Efficient Fully Structure-Preserving Signatures for Large Messages

Jens Groth*

University College London

Abstract. We construct both randomizable and strongly existentially unforgeable structure-preserving signatures for messages consisting of many group elements. To sign a message consisting of $N = mn$ group elements we have a verification key size of m group elements and signatures contain $n+2$ elements. Verification of a signature requires evaluating $n+1$ pairing product equations.

We also investigate the case of fully structure-preserving signatures where it is required that the secret signing key consists of group elements only. We show a variant of our signature scheme allowing the signer to pick part of the verification key at the time of signing is still secure. This gives us both randomizable and strongly existentially unforgeable fully structure-preserving signatures. In the fully structure preserving scheme the verification key is a single group element, signatures contain $m+n+1$ group elements and verification requires evaluating $n+1$ pairing product equations.

Keywords: Digital signatures, pairing-based cryptography, full structure-preservation.

1 Introduction

Structure-preserving signatures are pairing-based signatures where verification keys, messages and signatures all consist solely of group elements and the verification algorithm relies on generic group operations such as multiplications and pairings to verify a signature. Structure-preserving signatures are interesting because they compose well with other structure-preserving primitives such as ElGamal encryption [ElG85] and Groth-Sahai proofs [GS12] for instance. By combining different structure-preserving components it is possible to build advanced cryptographic schemes in a modular manner. Applications of structure-preserving signatures include blind signatures [AFG+10,FV10], group signatures [AFG+10,FV10,LPY12], homomorphic signatures [LPJY13,ALP13], delegatable anonymous credentials [Fuc11], compact verifiable shuffles [CKLM12],

* This research was supported by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 307937 and the Engineering and Physical Sciences Research Council grant EP/J009520/1.

network encoding [ALP12], oblivious transfer [GH08,CDEN12], tightly secure encryption [HJ12,ADK+13] and anonymous e-cash [ZLG12].

Since structure-preserving signatures are basic components when building cryptographic schemes it is crucial to make them as efficient as possible. All cryptographic protocols built on top of a structure-preserving signature scheme will be affected by its efficiency. There has therefore been a significant amount of research into finding barriers for how efficient structure-preserving signatures can be and constructing schemes achieving these bounds. Abe et al. [AGHO11] demonstrated a lower bound of 3 group elements for structure-preserving signatures (using Type III pairings, which is the most efficient type) and found matching constructions with 3 element signatures.

While the case of signing a single group element has been well studied, the question of signing larger messages has received less attention. Most structure-preserving schemes offering to sign many elements do so by increasing the size of the verification key linearly in the message to be signed. One could of course imagine chopping a large message into smaller pieces and signing each of them individually and then sign the resulting signatures to bind them together. However, this approach incurs a multiplicative overhead proportional to the size of the signatures we use, which due to the lower bound will be at least a factor 3. Also, such constructions would require the use of many pairing product equations in the verification of a signature.

Recently Abe et al. [AKOT15] introduced the notion of *fully* structure-preserving signatures. In a fully structure-preserving signature scheme also the secret key is required to consist of group elements only, which stands in contrast to most current structure-preserving signature schemes where the secret key consists of field elements. Fully structure-preservation is useful in several contexts, it is for instance often the case in a PKI that to get a public key certified one must demonstrate possession of a matching secret key. When the secret key consists of group elements it becomes possible to use Groth-Sahai proofs to give efficient proofs of knowledge of the secret key.

Abe et al. [AKOT15] also considered the question of signing messages that consist of many group elements. Surprisingly they showed that one can give fully structure-preserving signatures that only grow propotionately to the square root of the message size. The reason this is remarkable is that in structure-preserving signatures one cannot use collision-resistant hash-functions to reduce the message size since they are structure-destroying and furthermore it is known that size-reducing strictly structure-preserving commitments do not exist [AHO12]. They also showed a lower bound that says the combined length of the verification key and the signature size must be at least the square root of the message size, which holds regardless of whether the structure-preservation is full or not.

1.1 Our contribution

As we said earlier it is crucial to optimize efficiency of structure-preserving signatures. In this paper we investigate the case of signing large messages and present very efficient structure-preserving signature schemes for signing many

elements at once. Our signature schemes will be designed directly with large messages in mind and therefore be more efficient than constructions relying on the combination of multiple signature schemes.

We construct a structure-preserving signature scheme for messages consisting of $N = mn$ group elements. The verification key contains m elements and the signature size is $n + 2$ elements. This matches the best structure-preserving signature schemes for a single group element, in which case we would have a single group element verification key and a 3 element signature but unlike prior constructions our signature scheme scales very well for large messages. The verification process involves $n + 1$ pairing product equations, so also this matches state of the art for signing a single group element but scales well to handle larger messages.

Depending on the context, it may be desirable to use a strong signature scheme where it is not only infeasible to forge signatures on messages that have not been seen before but it is also infeasible to create a new different signatures on messages that have already been signed. In other circumstances, however, quite the opposite may be the case and it may be desirable to have signatures that can be randomized. In particular, when combining structure-preserving signatures with Groth-Sahai proofs, randomizability may be desirable since some of the signature elements can be revealed in the clear after being randomized.

Our signature scheme is very flexible in the sense that the same verification key can be used for both strong signatures and randomizable signatures at the same time. We define the notion of a combined signature scheme where the signer can choose for each message whether to make the signature strongly unforgeable or randomizable.

We also present a modified construction that is *fully* structure-preserving. In order to get full structure-preservation it is necessary for the signer to know discrete logarithms of group elements that are paired with the message since she does not know the discrete logarithms of the group elements in the message. Surprisingly this can be achieved in a simple way in our signature scheme by letting the signer pick most of the verification key herself. Due to this property we now get a fully structure-preserving signature scheme where the verification key is just a single group element and the signature consists of $m + n + 2$ group elements.

1.2 Related work

The name "structure-preserving signature" was coined by Abe et al. [AFG+10] but there are earlier works giving structure-preserving signatures with the first being [Gro06].

Abe et al. [AGHO11] gave the first 3 element signature scheme for fully asymmetric pairings (Type III) and also proved that this is optimal. Abe et al. [AGOT14] give 2 element signatures based on partially asymmetric pairings (Type II) but Chatterjee and Menezes [CM15] showed that structure preserving signatures in the partially asymmetric setting are less efficient than signatures based on fully asymmetric pairings. In this paper we therefore only consider the

fully asymmetric setting, which gives the best efficiency and thus is the most relevant case to consider.

A line of research [HJ12,ACD+12,ADK+13,LPY15,BCPW15] has worked on basing structure-preserving signatures on standard assumptions such as the decision Diffie-Hellman or the decision linear assumptions. The fully structure-preserving signatures by Abe et al. [AKOT15] is based on the natural double pairing assumption, which is implied by the DDH assumption. However, Abe et al. [AGO11] has showed that 3 element signatures cannot be proven secure under a non-interactive assumption using black-box reductions, so strong assumptions are needed to get optimal efficiency. We will therefore base the security of our signatures on the generic group model [Nec94,Sho97] instead of aiming for security under a well-established assumption.

The signature scheme in Abe et al. [AGOT14] can be seen to be fully structure-preserving. It is a 3 group element signature scheme and is selectively randomizable. Selective randomizability means that signatures are strong but the signer can choose to release a randomization token to make a signature randomizable. This notion is different from our notion of a combined signature scheme where the signer can choose to create randomizable or strong signatures. The advantage of selective randomizable signatures is that all signatures are verified with the same verification equation; the disadvantage is the need to issue randomization tokens when making a signature randomizable.

As discussed earlier the most directly related work is by Abe et al. [AKOT15] who introduced the notion of fully structure-preserving signatures and constructed a square root complexity scheme based on the double pairing assumption. We give a detailed performance comparison in Table 1. If we use $m \approx n \approx \sqrt{N}$ their verification key contains $11 + 6\sqrt{N}$ group elements, signatures contain $11 + 4\sqrt{N}$ group elements, and they require $5 + \sqrt{N}$ pairing product equations to verify a signature. In comparison, our fully structure-preserving signature scheme has a verification key with 1 group element, signatures consist of $2 + 2\sqrt{N}$ group elements, and we use $1 + \sqrt{N}$ pairing product equations to verify signatures.

Scheme	Parameters	Verification key	Signature	PPE
[AKOT15]	$4\ \mathbb{G}_1, 4\ \mathbb{G}_2$	$1\ \mathbb{G}_1, 10 + 3m + 3n\ \mathbb{G}_2$	$7 + m + n\ \mathbb{G}_1, 4 + 2n\ \mathbb{G}_2$	$5 + n$
Our SPS	$1\ \mathbb{G}_1, n + 1\ \mathbb{G}_2$	$m\ \mathbb{G}_1$	$1\ \mathbb{G}_1, 1 + n\ \mathbb{G}_2$	$1 + n$
Our fully SPS	$1\ \mathbb{G}_1, n + m\ \mathbb{G}_2$	$1\ \mathbb{G}_1$	$m\ \mathbb{G}_1, 1 + n\ \mathbb{G}_2$	$1 + n$

Table 1. Comparison of structure-preserving signature schemes for messages consisting of $N = mn$ elements in $\mathbb{G}_2$. We display public parameter, verification key and signature sizes measured in group elements in $\mathbb{G}_1$ and $\mathbb{G}_2$ and number of pairing product equations required for verifying a signature. The public parameters also contain a description of the bilinear group. The public parameters can be reused for other cryptographic schemes so their cost can be amortized.

2 Preliminaries

2.1 Bilinear groups

Throughout the paper we let $\mathcal{G}$ be an asymmetric bilinear group generator that on security parameter λ returns $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H) \leftarrow \mathcal{G}(1^\lambda)$ with the following properties:

- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are groups of prime order p
- $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear map
- G generates $\mathbb{G}_1$, H generates $\mathbb{G}_2$ and $e(G, H)$ generates $\mathbb{G}_T$
- There are efficient algorithms for computing group operations, evaluating the bilinear map, comparing group elements and deciding membership of the groups

In a bilinear group we refer to deciding group membership, computing group operations in $\mathbb{G}_1, \mathbb{G}_2$ or $\mathbb{G}_T$, comparing group elements and evaluating the bilinear map as the generic group operations. In the signature schemes we construct we only use generic group operations.

Galbraith, Paterson and Smart [GPS08] distinguish between 3 types of bilinear group generators. In the Type I setting (also called the symmetric setting) $\mathbb{G}_1 = \mathbb{G}_2$, in the Type II setting there is an efficiently computable isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$, and in the Type III setting no isomorphism that is efficiently computable in either direction between the source groups exists. Throughout the paper we will work in the Type III setting, which gives the most efficient operations and therefore is most important setting.

It will be useful to use the notation of Escala et al. [EHK+13] that keeps track of the discrete logarithm of group elements. They represent a group element X in $\mathbb{G}_1$ by $[x]_1$ when $X = G^x$ and a group element Y in $\mathbb{G}_2$ as $[y]_2$ when $Y = H^y$ and a group element $Z \in \mathbb{G}_T$ as $[z]_T$ when $Z = e(G, H)^z$. In this notation the source group generators G and H are $[1]_1$ and $[1]_2$.

The advantage of using this notation is that it highights the underlying linear algebra performed on the exponents when we do group operations. Multiplying two group elements $X, Y \in \mathbb{G}_1$ to get XY for instance corresponds to $[x]_1 + [y]_1 = [x + y]_1$. Exponentiation of $X \in \mathbb{G}_1$ with $y \in \mathbb{Z}_p$ to get X^y can be written $y[x]_1 = [yx]_1$. Using the bilinear map on $X \in \mathbb{G}_1$ and $Y \in \mathbb{G}_2$ to get $e(X, Y)$ can be written as $[x]_1[y]_2 = [xy]_T$.

We can represent vectors of group elements $\boldsymbol{X} = (X_1, \ldots, X_n)$ in $\mathbb{G}_1$ as $[\boldsymbol{x}]_1$. The operations taking place in the groups have natural linear algebra equivalents, e.g., exponentiation of a vector of group elements to a matrix of exponents to get a new vector of group elements can be written $[\boldsymbol{x}]_1 A = [\boldsymbol{x}A]_1$. A pairing product $\prod_{i=1}^n e(X_i, Y_i)$ can be written $[\boldsymbol{x}]_1 \cdot [\boldsymbol{y}]_2 = [\boldsymbol{x} \cdot \boldsymbol{y}]_T$. Exponentiation of a number of group elements to the same exponent to get $(X_1^a, \ldots, X_n^a)$ can be written $[\boldsymbol{x}]_1 a = [\boldsymbol{x}a]_1$.

2.2 Signature schemes

Our signature schemes work over an asymmetric bilinear group generated by $\mathcal{G}$. This group may be generated by the signer and included in the public verification key. In many cryptographic schemes it is convenient for the signer to work on top of a pre-existing bilinear group though. We will therefore in the description of our signatures explicitly distinguish between a setup algorithm **Setup** that produces public parameters pp and a key generation algorithm the signer uses to generate her own keys. The setup algorithm we use in our paper generates a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, [1]_1, [1]_2) \leftarrow \mathcal{G}(1^\lambda)$. It then extends the description of the bilinear group with additional randomly selected group elements. Our signature scheme does not rely on knowledge of the discrete logarithms of these random group elements, so the setup may be reused for many different signature schemes and other cryptographic schemes.

A signature scheme (with setup algorithm **Setup**) consists of efficient algorithms $(\mathbf{Setup}, \mathbf{Gen}, \mathbf{Sign}, \mathbf{Vfy})$.

Setup$(1^\lambda) \rightarrow pp$**:** The setup algorithm generates public parameters pp. They specify a message space $\mathcal{M}_{pp}$.

Gen$(pp) \rightarrow (vk, sk)$**:** The key generation algorithm takes public parameters pp as input and returns a public verification key vk and a secret signing key sk.

Sign$(pp, sk, m) \rightarrow \sigma$**:** The signing algorithm takes a signing key sk and a message $m \in \mathcal{M}_{pp}$ as input and returns a signature σ.

Vfy$(pp, vk, m, \sigma) \rightarrow 1/0$**:** The verification algorithm takes the verification key vk, a message m and a purported signature σ as input and returns either 1 (accept) or 0 (reject).

Definition 1 (Correctness). *The signature scheme* $(\mathbf{Setup}, \mathbf{Gen}, \mathbf{Sign}, \mathbf{Vfy})$ *is (perfectly) correct if for all security parameters* $k \in \mathbb{N}$

$$\Pr\left[\begin{matrix} pp \leftarrow \mathbf{Setup}(1^\lambda); (vk, sk) \leftarrow \mathbf{Gen}(pp) \\ m \leftarrow \mathcal{M}_{pp}; \sigma \leftarrow \mathbf{Sign}(pp, sk, m) \end{matrix} : \mathbf{Vfy}(pp, vk, m, \sigma) = 1\right] = 1.$$

2.3 Structure-preserving signature schemes

In this paper, we study structure-preserving signature schemes [AFG+10]. In a structure-preserving signature scheme the verification key, the messages and the signatures consist only of group elements from $\mathbb{G}_1$ and $\mathbb{G}_2$ and the verification algorithm evaluates the signature by deciding group membership of elements in the signature and by evaluating pairing product equations, which are equations of the form

$$\prod_i \prod_j e(X_i, X_j)^{a_{ij}} = 1,$$

where $X_1, X_2, \ldots \in \mathbb{G}_1$ are group elements appearing in pp, vk, m and σ and $a_{11}, a_{12}, \ldots \in \mathbb{Z}$ are constants.

Structure-preserving signatures are extremely versatile because they mix well with other pairing-based protocols. Groth-Sahai proofs [GS12] are for instance

designed with pairing product equations in mind and can therefore easily be applied to structure-preserving signatures.

Definition 2 (Structure-preserving signatures). *A signature scheme is said to be structure preserving over bilinear group generator $\mathcal{G}$ if*

- *public parameters include a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, [1]_1, [1]_2) \leftarrow \mathcal{G}(1^\lambda)$,*
- *verification keys consist of group elements in $\mathbb{G}_1$ and $\mathbb{G}_2$,*
- *messages consist of group elements in $\mathbb{G}_1$ and $\mathbb{G}_2$,*
- *signatures consist of group elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, and*
- *the verification algorithm only needs to decide membership in $\mathbb{G}_1$ and $\mathbb{G}_2$ and evaluate pairing product equations.*

Fully structure preserving signatures. Abe et al. [AKOT15] argue that in several applications it is desirable that also the secret signing keys only contain source group elements. They define a structure-preserving signature scheme to be *fully* structure preserving if the signing key sk consists of group elements in $\mathbb{G}_1$ and $\mathbb{G}_2$ and the correctness of the secret signing key with respect to the public verification key can be verified using pairing product equations.

3 Randomizable and strongly unforgeable signatures

A signature scheme is said to be existentially unforgeable if it is infeasible to forge a signature on a message that has not previously been signed. The standard definition of existential unforgeability allows the adversary to modify an existing signature on a message to a new signature on the same message. We say a signature scheme is randomizable if it is possible to randomize a signature on a message to get a new random signature on the same message. On the other hand, we say a signature scheme is *strongly* unforgeable when it is also infeasible to modify a signature, or more precisely it is infeasible to construct a valid message and signature pair that has not previously been seen.

Both strong signatures and randomizable signatures have many uses. We will therefore construct both strongly existentially unforgeable signatures and randomizable signatures. To capture the best of both worlds, we will define a combined signature scheme where the signer can decide whether a signature should be randomizable or strongly unforgeable. Randomizable signatures are constructed using signing algorithm $\mathbf{Sign}_0$ and verified by verification algorithm $\mathbf{Vfy}_0$. Strongly unforgeable signatures are constructed using signing algorithm $\mathbf{Sign}_1$ and verified by verification algorithm $\mathbf{Vfy}_1$.

A naïve combined signature scheme would have a verification key containing two verification keys, one for randomizable signatures and one for strong signatures. However, this solution has the disadvantage of increasing key size. Instead we will in this paper construct a combined signature scheme where the verification key is just a single group element that can be used to verify either type of signature. This dual use of the verification key means that we must carefully

consider the security implications of combining two signature schemes though, so we will now define a combined signature scheme.

A combined signature scheme $(\mathbf{Setup}, \mathbf{Gen}, \mathbf{Sign}_0, \mathbf{Vfy}_0, \mathbf{Rand}, \mathbf{Sign}_1, \mathbf{Vfy}_1)$ consists of 7 probabilistic polynomial time algorithms as described below.

$\mathbf{Setup}(1^\lambda, \text{size}) \to pp$**:** The setup algorithm takes the security parameter λ and description of the size of messages to be signed and generates public parameters. It defines a message space $\mathcal{M}_{pp}$ of messages that can be signed.

$\mathbf{Gen}(pp) \to (vk, sk)$**:** The key generation algorithm given public parameters generates a public verification key vk and a secret signing key sk.

$\mathbf{Sign}_0(pp, sk, m) \to \sigma$**:** The randomizable signature algorithm given the signing key and a message m returns a randomizable signature σ.

$\mathbf{Vfy}_0(pp, vk, m, \sigma) \to 1/0$**:** The randomizable signature verification algorithm given a message and a purported randomizable signature on it returns 1 if accepting the signature and 0 if rejecting the signature.

$\mathbf{Rand}(pp, vk, m, \sigma) \to \sigma'$**:** The randomization algorithm given a valid randomizable signature on a message returns a new randomized signature on the same message.

$\mathbf{Sign}_1(pp, sk, m) \to \sigma$**:** The strong signature algorithm given the signing key and a message m returns a strongly unforgeable signature σ.

$\mathbf{Vfy}_1(pp, vk, m, \sigma) \to 1/0$**:** The strong signature verification algorithm given a message and a purported strong signature on it returns 1 if accepting the signature and 0 if rejecting the signature.

We say a combined signature scheme has perfect correctness if the constituent randomizable and strongly unforgeable signature schemes $(\mathbf{Setup}, \mathbf{Gen}, \mathbf{Sign}_0, \mathbf{Vfy}_0)$ and $(\mathbf{Setup}, \mathbf{Gen}, \mathbf{Sign}_1, \mathbf{Vfy}_1)$ both are perfectly correct.

The combined signatures are perfectly randomizable if a randomized signature looks exactly like a fresh signature on the same message.

Definition 3 (Perfect randomizability). *The combined signature scheme is perfectly randomizable if for all $\lambda \in \mathbb{N}$ and all stateful adversaries $\mathcal{A}$*

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathbf{Setup}(1^\lambda); (vk, sk) \leftarrow \mathbf{Gen}(pp) \\ m \leftarrow \mathcal{A}(pp, vk, sk); \sigma, \sigma_0 \leftarrow \mathbf{Sign}_0(pp, sk, m) \; : \mathcal{A}(\sigma, \sigma_b) = b \\ \sigma_1 \leftarrow \mathbf{Rand}(pp, vk, m, \sigma); b \leftarrow \{0, 1\} \end{array}\right] = \frac{1}{2},$$

where $\mathcal{A}$ outputs $m \in \mathcal{M}_{pp}$.

To capture the attacks that can occur against a combined signature scheme, we assume the adversary may arbitrarily query a signer for randomizable or strong signatures. We want the signature scheme to be combined existentially unforgeable in the sense that even seeing randomizable signatures does not help in breaking strong existential unforgeability and on the other hand seeing strong signatures does not help in producing randomizable signatures.

Definition 4 (Combined existential unforgeability under chosen message attack). *The combined signature scheme is combined existentially unforgeable under adaptive chosen message attack (C-EUF-CMA) if for all probabilistic*

polynomial time adversaries $\mathcal{A}$

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathbf{Setup}(1^\lambda); (vk, sk) \leftarrow \mathbf{Gen}(pp) \\ (m, \sigma) \leftarrow \mathcal{A}^{\mathbf{Sign}_0(pp,sk,\cdot),\mathbf{Sign}_1(pp,sk,\cdot)}(pp, vk) \end{array} : \begin{array}{l} \mathbf{Vfy}_0(pp, vk, m, \sigma) = 1 \wedge m \notin Q_0 \quad \text{or} \\ \mathbf{Vfy}_1(pp, vk, m, \sigma) = 1 \wedge (m, \sigma) \notin Q_1 \end{array}\right]$$

is negligible, where $\mathcal{A}$ *outputs* $m \in \mathcal{M}_{pp}$ *and always queries on messages in* $\mathcal{M}_{pp}$ *and* Q_0 *is the set of messages that have been queried to* $\mathbf{Sign}_0$ *to get randomizable signatures and* Q_1 *is the set of message and signature pairs from queries to* $\mathbf{Sign}_1$ *to get strongly unforgeable signatures.*

4 Structure-preserving combined signature scheme

Fig. 1 describes a structure-preserving combined signature scheme that can be used to sign messages consisting of $N = mn$ group elements in $\mathbb{G}_2$. It has a verification key size of m group elements, a signature size of $n+2$ group elements, and verification involves evaluating $n + 1$ pairing product equations.

$\mathbf{Setup}(1^\lambda, m, n)$
$gk = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, [1]_1, [1]_2) \leftarrow \mathcal{G}(1^\lambda)$
$[\boldsymbol{y}]_2 \leftarrow \mathbb{G}_2^n$
Return $pp = (gk, m, n, [\boldsymbol{y}]_2)$

$\mathbf{Gen}(pp)$
$\boldsymbol{u} \leftarrow \mathbb{Z}_p^{m-1}$, $v \leftarrow \mathbb{Z}_p$
$vk = ([\boldsymbol{u}]_1, [v]_1)$
$sk = (\boldsymbol{u}, v)$
Return (vk, sk)

$\mathbf{Sign}_b(pp, sk, [M]_2)$
$z \leftarrow \mathbb{Z}_p^*$
$r = \frac{1}{z}$
$[s]_2 = z([y_1]_2 + [v]_2)$
$[\boldsymbol{t}]_2 = z\left((\boldsymbol{u}, 1)[M]_2 + v[\boldsymbol{y}]_2 + bv[s]_2\mathbf{1}\right)$
Return $\sigma = ([r]_1, [s]_2, [\boldsymbol{t}]_2)$

$\mathbf{Vfy}_b(pp, vk, [M]_2, \sigma)$
Parse $\sigma = ([r]_1, [s]_2, [\boldsymbol{t}]_2)$
Return 1 if and only if
$[M]_2 \in \mathbb{G}_2^{m \times n}$
$[r]_1 \in \mathbb{G}_1$
$[s]_2 \in \mathbb{G}_2$
$[\boldsymbol{t}]_2 \in \mathbb{G}_2^n$
$[r]_1[s]_2 = [1]_1[y_1]_2 + [v]_1[1]_2$
$[r]_1[\boldsymbol{t}]_2 = [(\boldsymbol{u}, 1)]_1[M]_2 + [v]_1[\boldsymbol{y}]_2 + b[v]_1[s]_2\mathbf{1}$

$\mathbf{Rand}(pp, vk, M, \sigma)$
Parse $\sigma = ([r]_1, [s]_2, [\boldsymbol{t}]_2)$
$\beta \leftarrow \mathbb{Z}_p^*$
$[r']_1 = \frac{1}{\beta}[r]_1$
$[s']_2 = \beta[s]_2$
$[\boldsymbol{t}']_2 = \beta[\boldsymbol{t}]_2$
Return $\sigma' = ([r']_1, [s']_2, [\boldsymbol{t}']_2)$

Fig. 1. Structure-preserving combined signature scheme. The signature and verification algorithms for randomizable and strongly unforgeable signatures, respectively, are quite similar. We have there described them at the same time indicating the choice by $b = 0$ for randomizable signatures and $b = 1$ for strongly unforgeable signatures.

In order to explain some of the design principles underlying the construction, let us first consider the special case where the message space is $\mathbb{G}_2$, i.e., we are signing a single group element and $N = m = n = 1$. The setup includes a random group element $[y]_2$, the verification key consists of a single group element

$[v]_1$, and both randomizable and strongly unforgeable signatures are of the form $\sigma = ([r]_1, [s]_2, [t]_2)$.

For a randomizable signature there are two verification equations

$$[r]_1[s]_2 = [1]_1[y]_2 + [v]_1[1]_2 \qquad [r]_1[t]_2 = [1]_1[m]_2 + [v]_1[y]_2.$$

It is easy to see that we can randomize the factors in $[r]_1[s]_2$ and $[r]_1[t]_2$ into $(\frac{1}{\beta}[r]_1)(\beta[s]_2)$ and $(\frac{1}{\beta}[r]_1)(\beta[t]_2)$ without changing the products themselves, which gives us randomizability of the signatures.

The first verification equation is designed to prevent the adversary from creating a forged signature from scratch after seeing the verification key only. An adversary *using only generic group operations* can do no better than computing $[r]_1 = \rho[1]_1 + \rho_v[v]_1$ and $[s]_2 = \sigma[1]_2 + \sigma_y[y]_2$ using known scalars $\rho, \rho_v, \sigma, \sigma_y \in \mathbb{Z}_p$. Looking at the underlying discrete logarithms, the first verification equation then corresponds to the polynomial equation

$$(\rho + \rho_v v)(\sigma + \sigma_y y) = y + v$$

in the unknown discrete logarithms v and y. This equation is not solvable: Looking at the $\rho_v \sigma v = v$ terms we see $\sigma \neq 0$. Looking at the $\rho\sigma_y y = y$ terms we see $\rho \neq 0$. But this would leave us with a constant term $\rho\sigma \neq 0$.

Now, what if the adversary instead of creating a signature from scratch tries to modify an existing signature or combine many existing signatures? Well, due to the randomness in the choice of $z \leftarrow \mathbb{Z}_p^*$ in the signing protocol each signature query will yield a signature with a different random $[r_i]_1$. As it turns out this randomization used in each signature makes it hard for the adversary to combine multiple signatures, or even modify one signature, in a meaningful way with generic group operations. The intuition is that generic group operations allow the adversary to take linear combinations of elements it has seen, however, the verificaction equations are quadratic.

In order to prevent randomization and get strong existential unforgeability the combined signature scheme modifies the latter verification equation by adding a $[v]_1[s]_2$ term. This gives us the following verification equations for strongly unforgeable signatures

$$[r]_1[s]_2 = [1]_1[y]_2 + [v]_1[1]_2 \qquad [r]_1[t]_2 = [1]_1[m]_2 + [v]_1[y]_2 + [v]_1[s]_2.$$

Now the randomization technique fails because a randomization of $[s]_2$ means we must change $[t]_2$ in a way that counteracts this change in the second verification equation. However, $[t]_2$ is paired with $[r]_1$ that also changes when $[s]_2$ changes. The adversary is therefore faced with a non-linear modification of the signatures and gets stuck because generic group operations only enable it to do linear modifications of signature elements.

We can extend the one-element signature scheme to sign a vector $[\boldsymbol{m}]_2$ with m group elements in $\mathbb{G}_2$ by extending the verification key by $m-1$ random group elements $[\boldsymbol{u}]_1 = [(u_1, \ldots, u_{m-1})]_1$. Now the verification equations become

$$[r]_1[s]_2 = [1]_1[y_1]_2 + [v]_1[1]_2 \qquad [r]_1[t]_2 = [(\boldsymbol{u}, 1)]_1 \cdot [\boldsymbol{m}]_2 + [v]_1[y]_2 + b[v]_1[s]_2,$$

where $b = 0$ for a randomizable signature and $b = 1$ for a strong signature. The idea is that the discrete logarithms of the elements in $[\boldsymbol{u}]_1$ are unknown to the adversary making it hard to change either group element in a previously signed message to get a new message that will verify under the same signature.

Finally, to sign mn group elements in $\mathbb{G}_2$ instead of m group elements we keep the first verification equation, which does not involve the message, but add $n-1$ extra verification equations similar to the second verification equation for a vector of group elements described above. This allows us to sign n vectors in parallel. In order to avoid linear combinations of message vectors and signature components being useful in other verification equations, we give each verification equation a separate $[v]_1[y_k]_2$ term, where $k = 1, \ldots, n$ is the number of the verification equation.

Theorem 1. *Fig. 1 gives a structure-preserving combined signature scheme that is C-EUF-CMA secure in the generic group model.*

Proof. Perfect correctness, perfect randomizability and structure-preservation follows by inspection. What remains now is to prove that the signature scheme is C-EUF-CMA secure in the generic group model. In the (Type III) generic bilinear group model the adversary may compute new group elements in either source group by taking arbitrary linear combinations of previously seen group elements in the same source group. We shall see that no such linear combination of group elements, viewed as formal Laurent polynomials in the variables picked by the key generator and the signing oracle, yields an existential forgery. It follows along the lines of the Uber assumption of Boneh, Boyen and Goh [BBG05] from the inability to produce forgeries when working with formal Laurent polynomials that the signature scheme is C-EUF-CMA secure in the generic bilinear group model.

Suppose the adversary makes q queries $[M_i]_2 \in \mathbb{G}_2^{m \times n}$ to get signatures

$$[r_i]_1 = [\frac{1}{z_i}]_1 \qquad [s_i]_2 = [z_i(y_1+v)]_2 \qquad [\boldsymbol{t}_i]_2 = [z_i\left((\boldsymbol{u},1)M_i + v\boldsymbol{y} + b_i z_i v(y_1+v)\right)]_2,$$

where $b_i = 0$ if query i is for a randomizable signature and $b_i = 1$ if query i is for a strong signature, and where M_i may depend on previously seen signature elements in $[s_j]_2, [\boldsymbol{t}_j]_2$ for $j < i$.

Viewed as Laurent polynomials we have that a signature $([r]_1, [s]_2, [\boldsymbol{t}]_2)$ generated by the adversary on $[M] \in \mathbb{G}_2^{m \times n}$ is of the form

$$r = \rho + v\rho_v + \boldsymbol{u}\boldsymbol{\rho}_u^\top + \sum_i \frac{1}{z_i}\rho_{r_i}$$

$$s = \sigma + \boldsymbol{\sigma}_y \boldsymbol{y}^\top + \sum_j \sigma_{s_j} z_j(y_1+v) + \sum_j \boldsymbol{\sigma}_{t_j} z_j\left((\boldsymbol{u},1)M_j + v\boldsymbol{y} + b_j z_j v(y_1+v)\mathbf{1}\right)$$

$$\boldsymbol{t} = \boldsymbol{\tau} + \boldsymbol{y}T_y + \sum_j z_j(y_1+v)\boldsymbol{\tau}_{s_j} + \sum_j z_j\left((\boldsymbol{u},1)M_j + v\boldsymbol{y} + b_j z_j v(y_1+v)\mathbf{1}\right)T_{t_j}$$

Similarly, all mn entries in M can be written on a form similar to s and all entries in queried matrices M_i can be written on a form similar to s where the sums are bounded by $j < i$.

For the first verification equation to be satisfied we must have $rs = y_1 + v$, i.e.,

$$\begin{pmatrix} \rho + \boldsymbol{u}\boldsymbol{\rho}_u^\top \\ +v\rho_v + \sum_i \frac{1}{z_i}\rho_{r_i} \end{pmatrix} \begin{pmatrix} \sigma + \boldsymbol{\sigma}_y \boldsymbol{y}^\top + \sum_j \sigma_{s_j} z_j (y_1 + v) \\ + \sum_j \boldsymbol{\sigma}_{t_j} z_j \big((\boldsymbol{u}, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + v)\boldsymbol{1}\big)^\top \end{pmatrix} = y_1 + v$$

We start by noting that $r \neq 0$ since otherwise rs cannot have the term y_1. Please observe that it is only in $\mathbb{G}_1$ that we have terms including indeterminates with negative power, i.e., $\frac{1}{z_i}$. In $\mathbb{G}_2$ all indeterminates have positive power, i.e., so $s_j, \boldsymbol{t}_j, M_j$ only contain proper multi-variate polynomials. Now suppose for a moment that $\rho_{r_i} = 0$ for all i. Then in order not to have a terms involving z_j's in rs we must have $\sum_j \sigma_{s_j} z_j (y_1 + v) + \sum_j \boldsymbol{\sigma}_{t_j} z_j \left((\boldsymbol{u}, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + v)\boldsymbol{1}\right)^\top = 0$. The term y_1 now gives us $\rho\sigma_{y,1} = 1$ and the term v gives us $\rho_v \sigma = 1$. This means $\rho \neq 0$ and $\sigma \neq 0$ and therefore we reach a contradiction since the constant term should be $\rho\sigma = 0$. We conclude that there must exist some ℓ for which $\rho_{r_\ell} \neq 0$.

Now we have the term $\rho_{r_\ell} \sigma \frac{1}{z_\ell} = 0$, which shows us $\sigma = 0$. The terms $\rho_{r_\ell} \sigma_{y,k} \frac{y_k}{z_\ell} = 0$ for $k = 1, \ldots, n$ give us $\boldsymbol{\sigma}_y = \boldsymbol{0}$.

The polynomials corresponding to s_j and $\boldsymbol{t}_j$ contain the indeterminate z_j in all terms, so no linear combination of them can give us a term where the indeterminate component is vy_k for some $k \in \{1, \ldots, n\}$. Since M_j is constructed as a linear combination of elements in the verification key and components in $\mathbb{G}_2$ from previously seen signatures, it too cannot contain a term where the indeterminate component is vy_k. The coefficient of $\frac{z_j}{z_\ell} v y_k$ is therefore $\rho_{r_\ell} \sigma_{t_j,k} = 0$ and therefore $\sigma_{t_j,k} = 0$ for every $j \neq \ell$ and $k \in \{1, \ldots, n\}$. This shows $\boldsymbol{\sigma}_{t_j} = \boldsymbol{0}$ for all $j \neq \ell$. Looking at the coefficients for vy_k for $k = 1, \ldots, n$ we see that $\boldsymbol{\sigma}_{t_\ell} = \boldsymbol{0}$ too.

The terms $\rho_{r_\ell} \sigma_{s_j} \frac{z_j}{z_l} v$ give us $\sigma_{s_j} = 0$ for all $j \neq \ell$. In order to get a coefficient of 1 for the term y_1 we see that $\sigma_{s_\ell} = \frac{1}{\rho_{r_\ell}}$, which is non-zero. Our analysis has now shown that

$$s = \frac{1}{\rho_{r_\ell}} z_\ell (y_1 + v).$$

Let us now analyze the structure of r. The term $\rho_v \sigma_\ell v^2 z_\ell = 0$ gives us $\rho_v = 0$. We know from our previous analysis that if there was a second $i \neq \ell$ for which $\rho_{r_i} \neq 0$ then also $\sigma_{\rho_\ell} = 0$, which it is not. Therefore for all $i \neq \ell$ we have $\rho_{r_i} = 0$. The term $\rho\sigma_{s_\ell} z_\ell y_1$ gives $\rho = 0$. The terms in $\sigma_{s_\ell} \boldsymbol{u} z_\ell v \boldsymbol{\rho}_u^\top$ give us $\boldsymbol{\rho}_u = \boldsymbol{0}$. Our analysis therefore shows

$$r = \rho_{r_\ell} \frac{1}{z_\ell}.$$

We now turn to the second verification equation, which is $rt_1 = (\boldsymbol{u}, 1)\boldsymbol{m}^\top + vy_1 + bvs$, where $\boldsymbol{m}^\top$ is the first column vector of M. The message vector is of

the form

$$\boldsymbol{m} = \begin{array}{c} \boldsymbol{\mu} + \boldsymbol{y}M_y + \sum_j \boldsymbol{\mu}_{s_j} z_j (y_1 + v) \\ + \sum_j z_j \left((\boldsymbol{u}, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + v) \mathbf{1} \right) M_{t_j} \end{array}$$

where $\boldsymbol{\mu}, M_y \boldsymbol{\mu}_{s_j}$ and M_{t_j} are suitably sized vectors and matrices with entries in $\mathbb{Z}_p$ chosen by the adversary. Similarly, we can write out $t_1 = \tau + \boldsymbol{\tau}_y \boldsymbol{y}^\top + \sum_j \tau_{s_j} z_j (y_1 + v) + \sum_j \boldsymbol{\tau}_{t_j} z_j \left((\boldsymbol{u}, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + v) \mathbf{1} \right)$ for elements and suitably sized vectors $\tau, \boldsymbol{\tau}_y, \tau_{s_j}, \boldsymbol{\tau}_{t_j}$ with entries in $\mathbb{Z}_p$ chosen by the adversary.

Writing out the second verification equation we have

$$\begin{aligned} & \rho_{r_\ell} \frac{1}{z_\ell} \begin{pmatrix} \tau + \boldsymbol{\tau}_y \boldsymbol{y}^\top + \sum_j \tau_{s_j} z_j (y_1 + v) \\ + \sum_j \boldsymbol{\tau}_{t_j} z_j \left((\boldsymbol{u}, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + v) \mathbf{1} \right) \end{pmatrix} \\ = & \, v y_1 + bv \left(\frac{1}{\rho_{r_\ell}} z_\ell (y_1 + v) \right) \\ & + (\boldsymbol{u}, 1) \begin{pmatrix} \boldsymbol{\mu} + \boldsymbol{y}M_y + \sum_j \boldsymbol{\mu}_{s_j} z_j (y_1 + v) \\ + \sum_j z_j \left((\boldsymbol{u}, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + v) \mathbf{1} \right) M_{t_j} \end{pmatrix}^\top . \end{aligned}$$

Looking at the coefficients of terms involving $\frac{1}{z_\ell}$ and $\frac{y_k}{z_\ell}$ we get $\tau = 0$ and $\boldsymbol{\tau}_y = \mathbf{0}$. Looking at the terms in $\rho_{r_\ell} \boldsymbol{\tau}_{t_j} \frac{z_j}{z_\ell} v\boldsymbol{y}$ we get $\boldsymbol{\tau}_{t_j} = \mathbf{0}$ for all $j \neq \ell$. Similarly, the terms $\rho_{r_\ell} \tau_{s_j} \frac{z_j}{z_\ell} v$ give us $\tau_{s_j} = 0$ for all $j \neq \ell$. We are now left with

$$\begin{aligned} & \rho_{r_\ell} \left(\tau_{s_\ell} (y_1 + v) + \boldsymbol{\tau}_{t_\ell} \left((\boldsymbol{u}, 1) M_\ell + v\boldsymbol{y} + b_\ell v z_\ell (y_1 + v) \mathbf{1} \right) \right) \\ = & \, v y_1 + bv \frac{1}{\rho_{r_\ell}} z_\ell (y_1 + v) \\ & + (\boldsymbol{u}, 1) \begin{pmatrix} \boldsymbol{\mu} + \boldsymbol{y}M_y + \sum_j \boldsymbol{\mu}_{s_j} z_j (y_1 + v) \\ + \sum_j z_j \left((\boldsymbol{u}, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + v) \mathbf{1} \right) M_{t_j} \end{pmatrix}^\top . \end{aligned}$$

Terms involving z_j and z_j^2 must cancel out, so we can assume $\boldsymbol{\mu}_{s_j} = \mathbf{0}$ and $M_{t_j} = 0$ for $j > \ell$. Since M_ℓ does not involve z_ℓ in any of its terms, we get from the terms in $(\boldsymbol{u}, 1) z_\ell v \boldsymbol{\mu}_{s_\ell}^\top$ that $\boldsymbol{\mu}_{s_\ell} = 0$. Since there can be no terms involving z_ℓ^2 we get $b_\ell \mathbf{1} M_{t_\ell}^\top = \mathbf{0}$. Looking at the coefficients for v we get $\tau_{s_\ell} = 0$. This leaves us with

$$\begin{aligned} & \rho_{r_\ell} \boldsymbol{\tau}_{t_\ell} \left((\boldsymbol{u}, 1) M_\ell + v\boldsymbol{y} + b_\ell v z_\ell (y_1 + v) \mathbf{1} \right)^\top \\ = & \, v y_1 + bv \frac{1}{\rho_{r_\ell}} z_\ell (y_1 + v) + (\boldsymbol{u}, 1) z_\ell \left((\boldsymbol{u}, 1) M_\ell + v\boldsymbol{y} \right) M_{t_\ell})^\top \\ & + (\boldsymbol{u}, 1) \begin{pmatrix} \boldsymbol{\mu} + \boldsymbol{y}M_y + \sum_{j<\ell} \boldsymbol{\mu}_{s_j} z_j (y_1 + v) \\ + \sum_{j<\ell} z_j \left((\boldsymbol{u}, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + v) \mathbf{1} \right) M_{t_j} \end{pmatrix}^\top . \end{aligned}$$

Looking at the terms involving $z_\ell v^2$ we see $\rho_{r_\ell} \boldsymbol{\tau}_{t_\ell} b_\ell \mathbf{1}^\top = b \frac{1}{\rho_{r_\ell}}$. This cancels out the first two parts involving z_ℓ. The only remaining terms involving z_ℓ now give us $M_{t_\ell} = 0$. This gives us

$$\begin{aligned} & \rho_{r_\ell} \boldsymbol{\tau}_{t_\ell} \left((\boldsymbol{u}, 1) M_\ell + v\boldsymbol{y} \right)^\top - \boldsymbol{y}_1 \\ = & \, (\boldsymbol{u}, 1) \begin{pmatrix} \boldsymbol{\mu} + \boldsymbol{y}M_y + \sum_{j<\ell} \boldsymbol{\mu}_{s_j}^{(\ell)} z_j (y_1 + v) \\ + \sum_{j<\ell} z_j \left((\boldsymbol{u}, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + v) \mathbf{1} \right) M_{t_j} \end{pmatrix}^\top \end{aligned}$$

Looking at the terms in $v\boldsymbol{y}$ we now get $\rho_{r_\ell}\boldsymbol{\tau}_{t_\ell} = (1, 0, \ldots, 0)$. Let the first column vector of M_ℓ be $\boldsymbol{m}_\ell^\top$ then we now have

$$(\boldsymbol{u}, 1)\boldsymbol{m}_\ell^\top = (\boldsymbol{u}, 1)\boldsymbol{m}^\top.$$

Writing

$$\boldsymbol{m}' = \begin{array}{c} \boldsymbol{m}_\ell - \boldsymbol{m} = \boldsymbol{\mu}' + \boldsymbol{y}M'_y + \sum_{j<\ell} \boldsymbol{\mu}'_{s_j} z_j (y_1 + v) \\ + \sum_{j<\ell} z_j \left((\boldsymbol{u}, 1)M_j + v\boldsymbol{y} + b_j v z_j (y_1 + v)\mathbf{1} \right) M'_{t_j} \end{array}$$

we now have

$$(\boldsymbol{u}, 1) \begin{pmatrix} \boldsymbol{\mu}' + \boldsymbol{y}M'_y + \sum_{j<\ell} \boldsymbol{\mu}'_{s_j} z_j (y_1 + v) \\ + \sum_{j<\ell} z_j \left((\boldsymbol{u}, 1)M_j + v\boldsymbol{y} + b_j v z_j (y_1 + v)\mathbf{1} \right) M'_{t_j} \end{pmatrix}^\top = 0.$$

The terms in $(\boldsymbol{u}, 1)\boldsymbol{\mu}'^\top$ tell us $\boldsymbol{\mu}' = \mathbf{0}$. Looking at terms involving $u_i y_k$ or y_k gives us $M'_y = 0$. Terms with z_j^2 tell us $b_j \mathbf{1} M'_{t_j} = \mathbf{0}$ for all j. Terms in $(\boldsymbol{u}, 1) z_j v \mu'_{s_j}$ tell us $\mu'_{s_j} = 0$ for all j. Finally, terms in $(\boldsymbol{u}, 1)(v\boldsymbol{y}M'_{t_j})$ give us $M'_{t_j} = 0$.

We have now deduced that $\boldsymbol{m}' = \mathbf{0}$ and therefore $\boldsymbol{m}_\ell = \boldsymbol{m}$. This means the first column in M for which the adversary has produced a signature is a copy of the first column in the queried message M_ℓ. Using the same analysis on the last $n - 1$ verification equations gives us that the other $n - 1$ columns also match. This means a generic adversary can only produce valid signatures for previously queried messages, so we have EUF-CMA security.

Finally, let us consider the case where $b = 1$, i.e., we are doing a strong signature verification. We saw earlier that $\rho_{r_\ell}\boldsymbol{\tau}_{t_\ell} b_\ell \mathbf{1}^\top = b_\ell = b\frac{1}{\rho_{r_\ell}}$ which can only be satisfied if $b_\ell = 1$ and $\rho_{r_\ell} = 1$. This means $s = s_\ell$ and $r = r_\ell$ and $M = M_\ell$ and therefore $\boldsymbol{t} = \boldsymbol{t}_\ell$. So the generic adversary can only satisfy the strong verification equation with $b = 1$ by copying both the message and signature from a previous query with $b_\ell = 1$.

On the other hand, if $b = 0$, i.e., we are verifying a randomizable signature, we see from $\rho_{r_\ell}\boldsymbol{\tau}_{t_\ell} b_l \mathbf{1}^\top = b_\ell = b\frac{1}{\rho_{r_\ell}}$ that $b_\ell = 0$. So the adversary has randomized a signature intended for randomization. □

5 Fully structure-preserving combined signature scheme

The earlier structure-preserving signature scheme uses knowledge of the discrete logarithms of $[\boldsymbol{u}]_1$ in a fundamental way since $[\boldsymbol{t}]_2$ contains a $z(\boldsymbol{u}, 1)[M]_2$ component that could not be computed without these discrete logarithms. This situation is common for all structure-preserving signature schemes for messages that are vectors of group elements. The need to specify such discrete logarithms in the signing key therefore prevents them from being fully structure-preserving.

Abe et al. [AKOT15] get around this problem by only pairing message group elements with signature group elements where the signer knows the discrete logarithms. Inspired by their work, we will let the signer pick $[\boldsymbol{u}]_1$ and include it in the signature.

To make this idea work we first make a minor modification to our signature scheme from before. We include a vector of $m-1$ group elements $[\boldsymbol{x}]_2$ in the setup and we modify $[s]_2$ to have the form $[s]_2 = z([y_1]_2 + \boldsymbol{u} \cdot [\boldsymbol{x}]_2 + [v]_2)$. The first verification equation then becomes

$$[r]_1[s]_2 = [1]_1[y_1]_2 + [\boldsymbol{u}]_1 \cdot [\boldsymbol{x}]_2 + [v]_1[1]_2.$$

If this was the only modification we made it is not hard to see that the same security proof we gave earlier will work again, we are only modifying the verification equation by a random constant $[\boldsymbol{u} \cdot \boldsymbol{x}]_T$. The surprising thing though is that the signature scheme remains secure if we let the signer pick the $[\boldsymbol{u}]_1$ part of the verification key herself and include it in the signature.

Letting the signer pick $[\boldsymbol{u}]_1$ as part of the verification key means that she can know their discrete logarithms. Since she also picks $z \leftarrow \mathbb{Z}_p^*$ herself she can now use linear operations to compute the $z(\boldsymbol{u}, 1)[M]_2$ part of $[\boldsymbol{t}]_2$. Furthermore, we have designed the scheme such that the rest can be computed with linear operations as well. To make randomizable signatures the signer just needs to know $[v]_2$ and $[v\boldsymbol{y}]_2$. To make strong signatures she additionally needs to know $[v\boldsymbol{x}]_2$ and $[v^2]_2$.

The resulting fully structure-preserving signature scheme is presented in Fig. 2 and can be used to sign messages consisting of $N = mn$ group elements in $\mathbb{G}_2$. It has a verification key size of 1 group elements, a signature size of $m + n + 1$ group elements, and verification involves evaluating $n + 1$ pairing product equations.

Theorem 2. *Fig. 2 gives a fully structure-preserving combined signature scheme that is C-EUF-CMA secure in the generic group model.*

Proof. Perfect correctness, perfect randomizability and structure-preservation follows by inspection. The secret key $sk = ([v]_2, [v\boldsymbol{x}]_2, [v\boldsymbol{y}]_2, [v^2]_2)$ consists of $m+n+1$ group elements and we can verify that it matches the verification key $vk = [v]_1$ by checking the pairing product equations

$$[v]_1[1]_2 = [1]_1[v]_2 \quad [v]_1[\boldsymbol{x}]_2 = [1]_1[v\boldsymbol{x}]_2 \quad [v]_1[\boldsymbol{y}]_2 = [1]_1[v\boldsymbol{y}]_2 \quad [v]_1[v]_2 = [1]_1[v^2]_2,$$

so the signature scheme is fully structure preserving.

What remains now is to prove that the signature scheme is C-EUF-CMA secure in the generic group model. In the (Type III) generic bilinear group model the adversary may compute new group elements in either source group by taking arbitrary linear combinations of previously seen group elements in the same source group. We shall see that no such linear combination of group elements, viewed as formal Laurent polynomials in the variables picked by the key generator and the signing oracle, yields an existential forgery. It follows along the lines of the Uber assumption in [BBG05] this that the signature scheme is C-EUF-CMA secure in the generic bilinear group model.

Suppose the adversary makes q queries $[M_i]_2 \in \mathbb{G}_2^{m \times n}$ to get signatures

$$[\boldsymbol{u}_i]_1 \qquad [r_i]_1 = [\frac{1}{z_i}]_1 \qquad [s_i]_2 = [z_i(y_1 + \boldsymbol{u}_i \cdot \boldsymbol{x} + v)]_2$$
$$[\boldsymbol{t}_i]_2 = [z_i\left((\boldsymbol{u}_i, 1)M_i + v\boldsymbol{y} + b_i z_i v(y_1 + \boldsymbol{u}_i \cdot \boldsymbol{x} + v)\right)]_2,$$

Setup$(1^\lambda, m, n)$
$gk = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, [1]_1, [1]_2) \leftarrow \mathcal{G}(1^\lambda)$
$[\boldsymbol{x}]_2 \leftarrow \mathbb{G}_2^{m-1}$
$[\boldsymbol{y}]_2 \leftarrow \mathbb{G}_2^{n}$
Return $pp = (gk, [\boldsymbol{x}]_2, [\boldsymbol{y}]_2)$

Gen(pp)
$v \leftarrow \mathbb{Z}_p$
$vk = [v]_1$
$sk = ([v]_2, [v\boldsymbol{x}]_2, [v\boldsymbol{y}]_2, [v^2]_2)$
Return (vk, sk)

Sign$_b(pp, sk, [M]_2)$
$\boldsymbol{u} \leftarrow \mathbb{Z}_p^{m-1}$, $z \leftarrow \mathbb{Z}_p^*$, $r = \frac{1}{z}$
$[s]_2 = z([y_1]_2 + \boldsymbol{u} \cdot [\boldsymbol{x}]_2 + [v]_2)$
$[\boldsymbol{t}]_2 = z \begin{pmatrix} (\boldsymbol{u}, 1)[M]_2 + [v\boldsymbol{y}]_2 \\ +bz([vy_1]_2 + \boldsymbol{u} \cdot [v\boldsymbol{x}]_2 + [v^2]_2)\mathbf{1} \end{pmatrix}$
Return $\sigma = ([\boldsymbol{u}]_1, [r]_1, [s]_2, [\boldsymbol{t}]_2)$

Vfy$_b(pp, vk, [M]_2, \sigma)$
Parse $\sigma = ([\boldsymbol{u}]_1, [r]_1, [s]_2, [\boldsymbol{t}]_2)$
Return 1 if and only if
$[M]_2 \in \mathbb{G}_2^{m \times n}$
$[r]_1 \in \mathbb{G}_1$, $[\boldsymbol{u}]_1 \in \mathbb{G}_1^{m-1}$
$[s]_2 \in \mathbb{G}_2$, $[\boldsymbol{t}]_2 \in \mathbb{G}_2^{n}$
$[r]_1[s]_2 = [1]_1[y_1]_2 + [\boldsymbol{u}]_1 \cdot [\boldsymbol{x}]_2 + [v]_1[1]_2$
$[r]_1[\boldsymbol{t}]_2 = [(\boldsymbol{u}, 1)]_1[M]_2 + [v]_1[\boldsymbol{y}]_2 + b[v]_1[s]_2\mathbf{1}$

Rand(pp, vk, M, σ)
Parse $\sigma = ([\boldsymbol{u}]_1, [r]_1, [s]_2, [\boldsymbol{t}]_2)$
$\boldsymbol{\alpha} \leftarrow \mathbb{Z}_p^{m-1}$
$\beta \leftarrow \mathbb{Z}_p^*$
$[\boldsymbol{u}']_1 = [\boldsymbol{u}]_1 + \boldsymbol{\alpha}[r]_1$
$[r']_1 = \frac{1}{\beta}[r]_1$
$[s']_2 = \beta([s]_2 + \boldsymbol{\alpha}[\boldsymbol{x}]_2)$
$[\boldsymbol{t}']_2 = \beta([t]_2 + (\boldsymbol{\alpha}, 0)[M]_2)$
Return $\sigma' = ([\boldsymbol{u}']_1, [r']_1, [s']_2, [\boldsymbol{t}']_2)$

Fig. 2. Fully structure-preserving combined signature scheme. Since they are quite similar we have described the randomizable signature and the strongly unforgable signature algorithms jointly. Setting $b = 0$ gives the algorithms for randomizable signatures and setting $b = 1$ gives the algorithms for strongly unforgeable signatures.

where $b_i = 0$ if query i is for a randomizable signature and $b_i = 1$ if query i is for a strong signature, and where M_i may depend on previously seen signature elements in $[s_j]_2, [\boldsymbol{t}_j]_2$ for $j < i$.

Viewed as Laurent polynomials we have that a signature $([\boldsymbol{u}]_1, [r]_1, [s]_2, [\boldsymbol{t}]_2)$ generated by the adversary on $[M] \in \mathbb{G}_2^{m \times n}$ is of the form

$$
\begin{aligned}
\boldsymbol{u} &= \boldsymbol{\alpha} + v\boldsymbol{\alpha}_v + \sum_i \boldsymbol{u}_i A_i + \sum_i \frac{1}{z_i}\boldsymbol{\alpha}_{r_i} \\
r &= \rho + v\rho_v + \sum_i \boldsymbol{u}_i \boldsymbol{\rho}_{u_i}^\top + \sum_i \frac{1}{z_i}\rho_{r_i} \\
s &= \sigma + \boldsymbol{\sigma}_x \boldsymbol{x}^\top + \boldsymbol{\sigma}_y \boldsymbol{y}^\top + \sum_j \sigma_{s_j} z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v) \\
&\quad + \sum_j \boldsymbol{\sigma}_{t_j} z_j \left((\boldsymbol{u}_j, 1) M_j + v\boldsymbol{y} + b_j z_j v (y_1 + \boldsymbol{u}\boldsymbol{x}^\top + v)\mathbf{1} \right) \\
\boldsymbol{t} &= \boldsymbol{\tau} + \boldsymbol{x}T_x + \boldsymbol{y}T_y + \sum_j z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v)\boldsymbol{\tau}_{s_j} \\
&\quad + \sum_j z_j \left((\boldsymbol{u}_j, 1) M_j + v\boldsymbol{y} + b_j z_j v (y_1 + \boldsymbol{u}\boldsymbol{x}^\top + v)\mathbf{1} \right) T_{t_j}
\end{aligned}
$$

Similarly, all mn entries in M can be written on a form similar to s and all entries in queried matrices M_i can be written on a form similar to s where the sums are bounded by $j < i$.

For the first verification equation to be satisfied we must have $rs = y_1 + \boldsymbol{u}\boldsymbol{x}^\top + v$, i.e.,

$$\begin{pmatrix} \rho + \sum_i \boldsymbol{u}_i \boldsymbol{\rho}_{u_i}^\top \\ +v\rho_v + \sum_i \frac{1}{z_i}\rho_{r_i} \end{pmatrix} \cdot \begin{pmatrix} \sigma + \boldsymbol{\sigma}_x \boldsymbol{x}^\top + \boldsymbol{\sigma}_y \boldsymbol{y}^\top + \sum_j \sigma_{s_j} z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v) \\ + \sum_j \boldsymbol{\sigma}_{t_j} z_j \big((\boldsymbol{u}_j, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v)\mathbf{1}\big)^\top \end{pmatrix}$$
$$= y_1 + \left(\boldsymbol{\alpha} + v\boldsymbol{\alpha}_v + \sum_i \boldsymbol{u}_i A_i + \sum_i \frac{1}{z_i} \boldsymbol{\alpha}_{r_i} \right) \boldsymbol{x}^\top + v$$

We start by noting that $r \neq 0$ since otherwise rs cannot have the term y_1. Please observe that it is only in $\mathbb{G}_1$ that we have terms including indeterminates with negative power, i.e., $\frac{1}{z_i}$. In $\mathbb{G}_2$ all indeterminates have positive power, i.e., so $s_j, \boldsymbol{t}_j, M_j$ only contain proper multi-variate polynomials. Now suppose for a moment that $\rho_{r_i} = 0$ for all i. Then in order not to have a terms involving z_j's in rs we must have

$$\sum_j \sigma_{s_j} z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v) + \sum_j \boldsymbol{\sigma}_{t_j} z_j \left((\boldsymbol{u}_j, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v)\mathbf{1}\right)^\top = 0.$$

The term y_1 now gives us $\rho\sigma_{y,1} = 1$ and the term v gives us $\rho_v\sigma = 1$. This means $\rho \neq 0$ and $\sigma \neq 0$ and therefore we reach a contradiction since the constant term should be $\rho\sigma = 0$. We conclude that there must exist some ℓ for which $\rho_{r_\ell} \neq 0$.

Now we have the term $\rho_{r_\ell}\sigma\frac{1}{z_\ell} = 0$, which shows us $\sigma = 0$. The terms $\rho_{r_\ell}\sigma_{y,k}\frac{y_k}{z_\ell} = 0$ for $k = 1, \ldots, n$ give us $\boldsymbol{\sigma}_y = \mathbf{0}$.

The polynomials corresponding to s_j and $\boldsymbol{t}_j$ contain the indeterminate z_j in all terms, so no linear combination of them can give us a term where the indeterminate component is vy_k for some $k \in \{1, \ldots, n\}$. Since M_j is constructed as a linear combination of elements in the verification key and components in $\mathbb{G}_2$ from previously seen signatures, it too cannot contain a term where the indeterminate component is vy_k. The coefficient of $\frac{z_j}{z_\ell}vy_k$ is therefore $\rho_{r_\ell}\sigma_{t_j,k} = 0$ and therefore $\sigma_{t_j,k} = 0$ for every $j \neq \ell$ and $k \in \{1, \ldots, n\}$. This shows $\boldsymbol{\sigma}_{t_j} = \mathbf{0}$ for all $j \neq \ell$. Looking at the coefficients for vy_k for $k = 1, \ldots, n$ we see that $\boldsymbol{\sigma}_{t_\ell} = \mathbf{0}$ too.

The terms $\rho_{r_\ell}\sigma_{s_j}\frac{z_j}{z_l}v$ give us $\sigma_{s_j} = 0$ for all $j \neq \ell$. In order to get a coefficient of 1 for the term y_1 we see that $\sigma_{s_\ell} = \frac{1}{\rho_{r_\ell}}$, which is non-zero. Our analysis has now shown that

$$s = \boldsymbol{\sigma}_x \boldsymbol{x}^\top + \frac{1}{\rho_{r_\ell}} z_\ell (y_1 + \boldsymbol{u}_\ell \boldsymbol{x}^\top + v).$$

Let us now analyze the structure of r. The term $\rho_v\sigma_\ell v^2 z_\ell = 0$ gives us $\rho_v = 0$. We know from our previous analysis that if there was a second $i \neq \ell$ for which $\rho_{r_i} \neq 0$ then also $\sigma_{\rho_\ell} = 0$, which it is not. Therefore for all $i \neq \ell$ we have $\rho_{r_i} = 0$. The term $\rho\sigma_{s_\ell} z_\ell y_1$ gives $\rho = 0$. The terms in $\boldsymbol{\rho}_{u_i}\sigma_{s_\ell}\boldsymbol{u}_i z_\ell v$ give us $\boldsymbol{\rho}_{u_i} = \mathbf{0}$ for all

i. Our analysis therefore shows

$$r = \rho_{r_\ell} \frac{1}{z_\ell}.$$

Finally, having simplifed r and s analysing the terms in $\boldsymbol{u}$ gives us

$$\boldsymbol{u} = \boldsymbol{u}_\ell + \rho_{r_\ell} \boldsymbol{\sigma}_x \frac{1}{z_\ell}.$$

We now turn to the second verification equation, which is $rt_1 = (\boldsymbol{u}, 1)\boldsymbol{m}^\top + vy_1 + bvs$, where $\boldsymbol{m}^\top$ is the first column vector of M. The message vector is of the form

$$\boldsymbol{m} = \begin{array}{l} \boldsymbol{\mu} + \boldsymbol{x}M_x + \boldsymbol{y}M_y + \sum_j \boldsymbol{\mu}_{s_j} z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v) \\ + \sum_j z_j \left((\boldsymbol{u}_j, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v)\mathbf{1} \right) M_{t_j} \end{array},$$

where $\boldsymbol{\mu}, M_x, M_y \boldsymbol{\mu}_{s_j}$ and M_{t_j} are suitably sized vectors and matrices with entries in $\mathbb{Z}_p$ chosen by the adversary. Similarly, we can write out $t_1 = \tau + \boldsymbol{\tau}_x \boldsymbol{x}^\top + \boldsymbol{\tau}_y \boldsymbol{y}^\top + \sum_j \tau_{s_j} z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v) + \sum_j \boldsymbol{\tau}_{t_j} z_j \left((\boldsymbol{u}, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v)\mathbf{1} \right)$ for elements and suitably sized vectors $\tau, \boldsymbol{\tau}_x, \boldsymbol{\tau}_y, \tau_{s_j}, \boldsymbol{\tau}_{t_j}$ with entries in $\mathbb{Z}_p$ chosen by the adversary.

Writing out the second verification equation we have

$$\begin{aligned} & \rho_{r_\ell} \frac{1}{z_\ell} \begin{pmatrix} \tau + \boldsymbol{\tau}_x \boldsymbol{x}^\top + \boldsymbol{\tau}_y \boldsymbol{y}^\top + \sum_j \tau_{s_j} z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v) \\ + \sum_j \boldsymbol{\tau}_{t_j} z_j \left((\boldsymbol{u}_j, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v)\mathbf{1} \right)^\top \end{pmatrix} \\ = {} & vy_1 + bv \left(\boldsymbol{\sigma}_x \boldsymbol{x}^\top + \frac{1}{\rho_{r_\ell}} z_\ell (y_1 + \boldsymbol{u}_\ell \boldsymbol{x}^\top + v) \right) \\ + {} & \left(\boldsymbol{u}_\ell + \rho_{r_\ell} \boldsymbol{\sigma}_x \frac{1}{z_\ell}, 1 \right) \begin{pmatrix} \boldsymbol{\mu} + \boldsymbol{x}M_x + \boldsymbol{y}M_y + \sum_j \boldsymbol{\mu}_{s_j} z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v) \\ + \sum_j z_j \left((\boldsymbol{u}_j, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v)\mathbf{1} \right) M_{t_j} \end{pmatrix}^\top. \end{aligned}$$

Looking at the coefficients of terms involving $\frac{1}{z_\ell}$ we get the following equalities for all $j \neq \ell$: $\tau = \boldsymbol{\sigma}_x \mu^\top \ (\frac{1}{z_\ell})$, $\boldsymbol{\tau}_x = \boldsymbol{\sigma}_x M_x^\top \ (\frac{x_k}{z_\ell})$, $\boldsymbol{\tau}_y = \boldsymbol{\sigma}_x M_y^\top \ (\frac{y_k}{z_\ell})$, $\tau_{s_j} = \boldsymbol{\sigma}_x \boldsymbol{\mu}_{s_j}^\top \ (\frac{v z_j}{z_\ell})$, $\boldsymbol{\tau}_{t_j} = \boldsymbol{\sigma}_x T_{t_j}^\top \ (\frac{v y_k z_j}{z_\ell})$. Cancelling out these terms we are left with

$$\begin{aligned} & \rho_{r_\ell} \left(\tau_{s_\ell} (y_1 + \boldsymbol{u}_\ell \boldsymbol{x}^\top + v) + \boldsymbol{\tau}_{t_\ell} \left((\boldsymbol{u}_\ell, 1) M_\ell + v\boldsymbol{y} + b_\ell v z_\ell (y_1 + \boldsymbol{u}_\ell \boldsymbol{x}^\top + v)\mathbf{1} \right)^\top \right) \\ = {} & vy_1 + bv \left(\boldsymbol{\sigma}_x \boldsymbol{x}^\top + \frac{1}{\rho_{r_\ell}} z_\ell (y_1 + \boldsymbol{u}_\ell \boldsymbol{x}^\top + v) \right) \\ + {} & \rho_{r_\ell} \boldsymbol{\sigma}_x \left(\boldsymbol{\mu}_{s_\ell} (y_1 + \boldsymbol{u}_\ell \boldsymbol{x}^\top + v) + \left((\boldsymbol{u}_\ell, 1) M_\ell + v\boldsymbol{y} + b_\ell v z_\ell (y_1 + \boldsymbol{u}_\ell \boldsymbol{x}^\top + v)\mathbf{1} \right) M_{t_\ell} \right)^\top \\ + {} & (\boldsymbol{u}_\ell, 1) \begin{pmatrix} \boldsymbol{\mu} + \boldsymbol{x}M_x + \boldsymbol{y}M_y + \sum_j \boldsymbol{\mu}_{s_j} z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v) \\ + \sum_j z_j \left((\boldsymbol{u}_j, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v)\mathbf{1} \right) M_{t_j} \end{pmatrix}^\top. \end{aligned}$$

Terms involving z_j and z_j^2 must cancel out, so we can assume $\mu_{s_j} = \mathbf{0}$ and $M_{t_j} = 0$ for $j > \ell$. Since M_ℓ does not involve z_ℓ in any of its terms, we get from

the terms in $(\boldsymbol{u}_\ell, 1) z_\ell v \mu_{s_\ell}^\top$ that $\boldsymbol{\mu}_{s_\ell} = 0$. Since there can be no terms involving z_ℓ^2 we get $b_\ell \mathbf{1} M_{t_\ell}^\top = \mathbf{0}$. Looking at the coefficients for v we get $\tau_{s_\ell} = \boldsymbol{\sigma}_x \boldsymbol{\mu}_{s_\ell}$. This leaves us with

$$
\begin{aligned}
&\rho_{r_\ell} \boldsymbol{\tau}_{t_\ell} \left((\boldsymbol{u}_\ell, 1) M_\ell + v\boldsymbol{y} + b_\ell v z_\ell (y_1 + \boldsymbol{u}_\ell \boldsymbol{x}^\top + v) \mathbf{1} \right)^\top \\
&= v y_1 + bv \left(\boldsymbol{\sigma}_x \boldsymbol{x}^\top + \frac{1}{\rho_{r_\ell}} z_\ell (y_1 + \boldsymbol{u}_\ell \boldsymbol{x}^\top + v) \right) \\
&+ \rho_{r_\ell} \boldsymbol{\sigma}_x \left(\left((\boldsymbol{u}_\ell, 1) M_\ell + v\boldsymbol{y} \right) M_{t_\ell} \right)^\top \\
&+ (\boldsymbol{u}_\ell, 1) \begin{pmatrix} \boldsymbol{\mu} + \boldsymbol{x} M_x + \boldsymbol{y} M_y + \sum_{j<\ell} \boldsymbol{\mu}_{s_j} z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v) \\ + \sum_{j<\ell} z_j \left((\boldsymbol{u}_j, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v) \mathbf{1} \right) M_{t_j} \end{pmatrix}^\top \\
&+ (\boldsymbol{u}_\ell, 1) z_\ell \left((\boldsymbol{u}_\ell, 1) M_\ell + v\boldsymbol{y} \right) M_{t_\ell} \big)^\top .
\end{aligned}
$$

Looking at the terms involving $z_\ell v^2$ we see $\rho_{r_\ell} \boldsymbol{\tau}_{t_\ell} b_\ell \mathbf{1}^\top = b \frac{1}{\rho_{r_\ell}}$. The only remaining terms involving z_ℓ now give us $M_{t_\ell} = 0$. This gives us

$$
\begin{aligned}
&\rho_{r_\ell} \boldsymbol{\tau}_{t_\ell} \left((\boldsymbol{u}_\ell, 1) M_\ell + v\boldsymbol{y} \right)^\top \\
&= v y_1 + bv \boldsymbol{\sigma}_x \boldsymbol{x}^\top \\
&+ (\boldsymbol{u}_\ell, 1) \begin{pmatrix} \boldsymbol{\mu} + \boldsymbol{x} M_x + \boldsymbol{y} M_y + \sum_{j<\ell} \boldsymbol{\mu}_{s_j} z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v) \\ + \sum_{j<\ell} z_j \left((\boldsymbol{u}_j, 1) M_j + v\boldsymbol{y} + b_j v z_j (y_1 + \boldsymbol{u}_j \boldsymbol{x}^\top + v) \mathbf{1} \right) M_{t_j} \end{pmatrix}^\top
\end{aligned}
$$

Looking at the terms in $v\boldsymbol{y}$ we now get $\rho_{r_\ell} \boldsymbol{\tau}_{t_\ell} = (1, 0, \ldots, 0)$. This means $(\boldsymbol{u}_\ell, 1) \boldsymbol{m}_\ell^\top = b \boldsymbol{\sigma}_x \boldsymbol{x}^\top + (\boldsymbol{u}_\ell, 1) \boldsymbol{m}^\top$, where $\boldsymbol{m}_\ell^\top$ is the first column of M_ℓ. Looking at the coefficients of vx_k we see that if $b\boldsymbol{\sigma}_x = \mathbf{0}$. Since $\boldsymbol{m}_\ell$ and $\boldsymbol{m}$ are independent of $\boldsymbol{u}_\ell$ this means $\boldsymbol{m} = \boldsymbol{m}_\ell$.

A similar argument can applied to the remaining $n-1$ verification equations showing us that in all columns M and M_ℓ match. This means $M = M_\ell$, so the signature scheme is existentially unforgeable both for randomizable signatures and strong signatures.

Finally, let us consider the case where $b = 1$, i.e., we are doing a strong signature verification. We have already seen that $b\boldsymbol{\sigma}_x = \mathbf{0}$ so when $b = 1$ this means $\boldsymbol{\sigma}_x = \mathbf{0}$. Since $\rho_{r_\ell} \boldsymbol{\tau}_{t_\ell} b_\ell \mathbf{1}^\top = b_\ell = b \frac{1}{\rho_{r_\ell}}$ we see that $b_\ell = 1$ and $\rho_{r_\ell} = 1$. This means $s = s_\ell$ and $r = r_\ell$ and $\boldsymbol{u} = \boldsymbol{u}_\ell$ and $M = M_\ell$ and therefore $\boldsymbol{t} = \boldsymbol{t}_\ell$. So the generic adversary can only satisfy the strong verification equation with $b = 1$ by copying both the message and signature from a previous query with $b_\ell = 1$.

On the other hand, if we have $b = 0$, i.e., we are verifying a randomizable signature, we see from $\rho_{r_\ell} \boldsymbol{\tau}_{t_\ell} b_l \mathbf{1}^\top = b_\ell = b \frac{1}{\rho_{r_\ell}}$ that $b_\ell = 0$. So the adversary has randomized a signature intended for randomization. □

Acknowledgment

We thank Masayuki Abe, Markulf Kohlweiss, Miyako Ohkubo and Mehdi Tibouchi for their comments and sharing an early version of [AKOT15] with us.

References

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

[ADK+13] Masayuki Abe, Bernardo David, Markulf Kohlweiss, Ryo Nishimaki, and Miyako Ohkubo. Tagged one-time signatures: Tight security and optimal tag size. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC*, volume 7778 of *LNCS*, pages 312–331. Springer, 2013.

[AFG+10] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, pages 209–236, 2010.

[AGHO11] Masayuki Abe, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. In *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 649–666, 2011.

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

[AGOT14] Masayuki Abe, Jens Groth, Miyako Ohkubo, and Mehdi Tibouchi. Unified, minimal and selectively randomizable structure-preserving signatures. In *TCC*, volume 8349 of *Lecture Notes in Computer Science*, pages 688–712, 2014.

[AHO12] Masayuki Abe, Kristiyan Haralambiev, and Miyako Ohkubo. Group to group commitments do not shrink. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 301–317, 2012.

[AKOT15] Masayuki Abe, Markulf Kohlweiss, Miyako Ohkubo, and Mehdi Tibouchi. Fully structure-preserving signatures and shrinking commitments. In *EUROCRYPT*, volume 9057 of *Lecture Notes in Computer Science*, pages 35–65, 2015.

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

[ALP13] Nuttapong Attrapadung, Benoît Libert, and Thomas Peters. Efficient completely context-hiding quotable and linearly homomorphic signatures. In *PKC*, volume 7778 of *Lecture Notes in Computer Science*, pages 386–404, 2013.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. Cryptology ePrint Archive, Report 2005/015, 2005.

[BCPW15] Fabrice Benhamouda, Geoffroy Couteau, David Pointcheval, and Hoeteck Wee. Implicit zero-knowledge arguments and applications to the malicious setting. In *CRYPTO*, volume 9216 of *Lecture Notes in Computer Science*, pages 107–129, 2015.

[CDEN12] Jan Camenisch, Maria Dubovitskaya, Robert R. Enderlein, and Gregory Neven. Oblivious transfer with hidden access control from attribute-based encryption. In *SCN*, volume 7485 of *Lecture Notes in Computer Science*, pages 559–579, 2012.

[CKLM12] Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable proof systems and applications. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 281–300, 2012.

[CM15] Sanjit Chatterjee and Alfred Menezes. Type 2 structure-preserving signature schemes revisited. In *ASIACRYPT*, 2015.

[EHK+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge L. Villar. An algebraic framework for diffie-hellman assumptions. In *CRYPTO*, volume 8043 of *Lecture Notes in Computer Science*, pages 129–147, 2013.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[Fuc11] Georg Fuchsbauer. Commuting signatures and verifiable encryption. In *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 224–245, 2011.

[FV10] Georg Fuchsbauer and Damien Vergnaud. Fair blind signatures without random oracles. In *AFRICACRYPT*, volume 6055 of *Lecture Notes in Computer Science*, pages 16–33, 2010.

[GH08] Matthew Green and Susan Hohenberger. Universally composable adaptive oblivious transfer. In *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 179–197, 2008.

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

[Gro06] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *ASIACRYPT*, volume 4248 of *Lecture Notes in Computer Science*, pages 444–459, 2006.

[GS12] Jens Groth and Amit Sahai. Efficient noninteractive proof systems for bilinear groups. *SIAM Journal on Computing*, 41(5):1193–1232, 2012.

[HJ12] Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. In *CRYPTO*, volume 7417 of *LNCS*, pages 590–607. Springer, 2012.

[LPJY13] Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Linearly homomorphic structure-preserving signatures and their applications. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pages 289–307, 2013.

[LPY12] Benoît Libert, Thomas Peters, and Moti Yung. Group signatures with almost-for-free revocation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO*, volume 7417 of *LNCS*, pages 571–589. Springer, 2012.

[LPY15] Benoît Libert, Thomas Peters, and Moti Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In *CRYPTO*, volume 9216 of *Lecture Notes in Computer Science*, pages 296–316, 2015.

[Nec94] Vasilii I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mat. Zametki*, 55(2):91–101, 1994.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266, 1997.

[ZLG12] Jiangxiao Zhang, Zhoujun Li, and Hua Guo. Anonymous transferable conditional e-cash. In Angelos D. Keromytis and Roberto Di Pietro, editors, *SecureComm*, volume 106 of *LNICST*, pages 45–60. Springer, 2012.