings required for verification in our schemes are 4 (1 of which is offline, i.e. can be precomputed) and 3, respectively. In both schemes, depending on the application, the number of pairing evaluations can be reduced by 1 since in both schemes two pairings in the equation share the same left-hand side argument. Our first construction has a feature that permits controlled randomizability (combined unforgeability) which might be of independent interest.

- We give a strongly unforgeable CMA-secure one-time USPSDH scheme with 1 element signatures. We also give different variants which sign vectors of messages while maintaining the same signature size.
- We give a strongly unforgeable one-time CMA-secure scheme for unilateral messages in the Type-III setting that matches the best existing optimal scheme in every respect.
- We investigate some lower bounds and prove some impossibility results for USPSDH schemes. Our (in)feasibility and lower bound results include the following:
  i) The impossibility of strongly existentially unforgeable schemes that are secure against an adversary that makes more than a single signing query. This implies that only one-time USPSDH schemes can have strong existential unforgeability against a chosen message attack.
  ii) A lower bound of 2 group element signatures for schemes that are secure against a random message attack for more than a single signing query. In essence, this means that all of our constructions are optimal.
  iii) A lower bound of 2 group elements for the verification key of optimal schemes. This applies even when the adversary is restricted to a single random message signing query. In essence, this means that our constructions are optimal in every respect.
- We give an optimal fully randomizable CMA-secure partially structure-preserving scheme that simultaneously signs a Diffie-Hellman pair and a vector in $\mathbb{Z}_p^k$.
- As a by-product, we give efficient instantiations of randomizable weakly blind signatures [12] which do not rely on random oracles and which are more efficient than existing constructions. The latter is a building block that is used, for instance, in the design of direct anonymous attestation protocols [15, 12].

Why are USPSDH schemes interesting? From our results, it is clear that USPSDH signature schemes outperform other variants of structure-preserving signatures since they yield shorter signatures and require less verification overhead since as we show, they circumvent the lower bounds in the Type-III setting. It is particularly interesting when the signatures are from the first short source group as the bit size of the elements of that group is at least half the size of those of the second source group. Note that all existing structure-preserving signatures for unilateral messages require a minimum of 3 group elements in the signature one of which at least must be from the second source group. While traditional structure-preserving signatures (on unilateral messages), those in Type-III in particular, have shorter messages, since message components of those schemes lie in one of the source groups and not in both, this is a small price to pay to get smaller signatures and more efficient verification. We stress that the size of messages in USPSDH schemes is still much shorter than schemes in the Type-II setting and those in Type-I based on finite fields of large characteristics. The latter is recommended as a replacement to bilinear groups based on finite fields of small characteristics following the recent advancement, e.g. [13, 30], in solving discrete logarithm in the latter setting.

Note that even though one needs to check the well-formedness of the message when verifying a USPSDH signature, such a check only needs to be performed once when verifying multiple signatures on the same message. Consider, for example, attribute-based signatures [39] where the signer needs to prove that she has multiple attributes from (possibly different) attribute authorities. The same applies to applications requiring a user to prove that she has multiple tokens/credentials/certificates from an authority or possibly different authorities.

In addition, such schemes work well in association with the popular (but less efficient) automorphic structure-preserving signature scheme of Abe et al. [25, 3] (whose message and verification key spaces lie in the message space of USPSDH schemes). The Abe et al. automorphic scheme [25, 3] has been used in constructing many cryptographic protocols, which include group signatures [22], anonymous credentials [25], and e-cash systems. Therefore, USPSDH schemes could lead to more practical instantiations of many cryptographic protocols, including direct anonymous attestation [15], which is a protocol deployed in practice.

Consider, for instance, an application where the user needs to prove (using the Groth-Sahai proof system [34]) possession of n signatures on some message (e.g. her verification key/identity/pseudonym) possibly from different signers. Since the best existing Type-III scheme requires at least 2 PPE equations to verify each individual signature, this would incur a total cost of $2n$ Groth-Sahai proofs. On the other hand, using, for example, any of our optimal USPSDH schemes, one would only need $n + 1$ Groth-Sahai proofs which is significantly better. Also, signatures of our schemes consist of only two group elements from the first short source group.

We compare in Table 1, the efficiency of our two new CMA secure schemes with existing schemes in the Type-III setting. In the last column of the table, we give two different estimations (separated by the word "OR") for the total number of pairings required for verification. The first estimation (which precedes the word "OR") combines pairings which share an input, i.e. collecting like terms, (which serves to reduce the number of pairings,) whereas the second estimation counts the pairings separately. Numbers superscripted with † are the number of pairings that can be precomputed. Since the well-formedness of the message only needs to be verified once when verifying multiple signatures on the same message, we do not count such cost for schemes whose message space is $\widehat{\mathbb{GH}}$, i.e. the set of Diffie-Hellman pairs, refer to Section 2.1. For all schemes listed, public parameters do not include the default group generators G and $\tilde{H}$.

We remark that our schemes even compete with standard non-structure-preserving signatures. For instance, our schemes are more efficient than the Camenisch-Lysyanskaya signature scheme [17] and Waters' scheme [43] in the Type-III setting [19]. Also, the size of our signatures and the verification key are the same as those of the recent (non-structure-preserving) scheme by Pointcheval and Sanders [41].

Scheme	σ		vk		Param		m	Randomize?	Assumptions	#PPE	#Pairings
	$\mathbb{G}$	$\mathbb{H}$	$\mathbb{G}$	$\mathbb{H}$	$\mathbb{G}$	$\mathbb{H}$					
[25]	3	2	1	1	3	1	$\widehat{\mathbb{GH}}$	No	q-ADHSDH + AWFCDH	3	7 OR $8+1^\dagger$
[3] I	5	2	10	4	-	-	$\mathbb{G}$	Partially	q-SFP	2	$8+4^\dagger$
[3] II	2	5	10	4	-	-	$\mathbb{H}$	Partially	q-SFP	2	$8+4^\dagger$
[4] I	2	1	1	3	-	-	$\mathbb{G}\times\mathbb{H}$	No	GGM	2	$5+2^\dagger$
[4] II	2	1	1	1	-	-	$\mathbb{H}$	Yes	GGM	2	$4+1^\dagger$
[28]	4	-	-	2	-	-	$\widehat{\mathbb{GH}}$	Yes	DH-LRSW	3	6 OR 7
[20] I	1	2	2	-	-	-	$\mathbb{H}$	No	GGM	2	$4+1^\dagger$
[20] II	1	2	2	-	-	-	$\mathbb{H}$	Yes	GGM	2	$5+1^\dagger$
[20] III	2	1	-	2	-	-	$\mathbb{G}$	Yes	GGM	2	$5+1^\dagger$
[6] I	3	1	-	1	1	-	$\mathbb{G}$	Yes	GGM	2	$4+2^\dagger$
[6] II	2	1	-	1	1	-	$\mathbb{G}$	No	GGM	2	$4+2^\dagger$
[10]	1	2	2	-	-	-	$\mathbb{H}$	Yes	GGM	2	$3+2^\dagger$
[33] I	1	2	1	-	-	1	$\mathbb{H}$	Yes	GGM	2	$3+3^\dagger$
[33] II	1	2	1	-	-	1	$\mathbb{H}$	No	GGM	2	$4+3^\dagger$
[29]	3	-	-	2	-	-	$\widehat{\mathbb{GH}}$	Yes	GGM	2	5
Ours I	2	-	-	2	-	-	$\widehat{\mathbb{GH}}$	Yes[1]	GGM	1	$2+1^\dagger$ OR $3+1^\dagger$
Ours II	2	-	-	2	-	-	$\widehat{\mathbb{GH}}$	Yes	GGM	1	2 OR 3

Table 1. Efficiency comparison between our optimal CMA secure schemes and existing schemes in the Type-III setting

Paper Organization. In Section 2, we give some preliminary definitions. In Section 3, we define unilateral structure-preserving signatures on Diffie-Hellman pairs. In Sections 4 & 5, we present constructions of optimal signature schemes and prove their security. In Sections 6 & 7, we present constructions of optimal one-time signature schemes and prove their security. In Section 8, we prove some lower bounds and give some impossibility results. In Section 9, we give an optimal CMA-secure partially structure-preserving scheme that simultaneously signs a Diffie-Hellman pair and a vector in $\mathbb{Z}_p^k$. We give some example applications of our schemes in Section 10.

Notation. We write $y = A(x;r)$ when the 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. By $[k]$, we denote the set $\{1,\ldots,k\}$. We will use capital letters for group elements and small letters for field elements.

[1] Randomization requires possession of at least 2 distinct signatures on the message in question.

2 Preliminaries

In this section we provide some preliminary definitions.

2.1 Bilinear Groups

A bilinear group is a tuple $\mathcal{P} := (\mathbb{G}, \mathbb{H}, \mathbb{T}, p, G, \tilde{H}, \hat{e})$ 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 $\hat{e}$ is a non-degenerate bilinear map $\hat{e} : \mathbb{G} \times \mathbb{H} \longrightarrow \mathbb{T}$. For clarity, elements of $\mathbb{H}$ will be accented with $\tilde{}$. We use multiplicative notation for all the groups. We let $\mathbb{G}^\times := \mathbb{G} \setminus \{1_{\mathbb{G}}\}$ and $\mathbb{H}^\times := \mathbb{H} \setminus \{1_{\mathbb{H}}\}$. In this paper, we work in the efficient Type-III setting [27], where $\mathbb{G} \neq \mathbb{H}$ and there is no efficiently computable isomorphism between the groups in either direction. We assume there is an algorithm $\mathsf{BGSetup}$ that on input a security parameter λ, outputs a description of bilinear groups.

The message space of the signature schemes we consider is the set of elements of the subgroup $\widehat{\mathbb{GH}}$ of $\mathbb{G} \times \mathbb{H}$ defined as the image of the map

$$\psi : \begin{cases} \mathbb{Z}_p \longrightarrow \quad \mathbb{G} \times \mathbb{H} \\ \;\; x \;\longmapsto (G^x, \tilde{H}^x) \end{cases}$$

Given an element $(M, \tilde{N}) \in \mathbb{G} \times \mathbb{H}$, one can efficiently test whether $(M, \tilde{N}) \in \widehat{\mathbb{GH}}$ by checking $\hat{e}(M, \tilde{H}) = \hat{e}(G, \tilde{N})$. [2]

2.2 Complexity Assumptions

Definition 1 (Decisional Diffie-Hellman (DDH) Assumption). *The DDH assumption holds relative to a group setup $\mathcal{G}$ if for all PPT adversaries $\mathcal{A}$*

$$\Pr\left[\begin{matrix}(\mathbb{G}, G, p) \leftarrow \mathcal{G}(1^\lambda);\; a, b, c \leftarrow \mathbb{Z}_p;\; t \leftarrow \{0, 1\}; \\ A := G^a;\; B := G^b;\; C := G^{tab+(1-t)c} \;: \mathcal{A}(G, A, B, C) = t\end{matrix}\right] \leq \frac{1}{2} + \nu(\lambda)\;\cdot$$

Definition 2 (Symmetric External Diffie-Hellman (SXDH) Assumption). *Given a bilinear group $\mathcal{P} := (\mathbb{G}, \mathbb{H}, \mathbb{T}, p, G, \tilde{H}, \hat{e})$, the SXDH assumption requires that the DDH assumption holds in both groups $\mathbb{G}$ and $\mathbb{H}$.*

2.3 Digital Signatures

A digital signature scheme over a bilinear group $\mathcal{P}$ generated by $\mathsf{BGSetup}$ for a message space $\mathcal{M}$ is a tuple $\mathcal{DS} := (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ whose definitions are:

- $\mathsf{KeyGen}(\mathcal{P})$ this randomized algorithm takes as input a bilinear group $\mathcal{P}$ and outputs a pair of secret/verification keys $(\mathsf{sk}, \mathsf{vk})$.
- $\mathsf{Sign}(\mathsf{sk}, m)$ takes as input a secret key sk and a message $m \in \mathcal{M}$, and outputs a signature σ.
- $\mathsf{Verify}(\mathsf{vk}, m, \sigma)$ this deterministic algorithm outputs 1 if σ is a vlaid signature on m w.r.t. the verification key vk.

Definition 3 (Correctness). *A signature scheme $\mathcal{DS}$ over a bilinear group generator* $\mathsf{BGSetup}$ *is* (perfectly) correct *if for all $\lambda \in \mathbb{N}$*

$$\Pr\left[\begin{matrix}\mathcal{P} \leftarrow \mathsf{BGSetup}(1^\lambda); (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(\mathcal{P}); \\ m \leftarrow \mathcal{M}; \sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m) : \mathsf{Verify}(\mathsf{vk}, m, \sigma) = 1\end{matrix}\right] = 1.$$

Definition 4 (Existential Unforgeability). *A signature scheme $\mathcal{DS}$ over a bilinear group generator* $\mathsf{BGSetup}$ *is* Existentially Unforgeable against adaptive Chosen Message Attack (EUF-CMA) *if for all $\lambda \in \mathbb{N}$ for all PPT adversaries $\mathcal{A}$*

$$\Pr\left[\begin{matrix}\mathcal{P} \leftarrow \mathsf{BGSetup}(1^\lambda); (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(\mathcal{P}); (\sigma^*, m^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathcal{P}, \mathsf{vk}) \\ : \mathsf{Verify}(\mathsf{vk}, m^*, \sigma^*) = 1 \quad \textit{and} \quad m^* \notin Q_{\mathsf{Sign}}\end{matrix}\right] \leq \nu(\lambda),$$

[2] The elements of this group are called Diffie-Hellman pairs in [25, 3].

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

We consider schemes which are re-randomizable (i.e. weakly unforgeable) in the sense that given a signature on a message m, anyone without knowledge of the signing key, can compute a new signature on the same message. A desirable property for such class of schemes is that randomized signatures are indistinguishable from fresh signatures on the same message. Thus, we define an algorithm $\mathsf{Randomize}$ which on input (vk, m, σ), with σ being a valid signature on m, outputs a new signature σ' on m.

Definition 5 (Randomizability). *A signature scheme $\mathcal{DS}$ over a bilinear group generator* $\mathsf{BGSetup}$ *is* randomizable *if for all $\lambda \in \mathbb{N}$ for all stateful adversaries $\mathcal{A}$*

$$\Pr\left[\begin{array}{l}\mathcal{P} \leftarrow \mathsf{BGSetup}(1^\lambda); (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(\mathcal{P});\\ (\sigma^*, m^*) \leftarrow \mathcal{A}(\mathcal{P}, \mathsf{sk}, \mathsf{vk}); b \leftarrow \{0,1\};\\ \sigma_0 \leftarrow \mathsf{Sign}(\mathsf{sk}, m^*); \sigma_1 \leftarrow \mathsf{Randomize}(\mathsf{vk}, m^*, \sigma^*);\\ \quad : \mathsf{Verify}(\mathsf{vk}, m^*, \sigma^*) = 1 \quad \textit{and} \quad \mathcal{A}(\sigma_b) = b\end{array}\right] = \frac{1}{2} + \nu(\lambda).$$

We say the scheme has *Perfect Randomizability* when $\nu(\lambda) = 0$. Note that the above definition of randomizability is stronger than the variant where the signature σ^* is generated by the challenger rather than the adversary herself.

When it is even infeasible for the adversary to output a new signature on a message that was queried to the sign oracle, we say the scheme is *Strongly Existentially Unforgeable against adaptive Chosen Message Attack (sEUF-CMA).*

A weaker variant of existential unforgeability, i.e. *Existential Unforgeability against a Random Message Attack (EUF-RMA)*, is similar to the above definition but on each call to the sign oracle, the oracle samples a message uniformly at random 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.

2.4 Structure-Preserving Signatures

Structure-preserving signatures [3] are signature schemes defined over bilinear groups where the messages, the verification key and signatures are all group elements and verifying signatures only involves deciding group membership of the signature components and evaluating pairing-product equations of the form of equation 1.

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

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. In a bilinear group based signature scheme, we refer to a signer that can only decide group membership, evaluate the bilinear map $\hat{e}$, compute the group operations in groups $\mathbb{G}, \mathbb{H}$ and $\mathbb{T}$, and compare group elements as a *generic signer.*

2.5 Randomizable Weakly Blind Signatures

A randomizable weakly blind signature scheme [12] is similar to a standard blind signature scheme [21] but unlike the latter, in the former the signer never gets to see the signed message. A randomizable blind signature scheme BS (with a two-move signature request phase) is a tuple of polynomial-time algorithms $\mathsf{BS} := (\mathsf{Setup_{BS}}, \mathsf{KeyGen_{BS}}, \mathsf{Request_{BS}}, \mathsf{Issue_{BS}}, \mathsf{Verify_{BS}}, \mathsf{Randomize_{BS}})$. All algorithms (bar $\mathsf{Setup_{BS}}$) are assumed to take as (implicit) input a parameter set $\mathsf{param_{BS}}$ output by $\mathsf{Setup_{BS}}$.

- $\mathsf{Setup_{BS}}(1^\lambda)$ outputs public parameters $\mathsf{param_{BS}}$.
- $\mathsf{KeyGen_{BS}}(\mathsf{param_{BS}})$ outputs a verification/secret key pair $(\mathsf{vk_{BS}}, \mathsf{sk_{BS}})$ for the signer.
- $(\mathsf{Request^0_{BS}}, \mathsf{Issue^1_{BS}}, \mathsf{Request^1_{BS}})$ is an interactive protocol between a user and a signer. The protocol is initiated by the user by calling $\mathsf{Request^0_{BS}}(\mathsf{vk_{BS}}, m)$ to obtain a value ρ_0 and some state information st^0_R (which is assumed to contain the message m). Then the signer and user execute, respectively,

$$(\beta_1, \mathsf{st}^1_I) \leftarrow \mathsf{Issue^1_{BS}}(\mathsf{sk_{BS}}, \rho_0) \quad \text{and} \quad \sigma \leftarrow \mathsf{Request^1_{BS}}(\beta_1, \mathsf{st}^0_R),$$

where σ is a signature on the message m (or the reject symbol $\perp$).

We write $\sigma \leftarrow \langle \mathsf{Request}_{\mathsf{BS}}(\mathsf{vk}_{\mathsf{BS}}, m), \mathsf{Issue}_{\mathsf{BS}}(\mathsf{sk}_{\mathsf{BS}})\rangle$ for the output of correct running of this protocol on the given inputs.

- $\mathsf{Verify}_{\mathsf{BS}}(\mathsf{vk}_{\mathsf{BS}}, m, \sigma)$ outputs 1 if σ is a valid signature on m and 0 otherwise.
- $\mathsf{Randomize}_{\mathsf{BS}}(\mathsf{vk}_{\mathsf{BS}}, \sigma)$ given a signature σ on an unknown message m, produces another valid signature σ' on the same message.

The security of randomizable weakly blind signatures [12] requires the following:

Definition 6 (Correctness). *A randomizable weakly blind signature scheme is* (perfectly) correct *if for all $\lambda \in \mathbb{N}$*

$$\Pr\left[\begin{array}{l} \mathsf{param}_{\mathsf{BS}} \leftarrow \mathsf{Setup}_{\mathsf{BS}}(1^{\lambda}); (\mathsf{vk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(\mathsf{param}_{\mathsf{BS}}); \\ m \leftarrow \mathcal{M}_{\mathsf{BS}}; \sigma \leftarrow \langle \mathsf{Request}_{\mathsf{BS}}(\mathsf{vk}_{\mathsf{BS}}, m), \mathsf{Issue}_{\mathsf{BS}}(\mathsf{sk}_{\mathsf{BS}})\rangle; \\ \sigma' \leftarrow \mathsf{Randomize}_{\mathsf{BS}}(\mathsf{vk}_{\mathsf{BS}}, \sigma) \\ \quad : \mathsf{Verify}_{\mathsf{BS}}(\mathsf{vk}_{\mathsf{BS}}, m, \sigma) = 1 \text{ and } \mathsf{Verify}_{\mathsf{BS}}(\mathsf{vk}_{\mathsf{BS}}, m, \sigma') = 1 \end{array}\right] = 1.$$

Definition 7 (Unforgeability). *A randomizable weakly blind signature scheme is* unforgeable *if for all $\lambda \in \mathbb{N}$, all PPT adversaries $\mathcal{A}$ have a negligible advantage in the game in Fig. 1.*

```
Experiment: $\mathsf{Exp}^{\mathrm{Unforge}}_{\mathsf{BS},\mathcal{A}}(\lambda)$:
– $\mathsf{param}_{\mathsf{BS}} \leftarrow \mathsf{Setup}_{\mathsf{BS}}(1^{\lambda})$.
– $(\mathsf{vk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(\mathsf{param}_{\mathsf{BS}})$.
– $\big((m_1, \sigma_1), \ldots, (m_{n+1}, \sigma_{n+1})\big) \leftarrow \mathcal{A}^{\mathsf{Issue}_{\mathsf{BS}}(\cdot,\cdot)}(\mathsf{vk}_{\mathsf{BS}}, \mathsf{param}_{\mathsf{BS}})$.
– Return 0 if any of the following holds. Otherwise, Return 1:
    ○ $\mathcal{A}$ called its oracle more than $n$ times.
    ○ $\exists i, j \in \{1, \ldots, n+1\}$ s.t. $i \neq j$, but $m_i = m_j$.
    ○ $\exists i \in \{1, \ldots, n+1\}$ s.t. $\mathsf{Verify}_{\mathsf{BS}}(\mathsf{vk}_{\mathsf{BS}}, m_i, \sigma_i) = 0$.
```

Fig. 1. The Unforgeability game for randomizable weakly blind signatures

Definition 8 (Weak Blindness). *A randomizable weakly blind signature scheme is* weakly blind *if for all $\lambda \in \mathbb{N}$, all PPT adversaries $\mathcal{A}$ have a negligible advantage in the game in Fig. 2.*

```
Experiment: $\mathsf{Exp}^{\mathrm{wBlind}}_{\mathsf{BS},\mathcal{A}}(\lambda)$:
– $\mathsf{param}_{\mathsf{BS}} \leftarrow \mathsf{Setup}_{\mathsf{BS}}(1^{\lambda})$.
– $(\mathsf{vk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}) \leftarrow \mathsf{KeyGen}_{\mathsf{BS}}(\mathsf{param}_{\mathsf{BS}})$.
– $m_0, m_1 \leftarrow \mathcal{M}_{\mathsf{BS}}$.
– $(\rho_0, \mathsf{st}^0_R) \leftarrow \mathsf{Request}^0_{\mathsf{BS}}(\mathsf{vk}_{\mathsf{BS}}, m_0)$.
– $(\beta_1, \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}(\mathsf{param}_{\mathsf{BS}}, \mathsf{vk}_{\mathsf{BS}}, \mathsf{sk}_{\mathsf{BS}}, \rho_0)$.
– $\sigma_0 \leftarrow \mathsf{Request}^1_{\mathsf{BS}}(\beta_1, \mathsf{st}^0_R)$.
– If $\sigma_0 = \perp$ or $\mathsf{Verify}_{\mathsf{BS}}(\mathsf{vk}_{\mathsf{BS}}, m_0, \sigma_0) = 0$ Then Return 0.
– $b \leftarrow \{0, 1\}$.
– If $b = 0$ Then $\sigma_1 \leftarrow \mathsf{Randomize}_{\mathsf{BS}}(\mathsf{vk}_{\mathsf{BS}}, \sigma_0)$.
– Else $\sigma_1 \leftarrow \langle \mathsf{Request}_{\mathsf{BS}}(\mathsf{vk}_{\mathsf{BS}}, m_1), \mathsf{Issue}_{\mathsf{BS}}(\mathsf{sk}_{\mathsf{BS}})\rangle$.
– $b^* \leftarrow \mathcal{A}(\mathsf{st}_{\mathcal{A}}, \sigma_0, \sigma_1)$.
– Return 1 If $b = b^*$ Else Return 0.
```

Fig. 2. The Weak Blindness game for randomizable weakly blind signatures

2.6 Groth-Sahai Proofs

Groth-Sahai (GS) proofs [34] are non-interactive proofs in the CRS model. We will use GS proofs that are secure under the SXDH assumption and that prove knowledge of witnesses to pairing-product equations of the form

$$\prod_{j=1}^{n} \hat{e}(A_j, \underline{\tilde{Y}_j}) \prod_{i=1}^{m} \hat{e}(\underline{X_i}, \tilde{B}_i) \prod_{i=1}^{m} \prod_{j=1}^{n} \hat{e}(\underline{X_i}, \underline{\tilde{Y}_j})^{\gamma_{i,j}} = \prod_{\ell=1}^{M} \hat{e}(G_\ell, \tilde{H}_\ell) \tag{2}$$

All underlined variables are part of the witness whereas the rest of the values are public constants. The language for these proofs is of the form $\mathcal{L} := \{\mathsf{statement} \mid \exists \, \mathsf{witness} : E(\mathsf{statement}, \mathsf{witness}) \text{ holds } \}$ where $E(\mathsf{statement}, \cdot)$ is a set of pairing-product equations. The system is defined by a tuple of algorithms $(\mathsf{GSSetup}, \mathsf{GSProve}, \mathsf{GSVerify}, \mathsf{GSExtract}, \mathsf{GSSimSetup}, \mathsf{GSSimProve})$. $\mathsf{GSSetup}$ takes as input the description of a bilinear group $\mathcal{P}$ and outputs a *binding* reference string crs and an extraction key xk. $\mathsf{GSProve}$ takes as input the string crs, a set of equations $\mathsf{statement}$ and a witness, and outputs a proof Ω for the satisfiability of the equations. $\mathsf{GSVerify}$ takes as input a set of equations, a string crs and a proof Ω and outputs 1 if the proof is valid, and 0 otherwise. $\mathsf{GSExtract}$ takes as input a binding crs, the extraction key xk and a valid proof Ω, and outputs the witness used for the proof. $\mathsf{GSSimSetup}$, on input a bilinear group $\mathcal{P}$, outputs a *hiding* string $\mathsf{crs}_{\mathrm{Sim}}$ and a trapdoor key tr that allows to simulate proofs. $\mathsf{GSSimProve}$ takes as input $\mathsf{crs}_{\mathrm{Sim}}$, a statement and the trapdoor tr and produces a simulated proof Ω_{Sim} without a witness. The distributions of strings crs and $\mathsf{crs}_{\mathrm{Sim}}$ are computationally indistinguishable and simulated proofs are indistinguishable from proofs generated by an honest prover. The proof system has perfect completeness, (perfect) soundness, composable witness-indistinguishability/composable zero-knowledge. We refer to [34] for the formal definitions and the details of the instantiations.

3 Unilateral Structure-Preserving Signatures on Diffie-Hellman Pairs

We define Unilateral Structure-Preserving Signatures on Diffie-Hellman Pairs (USPSDH) as structure-preserving signatures with the following extra conditions on top of those required by traditional structure-preserving signatures (cf. Section 2.4):

i) Messages are of the form $(M, \tilde{N}) \in \widehat{\mathbb{GH}} \subset \mathbb{G} \times \mathbb{H}$.
ii) Signatures are either of the form $\sigma = (S_1, \ldots, S_k) \in \mathbb{G}^k$, whereas the verification key is of the form $\mathsf{vk} = (\tilde{Y}_1, \ldots, \tilde{Y}_n) \in \mathbb{H}^n$ or signatures are of the form $\sigma = (\tilde{S}_1, \ldots, \tilde{S}_k) \in \mathbb{H}^k$, whereas the verification key is of the form $\mathsf{vk} = (Y_1, \ldots, Y_n) \in \mathbb{G}^n$.

We remark that there exist schemes, e.g. [28, 29] which conform to the above requirements. Also, there are schemes, e.g. [25, 3], which satisfy the first requirement but not the second.

The following lemma proves that our impossibility results and lower bound proofs in the next section hold even if one allows the verification key and public parameters (other than the group generator) to be from the same source group as the signature components.

Lemma 1. *Having a verification key component or a public parameter (other than the group generator) in the same group as the signature is redundant.*

Proof. Let us consider the case where the signature is of the form $\sigma = (S_1, \ldots, S_k) \in \mathbb{G}^k$ whereas the verification key $\mathsf{vk} = (X_1, \ldots, X_n, \tilde{Y}_1, \ldots, \tilde{Y}_{n'}) \in \mathbb{G}^n \times \mathbb{H}^{n'}$. The proof for the opposite case where the groups are transposed is similar.

Since X_i's are in same group as S_j's (for all possible choices of i and j), the verification equations cannot have any pairing of the form $\hat{e}(S_i, X_j)$. Thus, the only pairings that X_i can feature in in the verification equations are: $\hat{e}(X_i, \tilde{N})$, $\hat{e}(X_i, \tilde{H})$ or $\hat{e}(X_i, \tilde{Y}_j)$. In the first case, the pairing is equivalent to $\hat{e}(M, \tilde{H}^{x_i})$ where x_i is the discrete logarithm of X_i to the base G. Thus, we can replace X_i by $\tilde{X}_i := \tilde{H}^{x_i}$. In the latter two cases, we can WLOG move the result of the pairing to the right-hand side of the verification equation and relax Equation (1) to allow the right-hand side to be $Z_{\mathbb{T}}$ instead of $1_{\mathbb{T}}$.

□

4 Optimal CMA-Secure Scheme I

We give here a (weakly) existentially unforgeable against adaptive chosen-message attack signature scheme with signatures consisting of two elements from group $\mathbb{G}$. Besides checking membership of the message in $\widehat{\mathbb{G}\mathbb{H}}$, verifying a signature only requires the evaluation of 1 pairing-product equation with 4 pairings in total 1 of which can be precomputed. Depending on the application, the number of pairings can be further reduced to 3 pairings one of which can be precomputed since two of the pairings share the same left-hand side argument.

Given the description of Type-III bilinear groups $\mathcal{P}$ output by $\mathsf{BGSetup}(1^\lambda)$, the scheme is as follows:

- $\mathsf{KeyGen}(\mathcal{P})$: Select $x, y \leftarrow \mathbb{Z}_p^\times$. Set $\mathsf{sk} := (x, y)$ and $\mathsf{vk} := (\tilde{X}, \tilde{Y}) := (\tilde{H}^x, \tilde{H}^y) \in \mathbb{H}^2$.
- $\mathsf{Sign}(\mathsf{sk}, (M, \tilde{N}))$: To sign a message $(M, \tilde{N}) \in \widehat{\mathbb{G}\mathbb{H}}$, select $r \leftarrow \mathbb{Z}_p$, and set $R := G^r$, $S := \big((G^x \cdot M)^r \cdot G\big)^{\frac{1}{y}}$. Return $\sigma := (R, S) \in \mathbb{G}^2$.
- $\mathsf{Verify}(\mathsf{vk}, (M, \tilde{N}), \sigma = (R, S))$: Return 1 iff $R, S \in \mathbb{G}$, $(M, \tilde{N}) \in \widehat{\mathbb{G}\mathbb{H}}$, and the following holds:

$$\hat{e}(S, \tilde{Y}) = \hat{e}(R, \tilde{X})\hat{e}(R, \tilde{N})\hat{e}(G, \tilde{H})$$

Remark 1. Note that the signing algorithm can be performed even without knowledge of the exponent x if one has the element $X := G^x \in \mathbb{G}$ (instead of $x \in \mathbb{Z}_p$) as part of the secret key sk.

Correctness of the scheme follows by inspection and is straightforward to verify.

The signature is weakly unforgeable. For instance, given two distinct signatures $\sigma_1 = (R_1, S_1)$ and $\sigma_2 = (R_2, S_2)$ on a message $(M, \tilde{N})$, one can without knowledge of the signing key compute a new signature $\sigma' = (R', S')$ on the same message by computing e.g. $(R' := R_1^2 \cdot R_2^{-1}, S' := S_1^2 \cdot S_2^{-1})$.

Theorem 1. *The structure-preserving signature scheme is existentially weakly unforgeable against a chosen-message attack in the generic group model.*

Proof. Since the adversary is generic, it can only produce linear combinations of the signatures' elements, verification key elements and public parameters in each of the source groups. The linear combinations represent Laurent polynomials in the discrete logarithm of those elements. We will prove that no linear combinations produce Laurent polynomials corresponding to a forgery on a message that was not queried to the sign oracle.

Public elements in $\mathbb{H}$ are $\tilde{H}$, $\tilde{X}$, $\tilde{Y}$ which correspond to the discrete logarithms 1, x and y, respectively. Thus, this means that at the it-h sign query on $(M_i, \tilde{N}_i)$, $\tilde{N}_i$ can only be a linear combination of $\tilde{H}$ are $\tilde{X}$, $\tilde{Y}$, thus, we have

$$n_i = a_{n_i} + b_{n_i} x + c_{n_i} y$$

Similarly, M_i can only be a linear combination of $G, \{R_j\}_{j=1}^{i-1}, \{S_j\}_{j=1}^{i-1}$. Thus, we have

$$m_i = a_{m_i} + \sum_{j=1}^{i-1} b_{m_{i,j}} r_j + \sum_{j=1}^{i-1} c_{m_{i,j}} (\frac{r_j x}{y} + \frac{r_j m_j}{y} + \frac{1}{y})$$

After q signing queries, (m^*, n^*), which is the discrete logarithm of the forged message $(M^*, \tilde{N}^*)$ must be of the form

$$n^* = a_n + b_n x + c_n y$$
$$m^* = a_m + \sum_{i=1}^{q} b_{m_i} r_i + \sum_{i=1}^{q} c_{m_i} (\frac{r_i x}{y} + \frac{r_i m_i}{y} + \frac{1}{y})$$

Since we must have $(M^*, \tilde{N}^*) \in \widehat{\mathbb{G}\mathbb{H}}$, i.e. $m^* = n^*$, We must have $b_n = c_n = 0$ and $b_{m_i} = c_{m_i} = 0$ for all $i \in [q]$ and $a_m = a_n$. Thus, we have

$$m^* = n^* = a_m$$

Similarly, the forgery (R^*, S^*) can only be a linear combination of the group elements from $\mathbb{G}$, i.e. a linear combination of G, $\{R_i\}_{i=1}^q$ and $\{S_i\}_{i=1}^q$ and therefore we have

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

For the forgery to be a valid signature, r^* and s^* must satisfy $s^* y = r^* x + r^* m^* + 1$. Therefore, we must have

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

Thus, we must have

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

Note that there is no term in y, $r_i y$ on the right-hand side so we must have $a_s = 0$, and $b_{s,i} = 0$ for all i, so

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

There is no term in $\frac{x}{y}$ on the left-hand side so $c_{r,i} = 0$ for all i. Also, since there is no term in x on the left-hand side, we also have $a_r = 0$. Thus, we have

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

The monomial $r_i x$ implies $c_{s,i} = b_{r,i}$ for all i, whereas the monomial r_i implies $c_{s,i} m_i = b_{r,i} m^*$. Since we have $c_{s,i} = b_{r,i}$, this means we have $m^* = m_i$ for some i. Hence, the signature (R^*, S^*) is on a message pair $(M_i, \tilde{N}_i)$ that was queried to the sign oracle and thus is not a forgery on a new message. □

4.1 Randomizability/Strong Unforgeability

We prove the following theorem regarding the randomizability/strong unforgeability of the above signature scheme.

Theorem 2. *The scheme is strongly existentially unforgeable against an adversary that queries the signing oracle on each message once at most.*

Proof. Following from the proof of Theorem 1, we have for the adversary forgery to be valid, we must have:

$$\sum_{i=1}^{q} c_{s,i}(r_i x + r_i m_i + 1) = \sum_{i=1}^{q} b_{r,i} r_i x + \sum_{i=1}^{q} b_{r,i} r_i m^* + 1 \tag{3}$$

Let J be the subset of $\{1, \ldots, q\}$ containing indices of the signatures on the message m^* that was obtained from the signing oracle, i.e. J is the set of indices of the queries on message m^*. Let $R_{m^*} = \{R_i\}_{i \in J}$ and $S_{m^*} = \{S_i\}_{i \in J}$. From the left-hand side of (3), it is clear that S^* can only be a linear combination of elements of the set S_{m^*}. Similarly, R^* can only be a linear combination of elements of the set R_{m^*}. Since the adversary is restricted to at most a single signing query on each message, we have $0 \leq |S_{m^*}| = |R_{m^*}| \leq 1$. If $S_{m^*} = R_{m^*} = \emptyset$, a forgery on the message m^*, which was not queried to the signing oracle, would contradict Theorem 1.

Now, for (3) to hold, we must have $\sum c_{s,i} = 1$ which implies $b_{r,i} = 1$ and thus $r^* = r_i$ and the signature is that that was obtained from the sign oracle.

□

Let us now define the randomization algorithm Randomize for the above scheme as follows:

- $\mathsf{Randomize}\big(\mathsf{vk}, (M, \tilde{N}), \{\sigma_i = (R_i, S_i)\}_{i=1}^{2}\big)$: For any two distinct signatures σ_1 and σ_2 on the message $(M, \tilde{N})$, i.e. $R_1 \neq R_2$, satisfying $\mathsf{Verify}(\mathsf{vk}, (M, \tilde{N}), \sigma_i) = 1$ for all $i \in [2]$.
  To obtain a new signature σ' on $(M, \tilde{N})$, choose $a \leftarrow \mathbb{Z}_p$ and compute $b = 1 - a$ (which satisfies $a + b = 1$). Now compute $R' := R_1^a \cdot R_2^b$, $S' := S_1^a \cdot S_2^b$.
  Return $\sigma' := (R', S')$.

Theorem 3. *Randomized signatures are perfectly indistinguishable from fresh signatures on the same message.*

Proof. In the Sign algorithm, r is chosen uniformly at random from $\mathbb{Z}_p$, whereas in the Randomize algorithm, a (resp. b) is also chosen uniformly at random from $\mathbb{Z}_p$. Moreover, for any possible $r \in \mathbb{Z}_p$ such that $R = G^r$, there is $a \in \mathbb{Z}_p$ such that $r = ar_1 + (1 - a)r_2$ for any $r_1, r_2 \in \mathbb{Z}_p$ satisfying $r_1 \neq r_2$. Therefore, the distribution of signatures output by the Randomize algorithm is identical to that of signatures output the Sign algorithm. □

4.2 Combined Unforgeability for Messages

The notion of structure-preserving signature schemes with combined unforgeability [33] (similarly to selectively randomizable schemes [6]), are signature schemes where the same scheme can allow (at the discretion of the signer) either strongly unforgeable signature or ones that can be re-randomized.

We proved that in our scheme the only way to obtain a new signature on the same message is by linear combination of distinct signatures on the same message. One can exploit this feature so that the signer can decide which messages signatures upon which can be re-randomized and which cannot which might be useful for some applications. For those messages to be restricted, the signer only allows a single signing query on them, whereas for those signatures upon which can be re-randomized, the signing oracle returns at least two distinct signatures $\sigma = (R, S)$ and $\sigma' = (R', S')$ satisfying $R \neq R'$.

5 Optimal CMA-Secure Scheme II

We give here an efficient publicly re-randomizable structure-preserving scheme that is existentially unforgeable against adaptive chosen-message attack. The scheme yields signatures with two group elements from group $\mathbb{G}$.

Besides checking membership of the message in $\widehat{\mathbb{GH}}$, verifying a signature, requires 1 PPE equation with 3 pairings in total or 2 pairings and 1 point addition since 2 of the 3 pairings required share the same left argument. When verifying a signature, we additionally need to check that $R \in \mathbb{G}^\times$ (i.e. $R \in \mathbb{G} \backslash \{1_\mathbb{G}\}$).

Given the description of Type-III bilinear groups $\mathcal{P}$ output by $\mathsf{BGSetup}(1^\lambda)$, the scheme is as follows:

- $\mathsf{KeyGen}(\mathcal{P})$: Select $x, y \leftarrow \mathbb{Z}_p^\times$. Set $\mathsf{sk} := (x, y)$ and $\mathsf{vk} := (\tilde{X}, \tilde{Y}) := (\tilde{H}^x, \tilde{H}^y) \in \mathbb{H}^2$.
- $\mathsf{Sign}(\mathsf{sk}, (M, \tilde{N}))$: To sign a message $(M, \tilde{N}) \in \widehat{\mathbb{GH}}$, select $r \leftarrow \mathbb{Z}_p^\times$, and set $R := G^r$, $S := \big(G^x \cdot M\big)^{\frac{r}{y}}$. Return $\sigma := (R, S) \in \mathbb{G}^2$.
- $\mathsf{Verify}(\mathsf{vk}, (M, \tilde{N}), \sigma = (R, S))$: Return 1 iff $R \in \mathbb{G}^\times, S \in \mathbb{G}$, $(M, \tilde{N}) \in \widehat{\mathbb{GH}}$, and the following holds:

$$\hat{e}(S, \tilde{Y}) = \hat{e}(R, \tilde{X})\hat{e}(R, \tilde{N})$$

- $\mathsf{Randomize}(\mathsf{vk}, (M, \tilde{N}), \sigma = (R, S))$: Select $r' \leftarrow \mathbb{Z}_p^\times$, and set $R' := R^{r'}$, $S' := S^{r'}$. Return $\sigma' := (R', S')$.

Remark 2. Again, the signing algorithm can be performed even without knowledge of the exponent x if one has the element $X := G^x \in \mathbb{G}$ (instead of $x \in \mathbb{Z}_p$) as part of the secret key sk. Also, note that the component R of the signature is information-theoretically independent of the message and hence even when proving knowledge of a signature on the message, one can reveal this component of the signature after re-randomizing it.

Correctness of the scheme follows by inspection and is straightforward to verify. The scheme is perfectly randomizable as the distribution of re-randomized signatures is identical to that of fresh signatures on the same message.

Theorem 4. *The structure-preserving signature scheme is existentially weakly unforgeable against a chosen-message attack in the generic group model.*

Proof. Public elements in $\mathbb{H}$ are $\tilde{H}$, $\tilde{X}$, $\tilde{Y}$ which correspond to the discrete logarithms 1, x and y, respectively. We note that our proof of security only relies on the forgery being a valid element of $\widehat{\mathbb{GH}}$. In other words, the scheme is still secure even if the adversary queries the scheme on arbitrary messages from $\mathbb{G}$ for which it does not know the corresponding message component in $\mathbb{H}$.

During the it-h signing query on $(M_i, \tilde{N}_i)$, $\tilde{N}_i$ can only be a linear combination of $\tilde{H}$ are $\tilde{X}$, $\tilde{Y}$, thus, we have

$$n_i = a_{n_i} + b_{n_i}x + c_{n_i}y$$

Similarly, M_i can only be a linear combination of $G, \{R_j\}_{j=1}^{i-1}, \{S_j\}_{j=1}^{i-1}$. Thus, we have

$$m_i = a_{m_i} + \sum_{j=1}^{i-1} b_{m_{i,j}} r_j + \sum_{j=1}^{i-1} c_{m_{i,j}} (\frac{r_j x}{y} + \frac{r_j m_j}{y})$$

After q signing queries, (m^*, n^*), which is the discrete logarithm of the forged message $(M^*, \tilde{N}^*)$, must be of the form

$$\begin{aligned} n^* &= a_n + b_n x + c_n y \\ m^* &= a_m + \sum_{i=1}^{q} b_{m_i} r_i + \sum_{i=1}^{q} c_{m_i} (\frac{r_i x}{y} + \frac{r_i m_i}{y}) \end{aligned}$$

Since we must have $n^* = m^*$ for the forgery to be a valid element of $\widehat{\mathbb{GH}}$, we have

$$m^* = n^* = a_m$$

Similarly, the signature (R^*, S^*) have the form

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

For the forgery to be a valid signature, s^* and r^* must satisfy $s^*y = r^*x + r^*m^*$. So we must have

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

Thus, we must have

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

Note that there is no term in y or $r_i y$ on the right-hand side, so we must have $a_s = 0$, $b_{s,i} = 0$ for all i, Thus, we have

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

There is no term $\frac{r_i x^2}{y}$ on the left-hand side so $c_{r,i} = 0$ for all i. Also, since no term in x on the left-hand side, we also have $a_r = 0$. Thus, we have

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

The monomial $r_i x$ implies $c_{s,i} = b_{r,i}$ for all i. Since we require that $R^* \in \mathbb{G}^\times$, we must have $r^* \neq 0$ and therefore we must have at least a single value of $c_{s,i} = b_{r,i} \neq 0$. Now the monomial r_i implies $c_{s,i} m_i = b_{r,i} m^*$ which means $m^* = m_i$ for some i. Thus, the signature (R^*, S^*) is on a message pair $(M_i, \tilde{N}_i)$ that was queried to the sign oracle and thus is not a forgery.

6 Optimal CMA-Secure One-Time Signature Schemes

We give here a (strongly) existentially unforgeable one-time signature scheme that is secure against a chosen-message attack with one-element signatures. Besides checking membership of the message in $\widehat{\mathbb{G}\mathbb{H}}$, verification requires the evaluation of a single PPE equation with 3 pairings in total one of which can be pre-computed when verifying multiple signatures (under different keys) on the same message. Alternatively, verification can be performed by evaluating only 2 pairings and one point addition since two pairings share the same left-hand side argument.

We will show in Section 7 that the same scheme can also be used as a one-time structure-preserving signature scheme for messages in $\mathbb{G}$ (resp. $\mathbb{H}$) by replacing the pairing $\hat{e}(G, \tilde{N})$ in the PPE verification equation by $\hat{e}(M, \tilde{H})$. This essentially yields a new one-time signature scheme for unilateral messages in the Type-III setting matching the optimal one-time scheme in [6] in every respect.

Given the description of Type-III bilinear groups $\mathcal{P}$ output by $\mathsf{BGSetup}(1^\lambda)$, the scheme is as follows:

- $\mathsf{KeyGen}(\mathcal{P})$: Select $x, y \leftarrow \mathbb{Z}_p^\times$. Set $\mathsf{sk} := (x, y)$ and $\mathsf{vk} := (\tilde{X}, \tilde{Y}) := (\tilde{H}^x, \tilde{H}^y) \in \mathbb{H}^2$.
- $\mathsf{Sign}(\mathsf{sk}, (M, \tilde{N}))$: To sign a message $(M, \tilde{N}) \in \widehat{\mathbb{GH}}$, compute $\sigma := \big(G^x \cdot M\big)^{\frac{1}{y}}$.
- $\mathsf{Verify}(\mathsf{vk}, (M, \tilde{N}), \sigma)$: Return 1 iff $\sigma \in \mathbb{G}$, $(M, \tilde{N}) \in \widehat{\mathbb{GH}}$, and the following holds:

$$\hat{e}(\sigma, \tilde{Y}) = \hat{e}(G, \tilde{X})\hat{e}(G, \tilde{N})$$

Remark 3. Again, note that the signing algorithm can be performed even without knowledge of the exponent x if one has the element $X := G^x \in \mathbb{G}$ (instead of $x \in \mathbb{Z}_p$) as part of the secret key sk.

Correctness of the scheme follows by inspection and is straightforward to verify. The signning algorithm is deterministic and therefore for any message there is only 1 potential signature. We prove the following theorem.

Theorem 5. *The one-time structure-preserving signature scheme is strongly existentially unforgeable against a one-time chosen-message attack in the generic group model.*

Proof. We show that the linear combinations the generic adversary can produce out of the combinations of the elements of the signatures, verification key and public parameters in each of the source groups, cannot correspond to Laurent polynomials representing a valid forgery. Public elements in $\mathbb{H}$ are $\tilde{H}$, $\tilde{X}$, $\tilde{Y}$ which correspond to the discrete logairthms 1, x and y, respectively. Thus, this means that the message $(M, \tilde{N})$ queried to the sign oracle $\tilde{N}$ can only be a linear combination of $\tilde{H}$, $\tilde{X}$ and $\tilde{Y}$. After 1 signing queries, the message the adversary forges a signature on must be in the form

$$m^* = n^* = a_m$$

Similarly, the signature $\sigma^* = S^*$ must have the form

$$s^* = a_s + b_s\frac{x}{y} + b_s\frac{m}{y}$$

For the forgery to be a valid signature, s^* must satisfy $s^*y = x + m^*$. Therefore, we must have

$$\Big(a_s + b_s\frac{x}{y} + b_s\frac{m}{y}\Big)y = x + m^*$$

Thus, we must have

$$a_s y + b_s x + b_s m = x + m^*$$

There is no term in y on the right-hand side so we must have $a_s = 0$. Thus, we have

$$b_s x + b_s m = x + m^*$$

By the monomial x, we have $b_s = 1$. For the two sides to be equal, we must have $b_s m = m^*$. Since we have $b_s = 1$, it means we must have $m^* = m$. This means the forgery is on the same message queried to the sign oracle.

□

6.1 Signing a Vector of Diffie-Hellman Pairs

The above scheme can be extended to sign k Diffie-Hellman pairs as follows:

- $\mathsf{KeyGen}(\mathcal{P})$: Select $x_1, \ldots, x_k, y \leftarrow \mathbb{Z}_p^\times$. Set $\mathsf{sk} := (x_1, \ldots, x_k, y)$ and $\mathsf{vk} := (\tilde{X}_1, \ldots, \tilde{X}_k, \tilde{Y}) := (\tilde{H}^{x_1}, \ldots \tilde{H}^{x_k}, \tilde{H}^y) \in \mathbb{H}^{k+1}$.
- $\mathsf{Sign}(\mathsf{sk}, \big((M, \tilde{N})_1, \ldots, (M, \tilde{N})_k\big))$: To sign a vector of messages $\big((M, \tilde{N})_1, \ldots, (M, \tilde{N})_k\big) \in \widehat{\mathbb{GH}}^k$, return $\sigma := \big(G^{x_1} \cdot M_1 \cdot \prod_{i=2}^k M_i^{x_i}\big)^{\frac{1}{y}}$.
- $\mathsf{Verify}(\mathsf{vk}, \big((M, \tilde{N})_1, \ldots, (M, \tilde{N})_k\big), \sigma)$: Return 1 iff $\sigma \in \mathbb{G}$, $(M, \tilde{N})_i \in \widehat{\mathbb{GH}}$, and the following holds:

$$\hat{e}(\sigma, \tilde{Y}) = \hat{e}(G, \tilde{X}_1)\hat{e}(G, \tilde{N}_1)\prod_{i=2}^k \hat{e}(M_i, \tilde{X}_i)$$

Correctness of the scheme follows by inspection and is straightforward to verify. The scheme being deterministic ensures that for any vector of messages there is only 1 potential signature. The following theorem proves that a one-time chosen-message adversary has a negligible probability in producing a signature on a vector of messages different from the one it queried its sign oracle on.

Theorem 6. *The scheme is strongly existentially unforgeable against a one-time chosen-message attack.*

Proof. Let $\mathcal{A}$ be an adversary that breaks the unforgeability of the scheme. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that breaks the strong existential unforgeability of the single-message one-time scheme in Section 6.

Adversary $\mathcal{B}$ gets $\mathsf{vk} = (\tilde{X}, \tilde{Y})$ from its game and has a single-message one-time signing oracle. $\mathcal{B}$ constructs its verification key by choosing $x_2, \ldots, x_k \leftarrow \mathbb{Z}_p^\times$ and computing $\tilde{X}_i := \tilde{H}^{x_i}$, for $i = 2, \ldots, k$. It then forwards its verification key $\mathsf{vk}^* := (\tilde{X}, \tilde{X}_2, \ldots, \tilde{X}_k, \tilde{Y})$ to $\mathcal{A}$. When queried on the message $\big((M, \tilde{N})_1, \ldots, (M, \tilde{N})_k\big) \in \widehat{\mathbb{GH}}^k$, $\mathcal{B}$ computes $M := M_1 \cdot \prod_{i=2}^k M_i^{x_i}$ and $\tilde{N} := \tilde{N}_1 \cdot \prod_{i=2}^k \tilde{N}_i^{x_i}$ and forwards $(M, \tilde{N})$ to its own sign oracle and returns the resultant signature σ to $\mathcal{A}$. Eventually, when $\mathcal{A}$ returns its forgery σ^* on a message vector $\big((M^*, \tilde{N}^*)_1, \ldots, (M^*, \tilde{N}^*)_k\big) \in \widehat{\mathbb{GH}}^k$, where $\big((M^*, \tilde{N}^*)_1, \ldots, (M^*, \tilde{N}^*)_k\big) \neq \big((M, \tilde{N})_1, \ldots, (M, \tilde{N})_k\big)$, $\mathcal{B}$ computes $M^* := M_1^* \cdot \prod_{i=2}^k M_i^{*x_i}$ and $\tilde{N}^* := \tilde{N}_1^* \cdot \prod_{i=2}^k \tilde{N}_i^{*x_i}$ and returns σ^* and the message $(M^*, \tilde{N}^*)$ as its forgery in its game. Clearly, if $\mathcal{A}$ wins its game, $\mathcal{B}$ wins its game with the same probability.

□

6.2 Signing Messages in $\widehat{\mathbb{GH}} \times \mathbb{G}^{k-1}$

The scheme in Section 6.1 can also be used to sign messages in $\widehat{\mathbb{GH}} \times \mathbb{G}^{k-1}$. The scheme is as follows, where the KeyGen algorithm is the same as that in Section 6.1:

- $\mathsf{Sign}\Big(\mathsf{sk}, \big((M, \tilde{N}), (M_1, \ldots, M_{k-1})\big)\Big)$: To sign a vector of messages $\big((M, \tilde{N}), , (M_1, \ldots, M_{k-1})\big) \in \widehat{\mathbb{GH}} \times \mathbb{G}^{k-1}$, return $\sigma := \big(G^{x_1} \cdot M \cdot \prod_{i=2}^k M_{i-1}^{x_i}\big)^{\frac{1}{y}}$.
- $\mathsf{Verify}(\mathsf{vk}, \big((M, \tilde{N}), (M_1, , \ldots M_{k-1})\big), \sigma)$: Return 1 iff $\sigma \in \mathbb{G}$, $(M, \tilde{N}) \in \widehat{\mathbb{GH}}$, $M_i \in \mathbb{G}$ for $i = 1, \ldots, k-1$, and the following holds:

$$\hat{e}(\sigma, \tilde{Y}) = \hat{e}(G, \tilde{X}_1)\hat{e}(G, \tilde{N}) \prod_{i=2}^{k} \hat{e}(M_{i-1}, \tilde{X}_i)$$

Correctness of the scheme follows by inspection and is straightforward to verify. The scheme being deterministic ensures that for any vector of messages there is only 1 potential signature.

Since the messages $(M_1, \ldots, M_{k-1})$ do not have corresponding components in $\mathbb{H}$, we cannot use a reduction to the scheme in Section 6 as we did in Theorem 6. Alternatively, the following theorem proves that the scheme is secure in the generic group model.

Theorem 7. *The scheme is strongly existentially unforgeable against a one-time chosen-message attack.*

Proof. We show that the linear combinations the generic adversary can produce out of the combinations of the signatures' elements, verification key elements and public parameters in each of the source groups, cannot correspond to Laurent polynomials representing a valid forgery. Public elements in $\mathbb{H}$ are $\tilde{H}, \tilde{X}_1, \ldots, \tilde{X}_k, \tilde{Y}$ which correspond to the discrete logairthms 1, x_1, …, x_k and y, respectively. The message $\big((M^*, \tilde{N}^*), \{M_i^*\}_{i=1}^{k-1}\big)$ on which the adversary forges a signature $\sigma^* = S^*$ can only be of the form

$$n^* = a_n + \sum_{i=1}^{k} b_{n,i} x_i + c_n y$$

$$m^* = a_m + b_m \big(\frac{x_1}{y} + \frac{m}{y} + \sum_{i=2}^{k} \frac{m_{i-1} x_i}{y}\big)$$

$$m_i^* = a_{m_i} + b_{m_i} \big(\frac{x_1}{y} + \frac{m}{y} + \sum_{j=2}^{k} \frac{m_{j-1} x_j}{y}\big), \text{ for } i = 1, \ldots, k-1.$$

Since we require that $(M^*, \tilde{N^*}) \in \widehat{\mathbb{G}\mathbb{H}}$, we must have $m^* = n^* = a_m$. Similarly, the signature $\sigma^* = S^*$ must have the form

$$s^* = a_s + b_s\big(\frac{x_1}{y} + \frac{m}{y} + \sum_{i=2}^{k} \frac{m_{i-1}x_i}{y}\big)$$

For the forgery to be a valid signature, s^* must satisfy $s^* y = x_1 + m^* + \sum_{i=2}^{k} m_{i-1}^* x_i$. Therefore, we must have

$$\Big(a_s + b_s\big(\frac{x_1}{y} + \frac{m}{y} + \sum_{i=2}^{k} \frac{m_{i-1}x_i}{y}\big)\Big)y = x_1 + m^* + \sum_{i=2}^{k} m_{i-1}^* x_i$$

So we must have

$$a_s y + b_s\big(x_1 + m + \sum_{i=2}^{k} m_{i-1}x_i\big) = x_1 + m^* + \sum_{i=2}^{k} m_{i-1}^* x_i.$$

There is no term in y on the right-hand side so we must have $a_s = 0$, Thus, we have

$$b_s\big(x_1 + m + \sum_{i=2}^{k} m_{i-1}x_i\big) = x_1 + m^* + \sum_{i=2}^{k} m_{i-1}^* x_i.$$

By the monomial x_1, we have $b_s = 1$. For the two sides to be equal, we must have $b_s m = m^*$ and $b_s m_{i-1} x_i = m_{i-1}^* x_i$ for all $i = 2, \ldots, k$. Since we have $b_s = 1$, it means we must have $m^* = m$ and $m_{i-1} = m_{i-1}^*$ for all $i = 2, \ldots, k$. This means the forgery is on the same vector queried to the sign oracle. □

7 Optimal One-Time Scheme for Unilateral Messages

As mentioned earlier, the previous one-time signature scheme can be used to sign unilateral messages, i.e. messages in $\mathbb{G}^k$. Thus, we obtain a one-time structure-preserving scheme for a vector of unilateral messages matching the optimal scheme in the Type-III setting [6] in every respect. By transposing the groups, one can similarly sign messages in $\mathbb{H}^k$. The scheme is as follows:

- $\mathsf{KeyGen}(\mathcal{P})$: Select $x_1, \ldots, x_k, y \leftarrow \mathbb{Z}_p^\times$. Set $\mathsf{sk} := (x_1, \ldots, x_k, y)$ and $\mathsf{vk} := (\tilde{X}_1, \ldots, \tilde{X}_k, \tilde{Y}) := (\tilde{H}^{x_1}, \ldots \tilde{H}^{x_k}, \tilde{H}^{y}) \in \mathbb{H}^{k+1}$.
- $\mathsf{Sign}\big(\mathsf{sk}, (M_1, \ldots, M_k)\big)$: To sign a vector of messages $\big(M_1, \ldots, M_k\big) \in \mathbb{G}^k$, return $\sigma := \big(G^{x_1} \cdot M_1 \cdot \prod_{i=2}^{k} M_i^{x_i}\big)^{\frac{1}{y}}$.
- $\mathsf{Verify}(\mathsf{vk}, (M_1, \ldots, M_k), \sigma)$: Return 1 iff $\sigma \in \mathbb{G}$, $M_i \in \mathbb{G}$ for $i = 1, \ldots, k$, and the following holds:

$$\hat{e}(\sigma, \tilde{Y}) = \hat{e}(G, \tilde{X}_1)\hat{e}(M_1, \tilde{H}) \prod_{i=2}^{k} \hat{e}(M_i, \tilde{X}_i)$$

Efficiency. To sign a vector $\mathbb{G}^k$, the verification key consists of $k+1$ group elements from group $\mathbb{H}$. Signing requires $k+1$ exponentiations in $\mathbb{G}$, whereas verification requires 1 pairing-product equation involving $k+2$ pairings. The signature consists of a single group element from $\mathbb{G}$ (regardless of the length of the vector to be signed). Those costs are identical to those in the optimal one-time scheme in the Type-III setting in [6].

Correctness of the scheme follows by inspection and is straightforward to verify. The scheme being deterministic ensures that for any vector of messages there is only 1 potential signature. The proof for the following theorem, which proves the existential unforgeability of the scheme against a chosen-message attack in the generic group model, is very similar to the proof of Theorem 7. For completeness, we give the proof in Appendix A.

Theorem 8. *The scheme is strongly existentially unforgeable against a one-time chosen-message attack.*

8 Lower Bounds & Impossibility Results for USPSDH Schemes

In this section we investigate some lower bounds and prove some impossibility results for USPSDH Schemes. Our proofs are general and do not require the right-hand side of the verification equations to be $Z_{\mathbb{T}} = 1_{\mathbb{T}}$.

8.1 Impossibility of Strongly Unforgeable CMA Secure Schemes

We prove here that there exists no generic-signer USPSDH scheme that is strongly existentially unforgeable against an adversary that makes $q > 1$ chosen message signing queries. We note, however, that there exist such schemes that are RMA secure or where, for instance, we do not allow the adversary to query the sign oracle on the same message more than once.

Theorem 9. *There is no generic-signer USPSDH scheme that is strongly unforgeable against a chosen message attack for $q > 1$ queries.*

Proof. Let us consider the case where the signature $\sigma = (S_1, \ldots, S_k) \in \mathbb{G}^k$ whereas the verification key $\mathsf{vk} = (\tilde{X}_1, \ldots, \tilde{X}_n) \in \mathbb{H}^n$. The proof for the opposite case where the groups are transposed is similar. Such a scheme would have a number of verification equations of the form of Equation (4).

$$\prod_{i=1}^{k}\prod_{j=1}^{n} \hat{e}(S_i, \tilde{X}_j)^{a_{i,j,e}} \prod_{i=1}^{k} \hat{e}(S_i, \tilde{N})^{b_{i,e}} \prod_{i=1}^{n} \hat{e}(M, \tilde{X}_i)^{c_{i,e}} \hat{e}(M, N)^{d_e} = Z_{e\mathbb{T}} \tag{4}$$

In [6] (Lemma 1), Abe et al. proved that for a structure-preserving signature scheme to be secure against a random message attack for $q > 1$ signing queries, there must be, for each message, superpolynomially many potential signatures.

Now querying the sign oracle twice on the same message $(M, \tilde{N})$, we obtain two signatures $\sigma_1 = (S_{1,1}, \ldots, S_{1,k})$ and $\sigma_2 = (S_{2,1}, \ldots, S_{2,k})$. With overwhelming probability, we have that $\sigma_1 \neq \sigma_2$ (i.e. the signatures are distinct). Now the signature $\sigma^* = (S_1^*, \ldots, S_k^*)$ where for all $i \in [k]$, $S_i^* := S_{1,i}^2 \cdot S_{2,i}^{-1}$ is with overwhelming probability a new valid signature on the message $(M, \tilde{N})$. □

8.2 Impossibility of a Single Group Element Signature

The following theorem proves that there is no generic-signer USPSDH scheme with signatures consisting of 1 group element that is unforgeable against a random message attack for more than 1 signing query. The only exception are one-time signatures (in which the adversary is only allowed to make a single signing query).

Theorem 10. *There is no generic-signer USPSDH scheme with 1 group element signatures that is unforgeable against a random message attack for $q > 1$ signing queries.*

Proof. Let us consider the case where the signature $\sigma = S \in \mathbb{G}$, whereas the verification key $\mathsf{vk} = (\tilde{X}_1, \ldots, \tilde{X}_n) \in \mathbb{H}^n$. The proof for the opposite case where the groups are transposed is similar.

We start by proving the following lemma which proves that it is redundant for a USPSDH scheme (for a single Diffie-Hellman pair) with 1 group element signatures to require more than one verification equation (not counting the equation needed to verify the well-formedness of the message).

Lemma 2. *One verification equation is sufficient for verifying a one-element signature.*

Proof. Such a signature scheme would have verification equations of the form of Equation (5).

$$\prod \hat{e}(S, \tilde{X}_i)^{a_{i,e}} \prod \hat{e}(M, \tilde{X}_i)^{b_{i,e}} \hat{e}(S, \tilde{N})^{c_e} \hat{e}(M, \tilde{N})^{d_e} = Z_{e_{\mathbb{T}}} \tag{5}$$

Each of those equations is a (non-trivial) equation that is linear in S. Thus, we can compute a single non-trivial equation linear in S (which uniquely determines S) by a linear combination of all those verification equations and use such an equation for verification. If there is no such linear combination of the verification equations, they must be linearly dependent which means some of them are redundant. Thus, by excluding those, we can again reduce them to a single equation that is linear in S. □

Now note that for the signature scheme to be (perfectly) correct (and publicly verifiable), the signature on the message must verify using the (fixed) verification key and (fixed) public parameters (if any). By taking the discrete logarithms of the group elements in the (single) verification equation, we can write the verification equation as

$$s(\sum_{i=1}^{n} a_i x_i + cm) + m(\sum_{i=1}^{n} b_i x_i + dm) = z \tag{6}$$

The verification equation is a linear equation in s (the discrete logarithm of the signature S). Note that such a signature is not defined if $\sum_{i=1}^{n} a_i x_i + cm = 0$. This means there exists at most one potential signature for the message. For the sake of contradiction, assume that for a message $(M, \tilde{N})$ there exists two different signatures $\sigma = S$ and $\sigma' = S'$. Since the scheme is perfectly correct, we have

$$s(\sum_{i=1}^{n} a_i x_i + cm) + m(\sum_{i=1}^{n} b_i x_i + dm) = z \tag{7}$$

$$s'(\sum_{i=1}^{n} a_i x_i + cm) + m(\sum_{i=1}^{n} b_i x_i + dm) = z \tag{8}$$

By subtracting Equation (8) from Equation (7), we get

$$(s - s')(\sum_{i=1}^{n} a_i x_i + cm) = 0, \tag{9}$$

which implies that $s = s'$ which is a contradiction.

Since the signing algorithm is generic, a signature σ_i on a message $(M_i, \tilde{N}_i)$ is of the form $\sigma_i = M_i^{\alpha} \cdot G^{\beta}$ for some (fixed) $\alpha, \beta \in \mathbb{Z}_p$. Now given signatures σ_1 and σ_2 on a pair of distinct random messages $(M_1, \tilde{N}_1), (M_2, \tilde{N}_2)$, respectively. We have $\sigma_1 = M_1^{\alpha} \cdot G^{\beta}$ and $\sigma_2 = M_2^{\alpha} \cdot G^{\beta}$. Now by computing $\sigma^* := \sigma_1^{\gamma} \cdot \sigma_2^{(1-\gamma)}$ we obtain a valid forgery on the message $(M^*, \tilde{N}^*) := \big(M_1^{\gamma} \cdot M_2^{(1-\gamma)}, \tilde{N}_1^{\gamma} \cdot \tilde{N}_2^{(1-\gamma)}\big)$ for any $\gamma \in \mathbb{Z}_p$.

To see that the forgery is a valid signature, we have

$$\begin{aligned}\sigma^* &= \sigma_1^{\gamma} \cdot \sigma_2^{(1-\gamma)} \\ &= ({M_1}^{\alpha} \cdot G^{\beta})^{\gamma} \cdot (M_2^{\alpha} \cdot G^{\beta})^{(1-\gamma)} \\ &= (M_1^{\gamma} \cdot M_2^{(1-\gamma)})^{\alpha} \cdot G^{\beta\gamma} \cdot G^{\beta(1-\gamma)} \\ &= (M_1^{\gamma} \cdot M_2^{(1-\gamma)})^{\alpha} \cdot G^{\beta}\end{aligned}$$

This implies that at least two group elements are required in the signature for the scheme to be existentially unforgeable against a random message attack that uses $q > 1$ signing queries.

Remark 4. Note that since we are considering a random message attack (which is weaker than a chosen message attack) and hence here the signer rather than the adversary chooses the messages when answering signing queries. Also, note that unlike in the Type-II bilinear group setting, in the Type-III setting there is no efficiently computable isomorphism between the groups. One way that the signer picks a random message $(M, \tilde{N})$ is, for instance, by randomly choosing $m \leftarrow \mathbb{Z}_p$ and computing $(M, \tilde{N}) := \psi(m)$, the signer then performs signing generically, i.e. without exploiting knowledge of the exponent m. Alternatively, one can envisage a separate message sampling algorithm that does the above and returns $(M, \tilde{N})$ to the signer who in turn performs the generic signing algorithm.

□

Alternative Proof for Theorem 10. Our proof below relies on eliminating some terms from the verification equation which are redundant for a generic-signer scheme as it is hard for a generic signer, who does not know the discrete logarithm of the message, to produce a non-trivial signature whose verification equation uses any of the eliminated terms. Refer to the discussion in Section 8.5 for details.

Proof. Again, let us consider the case where the signature $\sigma = S \in \mathbb{G}$, whereas the verification key $\mathsf{vk} = (\tilde{X}_1, \ldots, \tilde{X}_n) \in \mathbb{H}^n$. The proof for the opposite case where the groups are transposed is similar. We first argue that since we are only considering generic signers, it is sufficient to consider a single verification equation of the form of Equation (10) instead of Equation (5).

$$\prod \hat{e}(S, \tilde{X}_i)^{a_i} \prod \hat{e}(M, \tilde{X}_i)^{b_i} = Z_{\mathbb{T}} \tag{10}$$

Since the signing algorithm is generic, s (the discrete logarithm of the signature S) cannot have a degree > 1 of m (the discrete logarithm of the message). This means that the verification equation cannot have the monomial $\hat{e}(M, \tilde{N})^d$ where $d \neq 0$ as that would require that s have a degree > 1 of m which would require knowledge of the discrete logarithm m. So we can WLOG assume that $d = 0$. Similarly, since m cannot appear in a term in the denominator in s when viewing s as a rational function as that would also require knowledge of the discrete logarithm m, the verification equation cannot have a monomial $\hat{e}(S, \tilde{N})^c$ for $c \neq 0$ either. Therefore, we can WLOG assume that $c = 0$. We remark here that all existing structure-preserving signature schemes in all bilinear group settings conform to the assumptions we are using. Again, refer to Section 8.5 for further justification.

Thus, we end up with two cases:

- Degree of $m = 0$: This means that S is independent of the message and hence the same signature σ is valid on any other message $(M', \tilde{N}') \in \widehat{\mathbb{GH}}$ where $(M', \tilde{N}') \neq (M, \tilde{N})$.
- Degree of $m = 1$: By taking the discrete logarithms of the group elements in Equation (10), we can write the verification equation as

$$s \sum_{i=1}^{n} a_i x_i + m \sum_{i=1}^{n} b_i x_i = z \tag{11}$$

  Given signatures $\sigma_1 = S_1$ on a random message $(M_1, \tilde{N}_1)$ and $\sigma_2 = S_2$ on a random message $(M_2, \tilde{N}_2)$, by choosing $\gamma \leftarrow \mathbb{Z}_p$, we can compute a valid signature $\sigma^* = S^*$ (i.e. that satisfies the verification equation in (10)) on the message $(M^* := M_1^{\gamma} \cdot M_2^{(1-\gamma)}, \tilde{N}^* := \tilde{N}_1^{\gamma} \cdot \tilde{N}_2^{(1-\gamma)})$ by computing $S^* := S_1^{\gamma} \cdot S_2^{(1-\gamma)}$. Since the messages $(M_1, \tilde{N}_1)$ and $(M_2, \tilde{N}_2)$ are chosen uniformly at random, we have an overwhelming probability that $(M^*, \tilde{N}^*) \notin \{(M_1, \tilde{N}_1), (M_2, \tilde{N}_2)\}$ and thus σ^* is a valid forgery on a new message.

□

8.3 Lower Bound on the Size of the Verification Key for Optimal One-Time Signatures

In Section 6 we have seen that a one-time USPSDH scheme can have signatures consisting of a single group element. Here we investigate lower bounds for the size of the verification for optimal generic-signer one-time USPSDH schemes.

We prove that a generic-signer EUF-RMA secure one-time USPSDH scheme with one element signatures must have a verification key with at least two group elements (excluding the default group generators G and $\tilde{H}$). The result proves that our (strongly existentially CMA unforgeable) construction in Section 6 is optimal in every respect. WLOG, when proving the following theorem, we assume that any public group elements (other than the group generators G and $\tilde{H}$) part of the public parameters (if any) are counted as part of the verification key.

Theorem 11. *A generic-signer one-time USPSDH scheme (with one element signatures) that is unforgeable against a random message attack must have a verification key with at least* 2 *elements.*

Proof. Let us consider the case where the signature $\sigma = S \in \mathbb{G}$ whereas the verification key $\mathsf{vk} = \tilde{X} \in \mathbb{H}$. The proof for the opposite case where the groups are transposed is similar. A USPSDH scheme with a one-element verification key and a one-element signature have a (single) verification equation (not counting the equation needed to check well-formedness of the message) of the form of Equation (12).

$$\hat{e}(S, \tilde{X})^a \hat{e}(S, \tilde{H})^b \hat{e}(M, \tilde{X})^c \hat{e}(M, \tilde{H})^d \hat{e}(S, \tilde{N})^u \hat{e}(M, \tilde{N})^v = Z_{\mathbb{T}} \tag{12}$$

Note that a generic signer computes the signature S as $S := M^{\alpha} \cdot G^{\beta}$ for some $\alpha, \beta \in \mathbb{Z}_p$. Our proof strategy is to first eliminate some terms which can not be computed by a generic signer from the verification equation in (12) which serves to simplify the proof. Note that without knowledge of the discrete logarithm of the message, it is hard for a generic signer to construct a non-trivial signature S where its discrete logarithm s contains the message m in a term in the denominator. Similarly, it is hard for a generic signer without knowledge of the discrete logarithm of the message to construct a signature that contains a term with degree > 1 in m. Therefore, we can WLOG assume that $u = v = 0$ in Equation (12). We remark here that all existing structure-preserving signature schemes (in all bilinear group settings) conform to the assumption we are making. Refer to Section 8.5 for more discussion on why such assumptions (which serve to simplify the proof) do not affect the generality of our proof.

We now show that any USPSDH scheme with a verification equation of the form of Equation (13) cannot be secure.

$$\hat{e}(S, \tilde{X})^{a} \hat{e}(S, \tilde{H})^{b} \hat{e}(M, \tilde{X})^{c} \hat{e}(M, \tilde{H})^{d} = Z_{\mathbb{T}} \tag{13}$$

Since the verification key (and the public parameters) contain only $\tilde{X}$, G, and $\tilde{H}$, we have $Z_{\mathbb{T}} = \hat{e}(G, \tilde{H})^{e} \hat{e}(G, \tilde{X})^{f}$. Note that the exponents $a, b, c, d, e, f \in \mathbb{Z}_p$ are all public and hence known to the adversary. By taking the discrete logarithms of the group elements in the verification equation, we can write the verification equation as

$$s(ax + b) + m(cx + d) = e + fx \tag{14}$$

Note here if $a = b = 0$, the equation is independent of the signature S. Similarly, if $c = d = 0$, the verification equation is independent of the message $(M, \tilde{N})$. Therefore, neither of those cases should occur as otherwise it is obvious that such a scheme is not secure. We now have four cases as follows:

- **Case** $bc \neq ad$**:** In this case, given a signature $\sigma = S$ on a random message $(M, \tilde{N})$, pick any $\alpha \leftarrow \mathbb{Z}_p \setminus \{1\}$ and let

  $$a_m := \frac{ea(\alpha - 1) - bf(\alpha - 1)}{bc - ad} \qquad \text{and} \qquad a_s := -\frac{ec(\alpha - 1) - df(\alpha - 1)}{bc - ad}$$

  By computing $\sigma^* = S^* := G^{a_s} \cdot S^{\alpha}$, one obtains a valid signature on the message $\big(M^*, \tilde{N}^*\big) := \big(G^{a_m} \cdot M^{\alpha}, \tilde{H}^{a_m} \tilde{N}^{\alpha}\big)$.
- **Case** $bc = ad \neq 0$**:** Given a signature $\sigma = S$ on a random message $(M, \tilde{N})$, pick any $\alpha \leftarrow \mathbb{Z}_p^{\times}$ and compute $\sigma^* = S^* := G^{\alpha} \cdot S$, which is a valid signature on the message $\big(M^*, \tilde{N}^*\big) := \big(G^{\frac{-b\alpha}{d}} \cdot M, \tilde{H}^{\frac{-b\alpha}{d}} \cdot \tilde{N}\big)$.

  In fact, one can forge a signature on any message $\big(M^*, \tilde{N}^*\big) := \big(G^{\alpha}, \tilde{H}^{\alpha}\big)$ for any $\alpha \in \mathbb{Z}_p$ where $\big(G^{\alpha}, \tilde{H}^{\alpha}\big) \neq (M, \tilde{N})$ by computing $\sigma^* = S^* := G^{\frac{-d\alpha}{b}} \cdot M^{\frac{d}{b}} \cdot S$.
- **Case** $bc = ad = 0$**,** $a \neq 0$ **and** $c \neq 0$**:** Here we have that $b = d = 0$. Given a signature $\sigma = S$ on a random message $(M, \tilde{N})$, $\sigma^* = S^* := G^{\frac{-c\alpha}{a}} \cdot S$ is a valid signature on the message $\big(M^*, \tilde{N}^*\big) := \big(G^{\alpha} \cdot M, \tilde{H}^{\alpha} \cdot \tilde{N}\big)$ for any $\alpha \in \mathbb{Z}_p^{\times}$.
- **Case** $bc = ad = 0$**,** $b \neq 0$ **and** $d \neq 0$**:** Here we have that $a = c = 0$. Given a signature $\sigma = S$ on a random message $(M, \tilde{N})$, $\sigma^* = S^* := G^{\frac{-d\alpha}{b}} \cdot S$ is a valid signature on the message $\big(M^*, \tilde{N}^*\big) := \big(G^{\alpha} \cdot M, \tilde{H}^{\alpha} \cdot \tilde{N}\big)$ for any $\alpha \in \mathbb{Z}_p^{\times}$.

This concludes the proof.

□

8.4 Lower Bound on the Size of the Verification Key for Optimal USPSDH Schemes

We have seen that an optimal USPSDH scheme must have two elements in the signature. We prove that our schemes in Sections 4 & 5 are also optimal w.r.t. to the size of the verification key. More precisely, we prove in the following theorem that there exists no USPSDH scheme with two element signatures and one verification equation (not counting the cost of checking the well-formedness of the message) that is unforgeable against a one-time random message attack. Again, WLOG, when proving the following theorem, we assume that any public group elements (other than the default group generators G and $\tilde{H}$) part of the public parameters (if any) are counted as part of the verification key.

Theorem 12. *There is no USPSDH scheme with two group element signatures and one PPE verification equation with one group element verification key that is unforgeable against a one-time random message attack.*

Proof. Lets consider the case where the signature $\sigma = (R, S) \in \mathbb{G}^2$ whereas the verification key $\mathsf{vk} := \tilde{X} \in \mathbb{H}$. The proof for the opposite case where the groups are transposed is similar. We first argue WLOG that a generic-signer scheme have a verification equation of the form of Equation (15).

$$\hat{e}(R, \tilde{X})^a \hat{e}(R, \tilde{N})^b \hat{e}(R, \tilde{H})^c \hat{e}(S, \tilde{X})^d \hat{e}(S, \tilde{H})^u \hat{e}(M, \tilde{X})^v \hat{e}(M, \tilde{H})^w = Z_{\mathbb{T}} \tag{15}$$

Since the signing algorithm is generic, and by using a similar argument to that used in the proof of Theorem 11, note that neither R nor S can have a degree > 1 of m (the discrete logarithm of the message). It is obvious that a scheme with signatures whose both components are independent of the message is insecure. Thus, at least one component of the signature must depend on the message. WLOG, lets assume that S depends on the message while R is independent of the message. If it is the other way around, we just need to replace the term $\hat{e}(R, \tilde{N})^b$ with $\hat{e}(S, \tilde{N})^b$ in Equation (15) and the proof is similar. If both components of the signature depend on the message, Equation (15) can be simplified by setting $b = 0$ which is a special case of the cases we prove.

Since we only have $\tilde{X}, G, \tilde{H}$ in the verification key (and the public parameters), we have $Z_{\mathbb{T}} = \hat{e}(G, \tilde{H})^e \hat{e}(G, \tilde{X})^f$. Note that the exponents $a, b, c, d, e, f, u, v, w \in \mathbb{Z}_p$ are all public and hence known to the adversary. By taking the discrete logarithms of the group elements, we can write the verification equation as

$$r(ax + bm + c) + s(dx + u) + m(vx + w) = e + fx \tag{16}$$

We start by listing 4 trivial forgery cases as follows:

1. **Case** $a = b = c = 0$**:** This means the verification equation is independent of the signature component R and thus we are back into the one-element signature case which is already proven by Theorem 11.
2. **Case** $d = u = 0$**:** This means the verification equation is independent of the signature component S and thus we are back into the one-element signature case which is already proven by Theorem 11.
3. **Case** $a = d = f = v = 0$**:** This means the verification equation is independent of the verification key (and hence the signature is independent of the signing key).
4. **Case** $b = v = w = 0$**:** This means the verification equation is independent of the message m and hence the signature is valid on any other message $m' \neq m$.

Excluding the above obvious trivial forgery cases, we can find a forgery by solving the following system of equations in the 9 unknowns $\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s$

$$\begin{aligned}
&u\alpha_s + e\gamma_s - e + b\alpha_r\alpha_m + c\alpha_r + w\alpha_m = 0 \\
&d\alpha_s + f\gamma_s - f + a\alpha_r + v\alpha_m = 0 \\
&u\beta_s - w\gamma_s + b\beta_r\alpha_m + b\alpha_r\beta_m + c\beta_r + w\beta_m = 0 \\
&u\delta_s - c\gamma_s + b\gamma_r\alpha_m + c\gamma_r = 0 \\
&d\delta_s - a\gamma_s + a\gamma_r = 0 \\
&d\beta_s - v\gamma_s + a\beta_r + v\beta_m = 0 \\
&\gamma_s - \gamma_r\beta_m = 0 \\
&\beta_r\beta_m = 0
\end{aligned}$$

This is a system of 8 equations in 9 unknowns. In particular, we get two different family of solutions depending on whether $\beta_m = 0$ (in which case we obtain forgeries knowing the verification key only, i.e. without making any signing queries) or $\beta_m \neq 0$ in which case the forgery requires making a single random-message signing query. In the first case (which we refer to hereafter as type I forgery), we obtain a solution $\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \Big(\mu, 0, \frac{de - fu + uv\mu - dw\mu}{bd\mu + cd - au}, 0, 0, \frac{bf\mu + cf - bv\mu^2 - cv\mu + aw\mu - ae}{bd\mu + cd - au}, 0, 0, 0\Big)$ for any $\mu \in \mathbb{Z}_p$. Thus, we obtain a forgery of the form $\sigma^* = (R^*, S^*) := \big(G^{\alpha_r}, G^{\alpha_s}\big)$ on the message

$(M^*, \tilde{N}^*) := \big(G^{\mu}, \tilde{H}^{\mu}\big)$. This is a valid forgery as long as we can find μ such that $bd\mu + cd - au \neq 0$. We will deal with the latter case below.

In the second case (which we refer to hereafter as type II forgery), given a signature $\sigma = (R, S)$ on a random message $(M, \tilde{N})$, we obtain a solution of the form

$$\begin{aligned}&\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \\ &\qquad \Big(\mu, \frac{bd\mu + cd - au}{cd - au}, \frac{uv\mu - dw\mu}{bd\mu + cd - au}, 0, \frac{cd - au}{bd\mu + cd - au}, \frac{aw\mu - cv\mu - bv\mu^2}{bd\mu + cd - au}, \frac{-bv\mu}{cd - au}, 1, \frac{ab\mu}{bd\mu + cd - au}\Big),\end{aligned}$$

which allows us to compute a forgery $\sigma^* = (R^*, S^*) := \big(G^{\alpha_r} \cdot M^{\beta_r} \cdot R^{\gamma_r}, G^{\alpha_s} \cdot M^{\beta_s} \cdot S^{\gamma_s} \cdot R^{\delta_s}\big)$ on $(M^*, \tilde{N}^*) := \big(G^{\alpha_m} \cdot M^{\beta_m}, \tilde{H}^{\alpha_m} \cdot \tilde{N}^{\beta_m}\big)$ for any $\mu \in \mathbb{Z}_p$ as long as $(\alpha_m, \beta_m) \neq (0, 1)$ to ensure that $(M^*, \tilde{N}^*) \neq (M, \tilde{N})$. This case gives us a valid forgery providing we can find such μ satisfying $bd\mu + cd - au \neq 0$ and $cd \neq au$. Again, we will deal with the latter two cases below.

From the above, it is clear that we can find a forgery on a new message unless $cd = au$ and either $b = 0$ or $d = 0$, which we now address.

- Case $d = 0$ and $cd = au$: Note here that since $d = 0$, we must have $u \neq 0$ as otherwise we are in the second trivial forgery case. Since $cd = au$ and $d = 0$ it follows that $a = 0$. Note here that since $a = d = 0$ we must have that either $f \neq 0$ or $v \neq 0$ as otherwise we are in the third trivial forgery case (i.e. the verification equation is independent of the verification key). We have two cases as follows:
  - Case $v \neq 0$: We can, for example, obtain a type I forgery by computing

    $$\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \Big(\frac{f}{v}, 0, \mu, 0, 0, \frac{ev - fw - cv\mu - bf\mu}{uv}, 0, 0, 0\Big),$$

    for any $\mu \in \mathbb{Z}_p$.
  - Case $f \neq 0$: We can, for example, obtain a type II forgery by computing

    $$\begin{aligned}&\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \\ &\qquad \Big(\mu, \frac{f - v\mu}{f}, \nu, 0, 1, \frac{ev\mu - fw\mu - cf\nu - bf\mu\nu}{fu}, \frac{bv\mu\nu - bf\nu}{fu}, \frac{f - v\mu}{f}, \frac{-bf\mu - cv\mu}{fu}\Big),\end{aligned}$$

    for any $\mu \in \mathbb{Z}_p^{\times}$ and $\nu \in \mathbb{Z}_p$.
- Case $b = 0$ and $cd = au$: We deal with two subcases as follows:
  - Case $cd = au \neq 0$: Since here we have $b = 0$, it must be the case that either $v \neq 0$ or $w \neq 0$ as otherwise we are in case 4 of trivial forgery cases. Note here that we have $d \neq 0$ and $u \neq 0$, We deal with 2 subcases as follows
    - Case $uv \neq dw$: We can obtain a type I forgery by computing

      $$\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \Big(\frac{de - fu}{dw - uv}, 0, \mu, 0, 0, \frac{euv - cuv\mu + cdw\mu - fuw}{u(uv - dw)}, 0, 0, 0\Big),$$

      for any $\mu \in \mathbb{Z}_p$.

      Also, we can obtain a type II forgery by computing

      $$\begin{aligned}&\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \\ &\qquad \Big(\frac{de - de\mu - fu + fu\mu}{dw - uv}, \mu, \nu, 0, 1, \frac{fuw\mu - fuw + cdw\nu - euv\mu + euv - cuv\nu}{u(uv - dw)}, 0, \mu, \frac{c\mu - c}{u}\Big),\end{aligned}$$

      for any $\mu \in \mathbb{Z}_p \setminus \{1\}$ and $\nu \in \mathbb{Z}_p$.
    - Case $uv = dw$: Note here that $v \neq 0$, $w \neq 0$, and $d \neq 0$. We can obtain a type II forgery by computing

      $$\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \Big(\mu, \nu, \xi, 0, \frac{1}{\nu}, \frac{-vw\mu - cv\xi}{dw}, \frac{v - v\nu}{d}, 1, \frac{cv\nu - cv}{dw\nu}\Big),$$

      for any $(\mu, \nu) \in \mathbb{Z}_p \times \mathbb{Z}_p^{\times} \setminus \{(0, 1)\}$ and $\xi \in \mathbb{Z}_p$.

- Case $cd = au = 0$: If $d = u = 0$ we are in case 2 of the trivial forgeries. If $c = a = 0$, we are in case 1 of trivial forgeries (i.e. the one-element signature case). We are left with two cases as follows:
  - Case $c = u = 0$: We deal with 2 subcases as follows:
    - Case $w \neq 0$: We can obtain a type I forgery by computing

      $$\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \Big(\frac{e}{w}, 0, \mu, 0, 0, \frac{fw - ev - aw\mu}{dw}, 0, 0, 0\Big),$$

      for any $\mu \in \mathbb{Z}_p$.
      Also, we can obtain a type II forgery by computing

      $$\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \Big(\frac{e - e\mu}{w}, \mu, \nu, 0, 1, \frac{fw - fw\mu - aw\nu - ev + ev\mu}{dw}, 0, \mu, \frac{a\mu - a}{d}\Big),$$

      for any $\mu \in \mathbb{Z}_p \setminus \{1\}$, $\nu \in \mathbb{Z}_p$.
    - Case $w = 0$: Note here that $d \neq 0$ as otherwise we are in case 2 of trivial forgeries. We can obtain a type II forgery by computing

      $$\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \Big(\mu, \nu, \xi, 0, \frac{1}{\nu}, \frac{-a\xi - v\mu}{d}, \frac{v - v\nu}{d}, 1, \frac{a\nu - a}{\nu d}\Big),$$

      for any $(\mu, \nu) \in \mathbb{Z}_p \times \mathbb{Z}_p^{\times} \setminus \{(0, 1)\}$, $\xi \in \mathbb{Z}_p$.
  - Case $d = a = 0$: Note here that $u \neq 0$ as otherwise we are in case 2 of trivial forgeries. We have two subcases as follows:
    - Case $v \neq 0$: We can obtain a type I forgery by computing

      $$\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \Big(\frac{f}{v}, 0, \mu, 0, 0, \frac{ev - fw - cv\mu}{uv}, 0, 0, 0\Big),$$

      for any $\mu \in \mathbb{Z}_p$. Also, we can obtain a type II forgery by computing

      $$\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \Big(\frac{f - f\mu}{v}, \mu, \nu, 0, 1, \frac{ev - cv\nu - ev\mu - fw + fw\mu}{uv}, 0, \mu, \frac{c\mu - c}{u}\Big),$$

      for any $\mu \in \mathbb{Z}_p \setminus \{1\}$ and $\nu \in \mathbb{Z}_p$.
    - Case $v = 0$: We can obtain a type II forgery by computing

      $$\big(\alpha_m, \beta_m, \alpha_r, \beta_r, \gamma_r, \alpha_s, \beta_s, \gamma_s, \delta_s\big) := \Big(\mu, \nu, \xi, 0, \frac{1}{\nu}, \frac{-c\xi - w\mu}{u}, \frac{w - w\nu}{u}, 1, \frac{c\nu - c}{u\nu}\Big),$$

      for any $(\mu, \nu) \in \mathbb{Z}_p \times \mathbb{Z}_p^{\times} \setminus \{(0, 1)\}$ and $\xi \in \mathbb{Z}_p$.

This concludes the proof. □

8.5 Further Discussion

In some of our lower bound proofs, we relied on eliminating some terms (i.e. pairings) from the verification equation. As mentioned earlier, all existing structure-preservation signature schemes in all 3 bilinear group settings conform to those assumptions. In this section, we provide further justification that such assumptions are inevitable.

As an example, consider a one-time USPSDH scheme with a one-element signature $\sigma = S \in \mathbb{G}$, a one-element verification key $\tilde{X} \in \mathbb{H}$ and a single verification equation of the form of Equation (17)

$$\hat{e}(S, \tilde{X})\hat{e}(S, \tilde{M}) = \hat{e}(G, \tilde{H}) \tag{17}$$

Note that when verifying a signature in the above example, one also needs to verify that the message $(M, \tilde{N})$ is well-formed, i.e. $(M, \tilde{N}) \in \widehat{\mathbb{G}\mathbb{H}}$. The above example can in some sense be viewed as a USPSDH scheme analogous to the weak Boneh-Boyen Signature [14]. The example above is a secure one-time USPSDH scheme against a random message attack in the generic group model (as long as the verification key is not given in $\mathbb{G}$). As can be seen, the verification key of such a scheme is a single group element, however, such a scheme does not contradict our lower bound proofs as there is no way a generic signer can produce the signature σ without knowing the discrete logarithm of the message. We note here that one can also use, for example, a similar argument against the lower bound proofs for the Type-II bilinear group setting in [7]. For instance, Theorem 4 in [7] proved that a Type-II structure-preserving signature scheme for messages in $\mathbb{H}$ with one-element signatures cannot have a verification key with a single group element. For the sake of illustration, consider a scheme in the Type-II setting for messages $\tilde{M} \in \mathbb{H}$ with a signature $\tilde{\sigma} = \tilde{S} \in \mathbb{H}$, a verification key $X \in \mathbb{G}$ and a single verification equation of the form of Equation (18)

$$\hat{e}(X, \tilde{S})\hat{e}(\Psi(\tilde{M}), \tilde{S}) = \hat{e}(G, \tilde{H}), \tag{18}$$

where $\Psi : \mathbb{H} \to \mathbb{G}$ is an isomorphism. Such a scheme is a secure one-time structure-preserving signature scheme against a random message attack in the Type-II setting in the generic group model. However, again, this should not be considered as a contradiction to Theorem 4 in [7] as it is infeasible for a generic signer to produce such signatures without knowing the discrete logarithm of the message.

As a second example, consider a USPSDH scheme with a single group element signature $\sigma = S \in \mathbb{G}$, a verification key $\tilde{X}, \tilde{Y}, \tilde{Z} \in \mathbb{H}$ and a single verification equation of the form of Equation (19)

$$\hat{e}(S, \tilde{Z}) = \hat{e}(G, \tilde{X})\hat{e}(M, \tilde{Y})\hat{e}(M, \tilde{N}) \tag{19}$$

Such a scheme is a secure USPSDH scheme against a random message attack in the generic group model. Nevertheless, this does not contradict our results as such a signature cannot be produced by a generic signer who does not know the discrete logarithm of the message $(M, \tilde{N})$.

Again, one can give a similar counterexample for the Type-II setting proved in [7]. Consider a structure-preserving signature scheme in the Type-II setting for messages $\tilde{M} \in \mathbb{H}$ with a single group element signature $\tilde{\sigma} = \tilde{S}$, a verification key $X, Y, Z \in \mathbb{G}$ and a single verification equation of the form of Equation (20)

$$\hat{e}(Z, \tilde{S}) = \hat{e}(X, \tilde{H})\hat{e}(Y, \tilde{M})\hat{e}(\Psi(M), \tilde{M}) \tag{20}$$

Such a scheme is a secure scheme against a random message attack in the generic group model in the Type-II setting. However, since signatures of this scheme cannot be produced by a generic signer, such a scheme should not be regarded as a contradiction to Theorem 5 in [7].

9 Optimal CMA-Secure Partially Structure-Preserving Signature Scheme for a Vector of Messages

We do not know how to construct a USPSDH scheme with optimal signatures (i.e. two group elements) and a single verification equation that can sign a vector of Diffie-Hellman pairs. However, we give here an optimal signature scheme (with two group element signatures and a single verification equation) that simultaneously signs a Diffie-Hellman pair and a vector from $\mathbb{Z}_p^k$, i.e. the message space of the scheme is $\widehat{\mathbb{GH}} \times \mathbb{Z}_p^k$. We call such a variant *partially structure-preserving* since other than allowing some components of the messages to be signed to not be group elements, the scheme satisfies the rest of the conditions required by the definition of structure-preserving signatures. In particular, the signatures, the verification key and part of the message are all group elements, and verification only requires the evaluation of pairing-product equations.

Given the description of Type-III bilinear groups $\mathcal{P}$ output by $\mathsf{BGSetup}(1^\lambda)$, the scheme is as follows:

- $\mathsf{KeyGen}(\mathcal{P})$: Select $x, y_1, \ldots, y_k, z \leftarrow \mathbb{Z}_p^\times$. Set $\tilde{X} := \tilde{H}^x$, $\tilde{Y}_i := \tilde{H}^{y_i}$ for all $i \in [k]$, $\tilde{Z} := \tilde{H}^z$. Set $\mathsf{sk} := (x, y_1, \ldots, y_k, z)$ and $\mathsf{vk} := (\tilde{X}, \tilde{Y}_1, \ldots, \tilde{Y}_k, \tilde{Z})$.
- $\mathsf{Sign}\Big(\mathsf{sk}, \big((M, \tilde{N}), \boldsymbol{u} = (u_1, \ldots, u_k)\big)\Big)$: To sign a Diffie-Hellman pair $(M, \tilde{N}) \in \widehat{\mathbb{GH}}$ and a vector $\boldsymbol{u} = (u_1, \ldots, u_k) \in \mathbb{Z}_p^k$, select $r \leftarrow \mathbb{Z}_p^\times$, and set $R := G^r$, $S := \big(M \cdot G^{x + \sum_{i=1}^k u_i y_i}\big)^{\frac{r}{z}}$. Return $\sigma := (R, S) \in \mathbb{G}^2$.
- $\mathsf{Verify}\Big(\mathsf{vk}, \big((M, \tilde{N}), \boldsymbol{u}\big), \sigma = (R, S)\Big)$: Return 1 iff $R \in \mathbb{G}^\times$, $(M, \tilde{N}) \in \widehat{\mathbb{GH}}$, and the following holds:

$$\hat{e}(S, \tilde{Z}) = \hat{e}(R, \tilde{N})\hat{e}(R, \tilde{X}) \prod_{i=1}^{k} \hat{e}(R, \tilde{Y}_i^{u_i})$$

- $\mathsf{Randomize}\Big(\mathsf{vk}, \big((M, \tilde{N}), \boldsymbol{u}\big), \sigma = (R, S)\Big)$: Select $r' \leftarrow \mathbb{Z}_p^{\times}$, and set $R' := R^{r'}$, $S' := S^{r'}$. Return $\sigma' := (R', S')$.

Correctness of the scheme follows by inspection and is straightforward to verify. The scheme is perfectly randomizable as the distribution of re-randomized signatures is identical to that of fresh signatures on the same message vector.

Theorem 13. *The partially structure-preserving signature scheme is existentially weakly unforgeable against a chosen-message attack in the generic group model.*

Proof. Public elements in $\mathbb{H}$ are $\tilde{H}$, $\tilde{X}$, $\{\tilde{Y}\}_{i=1}^{k}$, and $\tilde{Z}$ which correspond to the discrete logarithms 1, x, $\{y_i\}_{i=1}^{k}$, and z, respectively. After q signing queries, (m^*, n^*), which is the discrete logarithm of the forged Diffie-Hellman pair $(M^*, \tilde{N}^*)$, must be of the form

$$\begin{aligned} n^* &= a_n + b_n x + \sum_{i=1}^{k} c_{n,i} y_i + d_n z \\ m^* &= a_m + \sum_{i=1}^{q} b_{m_i} r_i + \sum_{i=1}^{q} c_{m_i} (\frac{r_i m_i}{z} + \frac{r_i x}{z} + \frac{r_i \sum_{j=1}^{k} u_{i,j} y_j}{z}) \end{aligned}$$

Since we must have $m^* = n^*$ for the forgery to be a valid element of $\widehat{\mathbb{GH}}$, we have

$$m^* = n^* = a_n = a_m$$

Similarly, the signature (R^*, S^*) have the form

$$\begin{aligned} r^* &= a_r + \sum_{i=1}^{q} b_{r_i} r_i + \sum_{i=1}^{q} c_{r_i} (\frac{r_i m_i}{z} + \frac{r_i x}{z} + \frac{r_i \sum_{j=1}^{k} u_{i,j} y_j}{z}) \\ s^* &= a_s + \sum_{i=1}^{q} b_{s_i} r_i + \sum_{i=1}^{q} c_{s_i} (\frac{r_i m_i}{z} + \frac{r_i x}{z} + \frac{r_i \sum_{j=1}^{k} u_{i,j} y_j}{z}) \end{aligned}$$

For the forgery to be a valid signature, s^* and r^* must satisfy $s^* z = r^* m^* + r^* x + r^* \sum_{j=1}^{k} u_j^* y_j$. So we must have

$$\begin{aligned} &\Big(a_s + \sum_{i=1}^{q} b_{s_i} r_i + \sum_{i=1}^{q} c_{s_i} (\frac{r_i m_i}{z} + \frac{r_i x}{z} + \frac{r_i \sum_{j=1}^{k} u_{i,j} y_j}{z})\Big) z \\ &= \Big(a_r + \sum_{i=1}^{q} b_{r_i} r_i + \sum_{i=1}^{q} c_{r_i} (\frac{r_i m_i}{z} + \frac{r_i x}{z} + \frac{r_i \sum_{j=1}^{k} u_{i,j} y_j}{z})\Big) m^* \\ &+ \Big(a_r + \sum_{i=1}^{q} b_{r_i} r_i + \sum_{i=1}^{q} c_{r_i} (\frac{r_i m_i}{z} + \frac{r_i x}{z} + \frac{r_i \sum_{j=1}^{k} u_{i,j} y_j}{z})\Big) x \\ &+ \Big(a_r + \sum_{i=1}^{q} b_{r_i} r_i + \sum_{i=1}^{q} c_{r_i} (\frac{r_i m_i}{z} + \frac{r_i x}{z} + \frac{r_i \sum_{j=1}^{k} u_{i,j} y_j}{z})\Big) \sum_{j=1}^{k} u_j^* y_j \end{aligned}$$

Thus, we must have

$$\begin{aligned} a_s z + \sum_{i=1}^{q} b_{s_i} r_i z + &\sum_{i=1}^{q} c_{s_i} (r_i m_i + r_i x + r_i \sum_{j=1}^{k} u_{i,j} y_j) \\ &= \Big(a_r + \sum_{i=1}^{q} b_{r_i} r_i + \sum_{i=1}^{q} c_{r_i} (\frac{r_i m_i}{z} + \frac{r_i x}{z} + \frac{r_i \sum_{j=1}^{k} u_{i,j} y_j}{z})\Big) m^* \\ &+ \Big(a_r + \sum_{i=1}^{q} b_{r_i} r_i + \sum_{i=1}^{q} c_{r_i} (\frac{r_i m_i}{z} + \frac{r_i x}{z} + \frac{r_i \sum_{j=1}^{k} u_{i,j} y_j}{z})\Big) x \\ &+ \Big(a_r + \sum_{i=1}^{q} b_{r_i} r_i + \sum_{i=1}^{q} c_{r_i} (\frac{r_i m_i}{z} + \frac{r_i x}{z} + \frac{r_i \sum_{j=1}^{k} u_{i,j} y_j}{z})\Big) \sum_{j=1}^{k} u_j^* y_j \end{aligned}$$

Note that there is no term in $\frac{r_i x^2}{z}$ on the left-hand side so we must have $c_{r_i} = 0$ for all i. Also, there is no term in z or $r_i z$ on the right-hand side so we must have $a_s = 0$ and $b_{s_i} = 0$ for all i. Thus, we have

$$\begin{aligned}\sum_{i=1}^{q} c_{s_i}(r_i m_i + r_i x + r_i \sum_{j=1}^{k} u_{i,j} y_j) \\ = \Big(a_r + \sum_{i=1}^{q} b_{r_i} r_i\Big) m^* + \Big(a_r + \sum_{i=1}^{q} b_{r_i} r_i\Big) x + \Big(a_r + \sum_{i=1}^{q} b_{r_i} r_i\Big) \sum_{j=1}^{k} u_j^* y_j\end{aligned}$$

There is no term in x on the left-hand side so we must have $a_r = 0$ and thus we have

$$\sum_{i=1}^{q} c_{s_i}(r_i m_i + r_i x + r_i \sum_{j=1}^{k} u_{i,j} y_j) = \sum_{i=1}^{q} b_{r_i} r_i m^* + \sum_{i=1}^{q} b_{r_i} r_i x + \sum_{i=1}^{q} b_{r_i} r_i \sum_{j=1}^{k} u_j^* y_j$$

The monomial $r_i x$ implies $c_{s_i} = b_{r_i}$ for all i. Since we require that $R^* \in \mathbb{G}^{\times}$, we must have $r^* \neq 0$ and therefore we must have at least a single value of $c_{s_i} = b_{r_i} \neq 0$. Now the monomial r_i implies $c_{s_i} m_i = b_{r_i} m^*$ which means $m^* = m_i$ for some $i \in [q]$. Also, the monomial $r_i y_j$ implies $c_{s_i} u_{i,j} = b_{r_i} u_j^*$ and hence we have $u_j^* = u_{i,j}$ for all $j \in [k]$. This means the forgery is on a vector $\big((M^*, \tilde{N}^*), \boldsymbol{u}^*\big)$ which was queried to the sign oracle and thus is not a forgery.

□

10 Efficient Randomizable weakly Blind Signatures without Random Oracles

Bernhard et al. [12] defined randomizable weakly blind signature schemes as one of the building blocks for their generic construction of Direct Anonymous Attestation (DAA) protocols [15].

```
$\mathsf{Setup_{BS}}(1^\lambda)$
    $\mathcal{P} \leftarrow \mathsf{BGSetup}(1^\lambda)$.
    $(\mathsf{crs}_i, \mathsf{xk}_i) \leftarrow \mathsf{GSSetup}(\mathcal{P})$ for $i = 1, 2$.
    Return $\mathsf{param_{BS}} := (\mathcal{P}, \mathsf{crs}_1, \mathsf{crs}_2)$.

$\mathsf{KeyGen_{BS}}(\mathsf{param_{BS}})$
    $x, y \leftarrow \mathbb{Z}_p$. $\tilde{X} := \tilde{H}^x; \tilde{Y} := \tilde{H}^y$.
    Return $\big(\mathsf{sk_{BS}} := (x, y), \mathsf{vk_{BS}} := (\tilde{X}, \tilde{Y})\big)$.

$\mathsf{Request^0_{BS}}(\mathsf{vk_{BS}}, (M, \tilde{N}))$
    $\pi \leftarrow \mathsf{GSProve}\Big(\mathsf{crs}_1, \{\tilde{N}\} : M \in \mathcal{L}_{\mathrm{U}}\Big)$.
    Return $(\rho_0 := (M, \pi), \mathsf{st}^0_R := (M, \tilde{N}))$.

$\mathsf{Issue^1_{BS}}(\mathsf{sk_{BS}}, \rho_0)$
    Parse $\rho_0$ as $(M, \pi)$.
    If $\mathsf{GSVerify}(\mathsf{crs}_1, M \in \mathcal{L}_{\mathrm{U}}, \pi) = 0$, Return $\perp$.
    $r \leftarrow \mathbb{Z}_p^\times$; $R := G^r$; $S := (G^x \cdot M)^{\frac{r}{y}}$.
    $\Omega \leftarrow \mathsf{GSProve}(\mathsf{crs}_2, \{\tilde{R}\} : (R, S, M) \in \mathcal{L}_{\mathrm{S}})$.
    Return $\beta_1 := \big((R, S), \Omega\big)$.

$\mathsf{Request^1_{BS}}(\mathsf{vk_{BS}}, \beta_1, \mathsf{st}^0_R)$
    Parse $\beta_1$ as $((R, S), \Omega)$.
    Parse $\mathsf{st}^0_R$ as $(M, \tilde{N})$.
    Return $\perp$ if any of the following hold:
      ∘ $R = 1_{\mathbb{G}}$.
      ∘ $\mathsf{GSVerify}(\mathsf{crs}_2, (R, S, M) \in \mathcal{L}_{\mathrm{S}}, \Omega) = 0$.
    Return $\sigma \leftarrow \mathsf{Randomize_{BS}}\big(\mathsf{vk_{BS}}, (R, S)\big)$.

$\mathsf{Verify_{BS}}(\mathsf{vk_{BS}}, (M, \tilde{N}), (R, S))$
    If $R = 1_{\mathbb{G}}$ or $\hat{e}(S, \tilde{Y}) \neq \hat{e}(R, \tilde{X})\hat{e}(R, \tilde{N})$
        Then Return 0.
    Else
        Return 1.

$\mathsf{Randomize_{BS}}(\mathsf{vk_{BS}}, \sigma)$
    Parse $\sigma$ as $(R, S)$.
    $r' \leftarrow \mathbb{Z}_p^\times$; $R' := R^{r'}$; $S' := S^{r'}$.
    Return $(R', S')$.
```

Fig. 3. Our Weakly Blind Signature Scheme

DAA protocols are outside of the scope of this paper but for the record we show that combining our publicly re-randomizable scheme from Section 5 with the SXDH-based instantiation of the Groth-Sahai proof system yields more efficient instantiations of randomizable weakly blind signature schemes than those used in [11, 29], which were used for constructing the first instantiations of DAA which do not rely on random oracles. The obtained weakly blind signature (See Fig. 3) yields signatures of size $2|\mathbb{G}|$ and require only 1 PPE equations (2 pairings in total) to verify. That our instantiation is more efficient than those in [11, 29] is obvious as the underlying signature scheme we use is more efficient than those used [11, 29].

In the construction, we use the following languages for the zero-knowledge proofs in the signing protocol for the user and signer, respectively:

$$\begin{aligned} \mathcal{L}_{\mathrm{U}} &: \Big\{ \big(M, \tilde{N}\big) \ : \ \hat{e}(G, \underline{\tilde{N}}) = \hat{e}(M, \tilde{H}) \Big\} \\ \mathcal{L}_{\mathrm{S}} &: \Big\{ \big((R, S, M), \tilde{R}\big) \ : \ \hat{e}(G, \underline{\tilde{R}}) = \hat{e}(R, \tilde{H}) \quad \wedge \quad \hat{e}(S, \tilde{Y}) = \hat{e}(R, \tilde{X})\hat{e}(M, \underline{\tilde{R}}) \Big\} \end{aligned}$$

Note that all equations in the above are of the form that one gets zero-knowledge Groth-Sahai proofs for. For simplicity, the languages above do not spell out the details of the auxiliary simulation-enabling equations.

The proof of the following theorem is provided in Appendix B.

Theorem 14. *Assuming the SXDH assumption holds and the structure-preserving signature scheme from Section 5 is existentially unforgeable, the weakly blind signature scheme in Fig. 3 is secure.*

References

1. M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki and M. Ohkubo. Constant-Size Structure-Preserving Signatures: Generic Constructions and Simple Assumptions. In *ASIACRYPT 2012*, Springer LNCS 7658, 4–24, 2012.
2. M. Abe, B. David, M. Kohlweiss, R. Nishimaki and M. Ohkubo. Tagged One-Time Signatures: Tight Security and Optimal Tag Size. In *PKC 2013*, Springer LNCS 7778, 312–331, 2013.
3. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *CRYPTO 2010*, Springer LNCS 6223, 209–236, 2010.
4. M. Abe, J. Groth, K. Haralambiev and M. Ohkubo. Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups. In *CRYPTO 2011*, Springer LNCS 6841, 649–666, 2011.
5. M. Abe, J. Groth and M. Ohkubo. Separating Short Structure-Preserving Signatures from Non-interactive Assumptions. In *ASIACRYPT 2011*, Springer LNCS 7073, 628–646, 2011.
6. M. Abe, J. Groth, M. Ohkubo and M. Tibouchi. Unified, Minimal and Selectively Randomizable Structure-Preserving Signatures. In TCC 2014, Springer LNCS 8349, 688–712, 2014.
7. M. Abe, J. Groth, M. Ohkubo and M. Tibouchi. Structure-Preserving Signatures from Type II Pairings. In CRYPTO 2014, Springer LNCS 8616, 390–407, 2014.
8. M. Abe, M. Kohlweiss, M. Ohkubo and M. Tibouchi. Fully Structure-Preserving Signatures and Shrinking Commitments. In *EUROCRYPT 2015*, Springer LNCS 9057, 35–65, 2015.
9. N. Attrapadung, B. Libert and T. Peters. Computing on authenticated data: new privacy definitions and constructions. In *ASIACRYPT 2012*, Springer LNCS 7658, 367–385, 2012.
10. G. Barthe, E. Fagerholm, D. Fiore, A. Scedrov, B. Schmidt and M. Tibouchi. Strongly-Optimal Structure Preserving Signatures from Type II Pairings: Synthesis and Lower Bounds. In *PKC 2015*, Springer LNCS 9020, 355–376, 2015.
11. D. Bernhard, G. Fuchsbauer and E. Ghadafi. Efficient Signatures of Knowledge and DAA in the Standard Model. In *ACNS 2013*, Springer LNCS 7954, 518–533, 2013.
12. D. Bernhard, G. Fuchsbauer, E. Ghadafi, N.P. Smart and B. Warinschi. Anonymous attestation with user-controlled linkability. In International Journal of Information Security, volume 12(3), 219–249, 2013.
13. R. Barbulescu, P. Gaudry, A. Joux and E. Thom. A Heuristic Quasi-Polynomial Algorithm for Discrete Logarithm in Finite Fields of Small Characteristic. In *EUROCRYPT 2014*, Springer LNCS 8441, 1–16, 2014.
14. D. Boneh and X. Boyen. Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups. In Journal of Cryptology, volume 21(2), 149–177, 2008.
15. E. Brickell, J. Camenisch and L. Chen. Direct anonymous attestation. *CCS 2004*, ACM, 132–145, 2004.
16. J. Camenisch, M. Dubovitskaya and K. Haralambiev. Efficient Structure-Preserving Signature Scheme from Standard Assumptions. In *SCN 2012*, Springer LNCS 7485, 76–94, 2012.
17. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO 2004*, Springer LNCS 3152, 56–72, 2004.
18. M. Chase and M. Kohlweiss. A New Hash-and-Sign Approach and Structure-Preserving Signatures from DLIN. In *SCN 2012*, Springer LNCS 7485, 131–148, 2012.

19. S. Chatterjee, D. Hankerson, E. Knapp and A. Menezes. Comparing two pairing-based aggregate signature schemes. *Des. Codes Cryptography*, **55**, 141–167. May 2010.
20. S. Chatterjee and A. Menezes. Type 2 Structure-Preserving Signature Schemes Revisited. In *Cryptology ePrint Archive, Report 2014/635*. `http://eprint.iacr.org/2014/635.pdf`.
21. D. Chaum. Blind signatures for untraceable payments. In *CRYPTO 1982*, Springer LNCS, 199–203, 1983.
22. D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT 1991*, Springer LNCS 547, 257–265, 1991.
23. T. ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In IEEE Transactions on Information Theory, volume 31(4), 469–472, 1985.
24. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO 1986*, Springer LNCS 263, 186–194, 1986.
25. G. Fuchsbauer. Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. In *Cryptology ePrint Archive, Report 2009/320*. `http://eprint.iacr.org/2009/320.pdf`.
26. G. Fuchsbauer. Commuting signatures and verifiable encryption. In *EUROCRYPT 2011*, Springer LNCS 6632, 224–245, 2011.
27. S. Galbraith, K. Paterson and N.P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, **156**, 3113–3121, 2008.
28. E. Ghadafi. Formalizing Group Blind Signatures and Practical Constructions without Random Oracles. In *ACISP 2013*, Springer LNCS 7959, 330–346, 2013.
29. E. Ghadafi. Short Structure-Preserving Signatures. In *CT-RSA 2016*, Springer LNCS 9610, 305–321, 2016.
30. R. Granger, T. Kleinjung and J. Zumbrägel. Breaking '128-bit Secure' Supersingular Binary Curves (or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$). In *CRYPTO 2014*, Springer LNCS 8617, 126–145, 2014.
31. M. Green and S. Hohenberger. Universally Composable Adaptive Oblivious Transfer. In *ASIACRYPT 2008*, Springer LNCS 5350, 179–197, 2008.
32. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *ASIACRYPT 2006*, Springer LNCS 4284, 444–459, 2006.
33. J. Groth. Efficient Fully Structure-Preserving Signatures for Large Messages . In *Cryptology ePrint Archive, Report 2015/824*. `http://eprint.iacr.org/2015/824.pdf`.
34. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In SIAM Journal on Computing, volume 41(5), 1193–1232, 2012.
35. D. Hofheinz and T. Jager. Tightly Secure Signatures and Public-Key Encryption. In *CRYPTO 2012*, Springer LNCS 7417, 590–607, 2012.
36. E. Kiltz, J. Pan and H. Wee. Structure-Preserving Signatures from Standard Assumptions, Revisited. In *CRYPTO 2015*, Springer LNCS 9216, 275–295, 2015.
37. B. Libert, T. Peters and M. Yung. Scalable Group Signatures with Revocation. In *EUROCRYPT 2012*, Springer LNCS 7237, 609–627, 2012.
38. B. Libert, T. Peters and M. Yung. Short Group Signatures via Structure-Preserving Signatures: Standard Model Security from Simple Assumptions. In *CRYPTO 2015*, Springer LNCS 9216, 296–316, 2015.
39. H.K. Maji, M. Prabhakaran and M. Rosulek. Attribute-Based Signatures. In *CT-RSA 2011*, Springer LNCS 6558, 376–392, 2011.
40. U. Maurer. Abstract models of computation in cryptography. In *Cryptography and Coding 2005*, Springer LNCS 3796, 1–12, 2005.
41. D. Pointcheval and O. Sanders. Short Randomizable Signatures. In *CT-RSA 2016*, Springer LNCS 9610, 111–126, 2016.
42. V. Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In *EUROCRYPT 1997*, Springer LNCS 3152, 41–55, 1997.
43. B. Waters. Efficient identity-based encryption without random oracles. *EUROCRYPT 2005*, Springer LNCS 3494, 114–127, 2005.

A Proof of Theorem 8

Proof. We show that the linear combinations the generic adversary can produce out of the combinations of the signatures' elements, verification key elements and public parameters in each of the source groups, cannot correspond to Laurent polynomials representing a valid forgery. Public elements in $\mathbb{H}$ are $\tilde{H}, \tilde{X}_1, \ldots, \tilde{X}_k, \tilde{Y}$ which correspond to the discrete logairthms 1, x_1, …, x_k and y, respectively. The message $\left(M_1^*, \ldots, M_k^*\right)$ on which the adversary forges a signature $\sigma^* = S^*$ can only be of the form

$$m_i^* = a_{m_i} + b_{m_i}\left(\frac{x_1}{y} + \frac{m_1}{y} + \sum_{j=2}^{k} \frac{m_j x_j}{y}\right), \text{ for } i = 1, \ldots, k.$$

Similarly, the signature $\sigma^* = S^*$ must have the form

$$s^* = a_s + b_s\big(\frac{x_1}{y} + \frac{m_1}{y} + \sum_{i=2}^{k} \frac{m_i x_i}{y}\big)$$

For the forgery to be a valid signature, s^* must satisfy $s^* y = x_1 + m_1^* + \sum_{i=2}^{k} m_i^* x_i$. Therefore, we must have

$$\Big(a_s + b_s\big(\frac{x_1}{y} + \frac{m_1}{y} + \sum_{i=2}^{k} \frac{m_i x_i}{y}\big)\Big)y = x_1 + m_1^* + \sum_{i=2}^{k} m_i^* x_i$$

So we must have

$$a_s y + b_s\big(x_1 + m_1 + \sum_{i=2}^{k} m_i x_i\big) = x_1 + m_1^* + \sum_{i=2}^{k} m_i^* x_i.$$

There is no term in y on the right-hand side so we must have $a_s = 0$, Thus, we have

$$b_s\big(x_1 + m_1 + \sum_{i=2}^{k} m_i x_i\big) = x_1 + m_1^* + \sum_{i=2}^{k} m_{i-1}^* x_i.$$

By the monomial x_1, we have $b_s = 1$. For the two sides to be equal, we must have $b_s m_1 = m_1^*$ and $b_s m_i x_i = m_i^* x_i$ for all $i = 2, \ldots, k$. Since we have $b_s = 1$, it means we must have $m_1^* = m_1$ and $m_i = m_i^*$ for all $i = 2, \ldots, k$. This means the forgery is on the same vector queried to the sign oracle.

□

B Proof of Theorem 14

Correctness of the construction follows from that of the signature scheme and the perfect completeness of Groth-Sahai proofs. Unforgeability and weak blindness are proven by the following two lemmata, respectively.

Lemma 3. *The weakly blind signature scheme in Fig. 3 is unforgeable if the structure-preserving signature scheme in Section 5 is existentially unforgeable, $\mathcal{NIZK}_1$ (used by the user to produce π) is sound and $\mathcal{NIZK}_2$ (used by the signer to produce Ω) is zero-knolwedge.*

Proof. We instantiate crs_1 used for $\mathcal{NIZK}_1$ as a binding crs and hence $\mathcal{NIZK}_1$ is perfectly sound, whereas crs_2 is instantiated as a hiding string and hence we can simulate proof Ω. By the security of $\mathcal{NIZK}_2$, an adversary has a negligible advantage in distinguishing between the two settings. Using an adversary $\mathcal{A}$ that breaks the unforgeability of the blind signature scheme, we construct an adversary $\mathcal{B}$ against the unforgeability of the structure-preserving signature scheme.

$\mathcal{B}$ gets the verification key $\mathsf{vk} = (\tilde{X}, \tilde{Y})$ from its game which it forwards to $\mathcal{A}$. $\mathcal{B}$ has access to a sign oracle in its game. To answer a signature query on a message, $\mathcal{B}$ uses the extraction key of $\mathcal{NIZK}_1$ to extract the witness $\tilde{N}$ and forwards $(M, \tilde{N})$ (which, by the soundness of $\mathcal{NIZK}_1$, is a valid Diffie-Hellman pair) to its sign oracle to get a signature $\sigma = (R, S)$. $\mathcal{B}$ then simulates the proof Ω (since it does know the exponent r used in the signature and hence cannot produce the element $\tilde{R}$). $\mathcal{B}$ returns $\big(\sigma = (R, S), \Omega\big)$ to $\mathcal{A}$.

Eventually, when $\mathcal{A}$ outputs its $n + 1$ message-signature pairs, $\mathcal{B}$ returns the extra pair that it did not query its oracle on as its forgery.

By the existential unforgeability of the signature scheme, we have that this only happens with a negligible probability.

This concludes the proof.

□

Lemma 4. *The weakly blind signature scheme in Fig.3 is weakly blind if $\mathcal{NIZK}_1$ is zero-knowledge, $\mathcal{NIZK}_2$ is sound and the DDH assumption holds in group $\mathbb{G}$.*

Proof. We instantiate crs_1 (used for $\mathcal{NIZK}_1$) as a hiding crs and hence we can simulate proof π. By the security of $\mathcal{NIZK}_1$, an adversary has a negligible advantage in distinguishing between the two settings. We instantiate the reference string crs_2 (used for $\mathcal{NIZK}_2$) as a binding string and hence the proof system $\mathcal{NIZK}_2$ is perfectly sound.

Using an adversary $\mathcal{A}$ against the weak blindness of the blind signature, we construct an adversary $\mathcal{B}$ that breaks the DDH assumption in group $\mathbb{G}$. $\mathcal{B}$ gets $A = G^a$, $B = G^b$ and $C = G^c$, where $a, b, c \in \mathbb{Z}_p$ are random exponents unknown to $\mathcal{B}$. $\mathcal{B}$ chooses $x, y \leftarrow \mathbb{Z}_p$ and computes $\tilde{X} := \tilde{H}^x$ and $\tilde{Y} := \tilde{H}^y$. It forwards $\mathsf{sk}_{\mathsf{BS}} := (x, y)$ and $\mathsf{vk}_{\mathsf{BS}} := (\tilde{X}, \tilde{Y})$ to $\mathcal{A}$.

$\mathcal{B}$ requests from $\mathcal{A}$ a blind signature on the message A. Note that since $\mathcal{B}$ does not know a, it simulates the proof π. $\mathcal{A}$ responds with a signature $(\sigma_0 = (R, S), \Omega)$. Since $\mathcal{NIZK}_2$ is perfectly sound, (R, S) is a valid signature on the message $(G^a, \tilde{H}^a)$, i.e. for some $r \in \mathbb{Z}_p$, we have $\sigma_0 = (G^r, G^{\frac{rx}{y}} \cdot G^{\frac{ra}{y}})$. To produce the challenge signature σ_1, $\mathcal{B}$ computes $\sigma_1 = (R^*, S^*) := (B, B^{\frac{x}{y}} \cdot C^{\frac{1}{y}})$. $\mathcal{B}$ returns $\big(\sigma_0 = \big(R, S\big), \sigma_1 = \big(R^*, S^*\big)\big)$ to $\mathcal{A}$. The advantage of $\mathcal{B}$ in breaking the DDH assumption in $\mathbb{G}$ is the same as that of $\mathcal{A}$ winning the weak blindness game.

We argue now that the challenge signatures given to $\mathcal{A}$ are distributed identically to those $\mathcal{A}$ would get in the blindness game. We have two cases:

- Case $c = ab$ (i.e. (A, B, C) is a valid DDH tuple): In this case, we have

  $$\sigma_1 = (R^*, S^*) = (G^b, G^{\frac{bx}{y}} \cdot G^{\frac{ab}{y}}),$$

  which is a valid randomized signature on the message $(G^a, \tilde{H}^a)$. In particular, $\sigma_1 = (R^*, S^*)$ is a valid signature on the message $(G^a, \tilde{H}^a)$ for randomness $\alpha = \frac{b}{r}$. Since b is random so is α. This case corresponds to the case $b = 0$ in the blindness game.
- Case c is a random element in $\mathbb{Z}_p$: In this case, we have

  $$\sigma_1 = (R^*, S^*) = (G^b, G^{\frac{bx}{y}} \cdot G^{\frac{c}{y}}),$$

  which is a valid signature on the message $(G^{\frac{c}{b}}, \tilde{H}^{\frac{c}{b}})$. Since c is random so is $\frac{c}{b}$. This case corresponds to the case $b = 1$ in the blindness game.

This concludes the proof.

□