Constant-Size Structure-Preserving Signatures Generic Constructions and Simple Assumptions

Masayuki Abe, Melissa Chase, Bernardo David,
Markulf Kohlweiss, Ryo Nishimaki, and Miyako Ohkubo

NTT Secure Platform Laboratories
{abe.masayuki,nishimaki.ryo}@lab.ntt.co.jp

Microsoft Research
{melissac,markulf}@microsoft.com

University of Brasilia,
bernardo.david@aluno.unb.br

Security Architecture Laboratory, NSRI, NICT
m.ohkubo@nict.go.jp

Abstract. This paper presents efficient structure-preserving signature schemes based on assumptions as simple as Decisional-Linear. We first give two general frameworks for constructing fully secure signature schemes from weaker building blocks such as variations of one-time signatures and random-message secure signatures. They can be seen as refinements of the Even-Goldreich-Micali framework, and preserve many desirable properties of the underlying schemes such as constant signature size *and structure preservation*. We then instantiate them based on simple (i.e., not q-type) assumptions over symmetric and asymmetric bilinear groups. The resulting schemes are structure-preserving and yield constant-size signatures consisting of 11 to 17 group elements, which compares favorably to existing schemes relying on q-type assumptions for their security.

Keywords. Structure-preserving signatures, One-time signatures, Groth-Sahai proof system, Random message attacks

1 Introduction

A structure-preserving signature (SPS) scheme [1] is a digital signature scheme with two structural properties (i) the verification keys, messages, and signatures are all elements of a bilinear group; and (ii) the verification algorithm checks a conjunction of pairing product equations over the key, the message and the signature. This makes them compatible with the efficient non-interactive proof system for pairing-product equations by Groth and Sahai (GS) [30]. Structure-preserving cryptographic primitives promise to combine the advantages of optimized number theoretic non-blackbox constructions with the modularity and insight of protocols that use only generic cryptographic building blocks.

Indeed the instantiation of known generic constructions with a SPS scheme and the GS proof system has led to many new and more efficient schemes: Groth [29] showed how to construct an efficient simulation-sound zero-knowledge proof system (ss-NIZK) building on generic constructions of [17, 39, 34]. Abe et al. [4] show how to obtain efficient round-optimal blind signatures by instantiating a framework by Fischlin [20]. SPS are also important building blocks for a wide range of cryptographic functionalities such as anonymous proxy signatures [22], delegatable anonymous credentials [6], transferable e-cash [23] and compact ver-

ifiable shuffles [16]. Most recently, [31] show how to construct a structure preserving tree-based signature scheme with a tight security reduction following the approach of [26, 18]. This signature scheme is then used to build a ss-NIZK which in turn is used with the Naor-Yung-Sahai [35, 38] paradigm to build the first CCA secure public-key encryption scheme with a tight security reduction. Examples for other schemes that benefit from efficient SPS are [7, 11, 8, 32, 27, 5, 37, 24, 21, 28].

Because properties (i) and (ii) are the only dependencies on the SPS scheme made by these constructions, any structure-preserving signature scheme can be used as a drop-in replacement. Unfortunately, all known efficient instantiations of SPS [4, 1, 2] are based on so-called q-type or interactive assumptions that are primarily justified based on the Generic Group model. An open question since Groth's seminal work [29] (only partially answered by [15]) is to construct a SPS scheme that is both efficient – in particular *constant-size* in the number of signed group elements – and that is based on assumptions that are as weak as those required by the GS proof system itself.

Our contribution. Our first contribution consists of two generic constructions for chosen message attack (CMA) secure signatures that combine variations of one-time signatures and signatures secure against random message attacks (RMA). Both constructions inherit the structure-preserving and constant-size properties from the underlying components. The second contribution consists in the concrete instantiations of these components which result in constant-size structure-preserving signature schemes that produce signatures consisting of only 11 to 17 group elements and that rely only on basic assumptions such as Decisional-Linear (DLIN) for symmetric bilinear groups and analogues of DDH and DLIN for asymmetric bilinear groups. To our knowledge, these are the first constant-size structure-preserving signature schemes that eliminate the use of q-type assumptions while achieving reasonable efficiency.

We instantiate the first generic construction for symmetric (Type-I) and the second for asymmetric (Type-III) pairing groups. See Table 1 in Section 7 for the summary of efficiency of the resulting schemes. We give more details on our generic constructions and their instantiations:

- The first generic construction ($\mathsf{SIG1}$) combines a new variation of one-time signatures which we call *tagged one-time signatures* and signatures secure against *random message attacks* (RMA). A tagged one-time signature scheme, denoted by TOS, is a signature scheme that attaches a fresh tag to a signature. It is unforgeable with respect to tags that are used only once. In our construction, a message is signed with our TOS scheme using a fresh random tag, and then the tag is signed with the second signature scheme, denoted by rSIG. Since the rSIG scheme only signs random tags, RMA-security is sufficient.
- The second generic construction ($\mathsf{SIG2}$) combines *partial one-time signatures* and signatures secure against *extended random message attacks* (XRMA). The latter is a novel notion that we explain below. Partial one-time signatures, denoted by POS, are one-time signatures for which only a part of the one-time key is renewed for every signing operation. They were first introduced by Bellare and Shoup [9] under the name of two-tier signatures. In our construction, a message is signed with the POS scheme and then the random one-time public-key is certified by the second signature scheme, denoted by xSIG. The difference between a TOS scheme and a POS scheme is that a one-time public-key is associated with a one-time secret-key. Since the one-time secret-key is needed for signing, it must be known to the reduction in the security proof. XRMA-security guarantees that xSIG is unforgeable even if the adversary is given auxiliary information associated with the randomly chosen messages (it is a random coin used for selecting the message). The auxiliary information facilitates access to the one-time secret-key by the reduction.

- To instantiate SIG1, we construct structure-preserving TOS and rSIG signature schemes based on DLIN over Type-I bilinear groups. Our TOS scheme yields constant-size signatures and tags. The resulting SIG1 scheme is structure-preserving, produces signatures consisting of 17 group elements, and relies solely on the DLIN assumption.
- To instantiate SIG2, we construct structure-preserving POS and xSIG signature schemes based on assumptions that are analogues of DDH and DLIN in Type-III bilinear groups. The resulting SIG2 scheme is structure-preserving, produces signatures consisting of 11 group elements for uniliteral messages in a base group or 14 group elements for biliteral messages from both base groups.

The role of partial one-time signatures is to compress a message into a constant number of random group elements. This observation is interesting in light of [3] that implies the impossibility of constructing collision resistant and shrinking structure-preserving hash functions, which could immediately yield constant-size signatures. Our (extended) RMA-secure signature schemes are structure-preserving variants of Waters' dual-signature scheme [41]. In general, the difficulty of constructing CMA-secure SPS arises from the fact that the exponents of the group elements chosen by the adversary as a message are not known to the reduction in the security proof. On the other hand, for RMA security, it is the challenger that chooses the message and therefore the exponents can be known in reductions. This is the crucial advantage for constructing (extended) RMA-secure structure-preserving signature schemes based on Waters' dual-signature scheme.

Finally, we mention a few new applications. Among these is the achievement of a drastic performance improvement when using our partial one-time signatures in the work by Hofheinz and Jager [31] to construct CCA-secure public-key encryption schemes with a proof of security that tightly reduces to DLIN or SXDH.

Related Works. Even, Goldreich and Micali [19] proposed a generic framework (the EGM framework) that combines a one-time signature scheme and a signature scheme that is secure against non-adaptive chosen message attacks (NACMA) to construct a signature scheme that is secure against adaptive chosen message attacks (CMA).

In fact, our generic constructions can be seen as refinements of the EGM framework. There are two reasons why the original framework falls short for our purpose. *The first* is that relaxing to NACMA does not seem a big help in constructing efficient structure-preserving signatures since the messages are still under the control of the adversary and the exponents of the messages are not known to the reduction algorithm in the security proof. As mentioned above, resorting to (extended) RMA is a great help in this regard. In [19], they also showed that CMA-secure signatures exist *iff* RMA-secure signatures exist. The proof, however, does not follow their framework and their impractical construction is mainly a feasibility result. In fact, we argue that RMA-security alone is not sufficient for the original EGM framework. As mentioned above, the necessity of XRMA security arises in the reduction that uses RMA-security to argue security of the ordinary signature scheme, as the reduction not only needs to know the random one-time public-keys, but also their corresponding one-time secret keys in order to generate the one-time signature components of the signatures. The auxiliary information in the XRMA definition facilitates access to these secret keys. Similarly, tagged one-time signatures avoid this problem as tags do not have associated secret values. *The second reason* that the EGM approach is not quite suited to our task is that the EGM framework produces signatures that are linear in the public-key size of the one-time signature scheme. Here, tagged or partial one-time signature schemes come in handy as they allow the signature size to be only linear in

the size of the part of the public key that is updated. Thus, to obtain constant-size signatures, we require the one-time part to be constant-size.

Hofheinz and Jager [31] constructed a SPS scheme by following the EGM framework. The resulting scheme allows tight security reduction to DLIN but the size of signatures depends logarithmically to the number of signing operation as their NACMA-secure scheme is tree-based like the Goldwasser-Micali-Rivest signature scheme [26]. Chase and Kohlweiss [15] and Camenisch, Dubovitskaya, and Haralambiev [13] constructed SPS schemes with security based on DLIN that improve the performance of Groth's scheme [29] by several orders of magnitude. The size of the resulting signatures, however, are still linear in the number of signed group elements, and an order of magnitude larger than in our constructions. Camenisch, Dubovitskaya, and Haralambiev constructed a constant-size SPS scheme based on simple assumptions over composite-order groups [12].

Full Version. In this extended abstract, we do not have enough space to write complete proofs, so we omitted them. Please see a full version on Cryptology ePrint Archive (2012/285).

2 Preliminaries

Notation. Appending element y to a sequence $X = (x_1, \ldots, x_n)$ is denoted by (X, y), i.e., $(X, y) = (x_1, \ldots, x_n, y)$. When algorithm A is defined for input x and output y, notation $\boldsymbol{y} \leftarrow A(\boldsymbol{x})$ for $\boldsymbol{x} := \{x_1, \ldots, x_n\}$ means that $y_i \leftarrow A(x_i)$ is executed for $i = 1, \ldots, n$ and $\boldsymbol{y}$ is set as $\boldsymbol{y} := (y_1, \ldots, y_n)$. For set X, notation $a \leftarrow X$ denote a uniform sampling from X. Independent multiple sampling from the same set X is denoted by $a, b, c, .. \leftarrow X$.

Bilinear groups. Let $\mathcal{G}$ be a bilinear group generator that takes security parameter 1^λ and outputs a description of bilinear groups $\Lambda := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are groups of prime order p, and e is an efficient and non-degenerating bilinear map $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Following the terminology in [25] this is a Type-III pairing. In the Type-III setting $\mathbb{G}_1 \neq \mathbb{G}_2$ and there are no efficient mapping between the groups in either direction. In the Type-III setting, we often use twin group elements, $(G^a, \hat{G}^a) \in \mathbb{G}_1 \times \mathbb{G}_2$ for some bases G and $\hat{G}$. For X in $\mathbb{G}_1$, notation $\hat{X}$ denotes for an element in $\mathbb{G}_2$ that $\log X = \log \hat{X}$ where logarithms are with respect to default bases that are uniformly chosen once for all and implicitly associated to Λ. Should their relation be explicitly stated, we write $X \sim \hat{X}$. We count the number of group elements to measure the size of cryptographic objects such as keys, messages, and signatures. For Type-III groups, we denote the size by (x, y) when it consists of x and y elements from $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. We refer to the Type-I setting when $\mathbb{G}_1 = \mathbb{G}_2$ (i.e., there are efficient mappings in both directions). This is also called the symmetric setting. In this case, we define $\Lambda := (p, \mathbb{G}, \mathbb{G}_T, e)$. When we need to be specific, the group description yielded by $\mathcal{G}$ will be written as Λ_{asym} and Λ_{sym}.

Assumptions. We first define computational and decisional Diffie-Hellman assumptions (CDH_1, DDH_1) and decisional linear assumption (DLIN_1) for Type-III bilinear groups. Corresponding more standard assumptions, CDH, DDH, and DLIN, in Type-I groups are obtained by setting $\mathbb{G}_1 = \mathbb{G}_2$ and $G = \hat{G}$ in the respective definitions.

Definition 1 (Computation co-Diffie-Hellman Assumption: CDH_1).
The CDH_1 assumption holds if, for any p.p.t. algorithm $\mathcal{A}$, the probability $\mathrm{Adv}^{\mathsf{co\text{-}cdh}}_{\mathcal{G},\mathcal{A}}(\lambda) := \Pr[\, Z = G^{xy} \mid \Lambda \leftarrow \mathcal{G}(1^\lambda); x, y \leftarrow \mathbb{Z}_p; Z \leftarrow \mathcal{A}(\Lambda, G, G^x, G^y, \hat{G}, \hat{G}^x, \hat{G}^y)\,]$ *is negligible in λ.*

Definition 2 (Decisional Diffie-Hellman Assumption in $\mathbb{G}_1$: DDH_1).
Given $\Lambda \leftarrow \mathcal{G}(1^\lambda)$, $G \leftarrow \mathbb{G}_1^*$, $(G^x, G^y, Z_b) \in {\mathbb{G}_1}^3$ where $Z_1 = G^{x+y}$, $Z_0 \leftarrow \mathbb{G}_1$ for random x and y, any p.p.t. algorithm $\mathcal{A}$ decides whether $b = 1$ or 0 only with advantage $\mathrm{Adv}^{DDH_1}_{\mathcal{G},\mathcal{A}}(\lambda)$ *that is negligible in λ.*

Definition 3 (Decisional Linear Assumption in $\mathbb{G}_1$: DLIN$_1$).
Given $\Lambda \leftarrow \mathcal{G}(1^\lambda)$, $(G_1, G_2, G_3) \leftarrow \mathbb{G}_1^{*3}$ and (G_1^x, G_2^y, Z_b) where $Z_1 = G_3^{x+y}$ and $Z_0 = G_3^z$ for random $x, y, z \in \mathbb{Z}_p$, any p.p.t. algorithm $\mathcal{A}$ decides whether $b = 1$ or 0 only with advantage $\mathrm{Adv}^{\mathsf{dlin1}}_{\mathcal{G},\mathcal{A}}(\lambda)$ that is negligible in λ.

For DDH$_1$ and DLIN$_1$, we define an analogous assumption in $\mathbb{G}_2$ (DDH$_2$) by swapping $\mathbb{G}_1$ and $\mathbb{G}_2$ in the respective definitions. In Type-III bilinear groups, it is assumed that both DDH$_1$ and DDH$_2$ hold simultaneously. The assumption is called the symmetric external Diffie-Hellman assumption (SXDH), and we define advantage $\mathrm{Adv}^{\mathsf{sxdh}}_{\mathcal{G},\mathcal{C}}$ by $\mathrm{Adv}^{\mathsf{sxdh}}_{\mathcal{G},\mathcal{C}}(\lambda) := \mathrm{Adv}^{\mathsf{ddh1}}_{\mathcal{G},\mathcal{A}}(\lambda) + \mathrm{Adv}^{\mathsf{ddh2}}_{\mathcal{G},\mathcal{B}}(\lambda)$. We extend DLIN in a similar manner as DDH, and SXDH.

Definition 4 (External Decision Linear Assumption in $\mathbb{G}_1$: XDLIN$_1$).
Given $\Lambda \leftarrow \mathcal{G}(1^\lambda)$, $(G_1, G_2, G_3) \leftarrow \mathbb{G}_1^{*3}$ and $(G_1^x, G_2^y, \hat{G}_1, \hat{G}_2, \hat{G}_3, \hat{G}_1^x, \hat{G}_2^y, Z_b)$ where $(G_1, G_2, G_3) \sim (\hat{G}_1, \hat{G}_2, \hat{G}_3)$, $Z_1 = G_3^{x+y}$, and $Z_0 = G_3^z$ for random $x, y, z \in \mathbb{Z}_p$, any p.p.t. algorithm $\mathcal{A}$ decides whether $b = 1$ or 0 only with advantage $\mathrm{Adv}^{\mathsf{xdlin}}_{\mathcal{G},\mathcal{A}}(\lambda)$ that is negligible in λ.

The XDLIN$_1$ assumption is equivalent to the DLIN$_1$ assumption in the generic bilinear group model [40, 10] where one can simulate the extra elements, $\hat{G}_1, \hat{G}_2, \hat{G}_3, \hat{G}_1^x, \hat{G}_2^y$, in XDLIN$_1$ from $G_1, G_2, G_3, G_1^x, G_2^y$ in DLIN$_1$. We define the XDLIN$_2$ assumption analogously by giving $\hat{G}_3^{x+y}$ or $\hat{G}_3^z$ as Z_b, to $\mathcal{A}$ instead. Then we define the simultaneous external DLIN assumption, SXDLIN, that assumes that both XDLIN$_1$ and XDLIN$_2$ hold at the same time. By $\mathrm{Adv}^{\mathsf{xdlin2}}_{\mathcal{G},\mathcal{A}}$ ($\mathrm{Adv}^{\mathsf{sxdlin}}_{\mathcal{G},\mathcal{A}}$, resp.), we denote the advantage function for XDLIN$_2$ (and SXDLIN, resp.).

Definition 5 (Double Pairing Assumption in $\mathbb{G}_1$ [4]:DBP$_1$).
Given $\Lambda \leftarrow \mathcal{G}(1^\lambda)$ and $(G_z, G_r) \leftarrow \mathbb{G}_1^{*2}$, any p.p.t. algorithm $\mathcal{A}$ outputs $(Z, R) \in \mathbb{G}_2^{*2}$ that satisfies $1 = e(G_z, Z)\ e(G_r, R)$ only with probability $\mathrm{Adv}^{\mathsf{dbp1}}_{\mathcal{G},\mathcal{A}}(\lambda)$ that is negligible in λ.

The double pairing assumption in $\mathbb{G}_2$ (DBP$_2$) is defined in the same manner by swapping $\mathbb{G}_1$ and $\mathbb{G}_2$. It is known that DBP$_1$ (DBP$_2$, resp.) is implied by DDH$_1$ (DDH$_2$, resp.) and the reduction is tight [4]. Note that the double pairing assumption does not hold in Type-I groups since $Z = G_r$, $R = G_z^{-1}$ is a trivial solution. The following analogous assumption will be useful in Type-I groups.

Definition 6 (Simultaneous Double Pairing Assumption [14]: SDP).
Given $\Lambda \leftarrow \mathcal{G}(1^\lambda)$ and $(G_z, G_r, H_z, H_s) \leftarrow \mathbb{G}^{*4}$, any p.p.t. algorithm $\mathcal{A}$ outputs $(Z, R, S) \in \mathbb{G}^{*3}$ that satisfies $1 = e(G_z, Z)\ e(G_r, R) \wedge 1 = e(H_z, Z)\ e(H_s, S)$ only with probability $\mathrm{Adv}^{\mathsf{sdp}}_{\mathcal{G},\mathcal{A}}(\lambda)$ that is negligible in λ.

As shown in [14] for the Type-I setting, the simultaneous double pairing assumption holds for $\mathcal{G}$ if the decisional linear assumption holds for $\mathcal{G}$.

3 Definitions

Common setup. All building blocks make use of a common setup algorithm Setup that takes the security parameter 1^λ and outputs a global parameters gk that is given to all other algorithms. Usually gk consists of a description Λ of a bilinear group setup and a default generator for each group. In this paper, we include several additional generators in gk for technical reasons. Note that when the resulting signature scheme is used in multi-user applications different

additional generators need to be assigned to individual users or one needs to fall back on the common reference string model, whereas Λ and the default generators can be shared. Thus we count the size of gk when we assess the efficiency of concrete instantiations. For ease of notation, we make gk implicit except w.r.t. key generation algorithms.

Signature schemes. We use the following syntax for signature schemes suitable for the multi-user and multi-algorithm setting. The key generation function takes global parameter gk generated by Setup (usually it takes security parameter 1^λ), and the message space $\mathcal{M}$ is determined solely from gk (usually it is determined from a public-key).

Definition 7 (Signature Scheme). *A signature scheme* SIG *is a tuple of three polynomial-time algorithms* $(\mathsf{Key}, \mathsf{Sign}, \mathsf{Vrf})$ *that;*

- $\mathsf{SIG.Key}(gk)$ *generates a long-term public-key* vk *and a secret-key* sk*.*
- $\mathsf{SIG.Sign}(sk, msg)$ *takes* sk *and message* msg *and outputs signature* σ*.*
- $\mathsf{SIG.Vrf}(vk, msg, \sigma)$ *outputs* 1 *for acceptance or* 0 *for rejection.*

Correctness requires that $1 = \mathsf{SIG.Vrf}(vk, msg, \sigma)$ *holds for any* gk *generated by* Setup*, any keys generated as* $(vk, sk) \leftarrow \mathsf{SIG.Key}(gk)$*, any message* $msg \in \mathcal{M}$*, and any signature* $\sigma \leftarrow \mathsf{SIG.Sign}(sk, msg)$*.*

Definition 8 (Attack Game(ATK)). *Let* $\mathcal{O}sig$ *be an oracle and* $\mathcal{A}$ *be an oracle algorithm. We define a meta attack game as a sequence of execution of algorithms as follows: ATK*$(\mathcal{A}, \lambda) =$

$$\left[gk \leftarrow \mathsf{Setup}(1^\lambda), pre \leftarrow \mathcal{A}(gk), (vk, sk) \leftarrow \mathsf{SIG.Key}(gk), (\sigma^\dagger, msg^\dagger) \leftarrow \mathcal{A}^{\mathcal{O}sig}(vk)\right]$$

Adversary $\mathcal{A}$ *commits to* pre*, which is typically a set of messages, in the first run. This formulation is to capture non-adaptive attacks. It is implicit that a state information is passed to the second run of* $\mathcal{A}$*. Let* Q_m *be a set of messages, for which* $\mathcal{A}$ *requests signatures from its oracle before outputting the resulting forgery. The output of ATK is* $(vk, \sigma^\dagger, msg^\dagger, Q_m)$*.*

Definition 9 (Adaptive Chosen-Message Attack (CMA)). *Adaptive chosen message attack security is defined by the attack game ATK where* pre *is empty and oracle* $\mathcal{O}sig$ *is the signing oracle that, on receiving a message* msg*, performs* $\sigma \leftarrow \mathsf{SIG.Sign}(sk, msg)$*, and returns* σ*.*

Definition 10 (Random Message Attack (RMA)[19]). *Random message attack security is defined by the attack game ATK where* pre *is empty and oracle* $\mathcal{O}sig$ *is the following: on receiving a request, it chooses* msg *uniformly from* $\mathcal{M}$ *defined by* gk*, computes signature* $\sigma \leftarrow \mathsf{SIG.Sign}(sk, msg)$*, and returns* (σ, msg)*.*

Let MSGGen be a uniform message generator. It is a probabilistic algorithm that takes gk and outputs $msg \in \mathcal{M}$ that distributes uniformly over $\mathcal{M}$. Furthermore, MSGGen outputs auxiliary information *aux* that may give a hint about the random coins used for selecting msg.

Definition 11 (Extended Random Message Attack (XRMA)). *Extended random message attack is attack game ATK where* pre *is empty and oracle* $\mathcal{O}sig$ *is the following. On receiving a request, it runs* $(msg, aux) \leftarrow \mathsf{MSGGen}(gk)$*, computes* $\sigma \leftarrow \mathsf{SIG.Sign}(sk, msg)$*, and returns* (σ, msg, aux)*.*

Definition 12 (Unforgeability against ATK). *Signature scheme* SIG *is unforgeable against attack ATK (UF-ATK) where ATK* $\in \{$*CMA*, *RMA*, *XRMA*$\}$*, if for all p.p.t. oracle algorithm* $\mathcal{A}$ *the advantage function* $\mathrm{Adv}^{\mathsf{uf\text{-}atk}}_{\mathsf{SIG},\mathcal{A}} := \Pr\left[msg^\dagger \notin Q_m \ \wedge \ 1 = \mathsf{SIG.Vrf}(vk, \sigma^\dagger, msg^\dagger) \,\middle|\, (vk, \sigma^\dagger, msg^\dagger, Q_m) \leftarrow \mathit{ATK}(\mathcal{A}, \lambda)\right]$ *is negligibel in* λ*.*

Fact 1. UF-CMA $\Rightarrow$ UF-XRMA $\Rightarrow$ UF-RMA, i.e., $\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda) \geq \mathrm{Adv}^{\mathsf{uf\text{-}xrma}}_{\mathsf{SIG},\mathcal{A}}(\lambda) \geq \mathrm{Adv}^{\mathsf{uf\text{-}rma}}_{\mathsf{SIG},\mathcal{A}}(\lambda)$.

Partial one-time and tagged one-time signatures. Partial one-time signatures, also known as two-tier signatures [9], are a variation of one-time signatures where only part of the public-key must be updated for every signing, while the remaining part can be persistent.

Definition 13 (Partial One-Time Signature Scheme [9]). *A partial one-time signatures scheme* POS *is a set of polynomial-time algorithms* $\mathsf{POS}.\{\mathsf{Key}, \mathsf{Update}, \mathsf{Sign}, \mathsf{Vrf}\}$.

- $\mathsf{POS.Key}(gk)$ *generates a long-term public-key* pk *and a secret-key* sk*. The message space* $\mathcal{M}_o$ *is associated with* pk*. (Recall that we require that* $\mathcal{M}_o$ *be completely defined by* gk*.)*
- $\mathsf{POS.Update}()$ *takes* gk *as implicit input, and outputs a pair of one-time keys* (opk, osk)*. We denote the space for* opk *by* $\mathcal{K}_{opk}$*.*
- $\mathsf{POS.Sign}(sk, msg, osk)$ *outputs a signature* σ *on message* msg *based on* sk *and* osk*.*
- $\mathsf{POS.Vrf}(pk, opk, msg, \sigma)$ *outputs* 1 *for acceptance, or* 0 *for rejection.*

For correctness, it is required that $1 = \mathsf{POS.Vrf}(pk, opk, msg, \sigma)$ *holds except for negligible probability for any* gk*,* pk*,* opk*,* σ*, and* $msg \in \mathcal{M}_o$*, such that* $gk \leftarrow \mathsf{Setup}(1^\lambda)$*,* $(pk, sk) \leftarrow \mathsf{POS.Key}(gk)$*,* $(opk, osk) \leftarrow \mathsf{POS.Update}()$*,* $\sigma \leftarrow \mathsf{POS.Sign}(sk, msg, osk)$*.*

A tagged one-time signature scheme is a signature scheme whose signing function in addition to the long-term secret key takes a tag as input. A tag is one-time, i.e., it must be different for every signing.

Definition 14 (Tagged One-Time Signature Scheme). *A tagged one-time signature scheme* TOS *is a set of polynomial-time algorithms* $\mathsf{TOS}.\{\mathsf{Key}, \mathsf{Tag}, \mathsf{Sign}, \mathsf{Vrf}\}$.

- $\mathsf{TOS.Key}(gk)$ *generates a long-term public-key* pk *and a secret-key* sk*. The message space* $\mathcal{M}_t$ *is associated with* pk*.*
- $\mathsf{TOS.Tag}()$ *takes* gk *as implicit input and outputs* tag*. By* $\mathcal{T}$*, we denote the space for* tag*.*
- $\mathsf{TOS.Sign}(sk, msg, tag)$ *outputs signature* σ *for message* msg *based on* sk *and* tag*.*
- $\mathsf{TOS.Vrf}(pk, tag, msg, \sigma)$ *outputs* 1 *for acceptance, or* 0 *for rejection.*

Correctness requires that $1 = \mathsf{TOS.Vrf}(pk, tag, msg, \sigma)$ *holds except for negligible probability for any* gk*,* pk*,* tag*,* σ*, and* $msg \in \mathcal{M}_t$*, such that* $gk \leftarrow \mathsf{Setup}(1^\lambda)$*,* $(pk, sk) \leftarrow \mathsf{TOS.Key}(gk)$*,* $tag \leftarrow \mathsf{TOS.Tag}()$*,* $\sigma \leftarrow \mathsf{TOS.Sign}(sk, msg, tag)$*.*

A TOS scheme is POS scheme for which $tag = osk = opk$. We can thus give a security notion for POS schemes that also applies to TOS schemes by reading $\mathsf{Update} = \mathsf{Tag}$ and $tag = osk = opk$.

Definition 15 (Unforgeability against One-Time Adapative Chosen-Message Attacks). *A partial one-time signature scheme is unforgeable against one-time adaptive chosen message attacks (OT-CMA) if for all p.p.t. oracle algorithm* $\mathcal{A}$ *the advantage function* $\mathrm{Adv}^{\mathsf{ot\text{-}cma}}_{\mathsf{POS},\mathcal{A}}$ *is negligible in* λ*, where* $\mathrm{Adv}^{\mathsf{ot\text{-}cma}}_{\mathsf{POS},\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l|l} \exists(opk, msg, \sigma) \in Q_m \text{ s.t.} & gk \leftarrow \mathsf{Setup}(1^\lambda), \\ opk^\dagger = opk \ \wedge \ msg^\dagger \neq msg \ \wedge & (pk, sk) \leftarrow \mathsf{POS.Key}(gk), \\ 1 = \mathsf{POS.Vrf}(pk, opk^\dagger, \sigma^\dagger, msg^\dagger) & (opk^\dagger, \sigma^\dagger, msg^\dagger) \leftarrow \mathcal{A}^{\mathcal{O}t, \mathcal{O}sig}(pk) \end{array}\right].$$

Q_m is initially an empty list. $\mathcal{O}t$ is the one-time key generation oracle that on receiving a request invokes a fresh session j, performs $(opk_j, osk_j) \leftarrow \mathsf{POS.Update}()$, and returns opk_j. $\mathcal{O}sig$ is the signing oracle that, on receiving a message msg_j for session j, performs $\sigma_j \leftarrow \mathsf{POS.Sign}(sk, msg_j, osk_j)$, returns σ_j to $\mathcal{A}$, and records (opk_j, msg_j, σ_j) to the list Q_m. $\mathcal{O}sig$ works only once for every session. Strong unforgeability is defined as well by replacing condition $msg^\dagger \neq msg$ with $(msg^\dagger, \sigma^\dagger) \neq (msg, \sigma)$.

We define a non-adaptive variant (OT-NACMA) of the above notion by integrating $\mathcal{O}t$ into $\mathcal{O}sig$ so that opk_j and σ_j are returned to $\mathcal{A}$ at the same time. Namely, $\mathcal{A}$ must submit msg_j before seeing opk_j. If a scheme is secure in the sense of OT-CMA, the scheme is also secure in the sense of OT-NACMA. If a scheme is strongly unforgeable, it is unforgeable as well. By $\mathrm{Adv}^{\mathsf{ot\text{-}nacma}}_{\mathsf{POS},\mathcal{A}}(\lambda)$ we denote the advantage of $\mathcal{A}$ in this non-adaptive case. For TOS, we use the same notations, OT-CMA and OT-NACMA, and define advantage functions $\mathrm{Adv}^{\mathsf{ot\text{-}cma}}_{\mathsf{TOS},\mathcal{A}}$ and $\mathrm{Adv}^{\mathsf{ot\text{-}nacma}}_{\mathsf{TOS},\mathcal{A}}$ accordingly. For strong unforgeabiltiy, we use label sot-cma and sot-nacma.

We define a condition that is relevant for coupling random message secure signature schemes with partial one-time and tagged one-time signature schemes in later sections.

Definition 16 (Tag/One-time Public-Key Uniformity). TOS *is called uniform-tag if* TOS.Tag *outputs tag that uniformly distributes over tag space $\mathcal{T}$. Similarly,* POS *is called uniform-key if* POS.Update *outputs opk that uniformly distributes over key space $\mathcal{K}_{opk}$.*

Structure-preserving signatures. A signature scheme is structure-preserving over a bilinear group Λ, if public-keys, signatures, and messages are all base group elements of Λ, and the verification only evaluates pairing product equations. Similarly, POS schemes are structure-preserving if their public-keys, signatures, messages, and tags or one-time public-keys consist of base group elements and the verification only evaluates pairing product equations.

4 Generic Constructions

4.1 SIG1: Combining tagged one-time and RMA-secure signatures

Let rSIG be a signature scheme with message space $\mathcal{M}_{\mathsf{r}}$, and TOS be a tagged one-time signature scheme with tag space $\mathcal{T}$ such that $\mathcal{M}_{\mathsf{r}} = \mathcal{T}$. We construct a signature scheme SIG1 from rSIG and TOS. Let gk be a global parameter generated by $\mathsf{Setup}(1^\lambda)$.

- $\mathsf{SIG1.Key}(gk)$: Run $(pk_t, sk_t) \leftarrow \mathsf{TOS.Key}(gk)$, $(vk_r, sk_r) \leftarrow \mathsf{rSIG.Key}(gk)$. Output $vk := (pk_t, vk_r)$ and $sk := (sk_t, sk_r)$.
- $\mathsf{SIG1.Sign}(sk, msg)$: Parse sk into (sk_t, sk_r). Run $tag \leftarrow \mathsf{TOS.Tag}()$, $\sigma_t \leftarrow \mathsf{TOS.Sign}(sk_t, msg, tag)$, $\sigma_r \leftarrow \mathsf{rSIG.Sign}(sk_r, tag)$. Output $\sigma := (tag, \sigma_t, \sigma_r)$.
- $\mathsf{SIG1.Vrf}(vk, \sigma, msg)$: Parse vk and σ accordingly. Output 1, if $1 = \mathsf{TOS.Vrf}(pk_t, tag, \sigma_t, msg)$ and $1 = \mathsf{rSIG.Vrf}(vk_r, \sigma_r, tag)$. Output 0, otherwise.

We prove the above scheme is secure by showing a reduction to the security of each component. As our reductions are efficient in their running time, we only relate success probabilities.

Theorem 17. SIG1 *is UF-CMA if* TOS *is uniform-tag and OT-NACMA, and* rSIG *is UF-RMA. In particular,* $\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SIG1},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathsf{ot\text{-}nacma}}_{\mathsf{TOS},\mathcal{B}}(\lambda) + \mathrm{Adv}^{\mathsf{uf\text{-}rma}}_{\mathsf{rSIG},\mathcal{C}}(\lambda)$.

Proof. Any signature that is accepted by the verification algorithm must either reuse an existing tag, or sign a new tag. The success probability $\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SIG1},\mathcal{A}}(\lambda)$ of an attacker on SIG1 is bounded by the sum of the success probabilities $\mathrm{Adv}^{\mathsf{ot\text{-}nacma}}_{\mathsf{TOS},\mathcal{B}}(\lambda)$ of an attacker on TOS and the success probability $\mathrm{Adv}^{\mathsf{uf\text{-}rma}}_{\mathsf{rSIG},\mathcal{C}}(\lambda)$ of an attacker on rSIG.

Game 0: The actual Unforgeability game. $\Pr[\textbf{Game 0}] = \mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SIG1},\mathcal{A}}(\lambda)$.

Game 1: The real security game except that the winning condition is changed to no longer accept repetition of tags.

Lemma 18. $|\Pr[\textbf{Game 0}] - \Pr[\textbf{Game 1}]| \leq \mathrm{Adv}^{\mathsf{ot\text{-}nacma}}_{\mathsf{TOS},\mathcal{B}}(\lambda)$

Game 2: The fully idealized game. The winning condition is changed to reject all signatures.

Lemma 19. $|\Pr[\textbf{Game 1}] - Pr[\textbf{Game 2}]| \leq \mathrm{Adv}^{\mathsf{uf\text{-}rma}}_{\mathsf{rSIG},\mathcal{C}}(\lambda)$

Thus $\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SIG1},\mathcal{A}}(\lambda) = \Pr[\textbf{Game 0}] \leq \mathrm{Adv}^{\mathsf{ot\text{-}nacma}}_{\mathsf{TOS},\mathcal{B}}(\lambda) + \mathrm{Adv}^{\mathsf{uf\text{-}rma}}_{\mathsf{rSIG},\mathcal{C}}(\lambda)$ as claimed.

Theorem 20. *If* TOS.Tag *produces constant-size tags and signatures in the size of input messages, the resulting* SIG1 *produces constant-size signatures as well. Furthermore, if* TOS *and* rSIG *are structure-preserving, so is* SIG1.

We omit the proof of Theorem 20 as it is done simply by examining the construction.

4.2 SIG2: Combining partial one-time and XRMA-secure signatures

Let xSIG be a signature scheme with message space $\mathcal{M}_{\mathsf{x}}$, and POS be a partial one-time signature scheme with one-time public-key space $\mathcal{K}_{opk}$ such that $\mathcal{M}_{\mathsf{x}} = \mathcal{K}_{opk}$. We construct a signature scheme SIG2 from xSIG and POS. Let gk be a global parameter generated by $\mathsf{Setup}(1^\lambda)$.

- $\mathsf{SIG2.Key}(gk)$: Run $(pk_p, sk_p) \leftarrow \mathsf{POS.Key}(gk)$, $(vk_x, sk_x) \leftarrow \mathsf{xSIG.Key}(gk)$. Output $vk := (pk_p, vk_x)$ and $sk := (sk_p, sk_x)$.
- $\mathsf{SIG2.Sign}(sk, msg)$: Parse sk into (sk_p, sk_x). Run $(opk, osk) \leftarrow \mathsf{POS.Update}()$, $\sigma_p \leftarrow \mathsf{POS.Sign}(sk_p, msg, osk)$, $\sigma_x \leftarrow \mathsf{xSIG.Sign}(sk_x, opk)$. Output $\sigma := (opk, \sigma_p, \sigma_x)$.
- $\mathsf{SIG2.Vrf}(vk, \sigma, msg)$: Parse vk and σ accordingly. Output 1 if $1 = \mathsf{POS.Vrf}(pk_p, opk, \sigma_p, msg)$, and $1 = \mathsf{xSIG.Vrf}(vk_x, \sigma_x, opk)$. Output 0, otherwise.

Theorem 21. SIG2 *is UF-CMA if* POS *is uniform-key and OT-NACMA, and* xSIG *is UF-XRMA w.r.t.* POS.Update *as the message generator. In particular,* $\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SIG2},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathsf{ot\text{-}nacma}}_{\mathsf{POS},\mathcal{B}}(\lambda) + \mathrm{Adv}^{\mathsf{uf\text{-}xrma}}_{\mathsf{xSIG},\mathcal{C}}(\lambda)$.

Proof. The proof is almost the same as that for Theorem 17. The only difference appears in constructing $\mathcal{C}$ in the second step. Since POS.Update is used as the extended random message generator, the pair (msg, aux) is in fact (opk, osk). Given (opk, osk), adversary $\mathcal{C}$ can run $\mathsf{POS.Sign}(sk, msg, osk)$ to yield legitimate signatures.

Theorem 22. *If* POS *produces constant-size one-time public-keys and signatures in the size of input messages, resulting* SIG2 *produces constant-size signatures as well. Furthermore, if* POS *and* xSIG *are structure-preserving, so is* SIG2.

5 Instantiating SIG1

We instantiate the building blocks TOS and rSIG of our first generic construction to obtain our first SPS scheme. We do so in Type-I bilinear group setting. The resulting SIG1 scheme is an efficient structure-preserving signature scheme based only on the DLIN assumption.

Setup for Type-I groups. The following setup procedure is common for all instantiations in this section. The global parameter gk is given to all functions implicitly.

$\mathsf{Setup}(1^\lambda)$: Run $\Lambda = (p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$ and pick random generators $(G, C, F, U_1, U_2) \leftarrow \mathbb{G}^{*5}$. Output $gk := (\Lambda, G, C, F, U_1, U_2)$.

The parameters gk fix the message space $\mathcal{M}_{\mathsf{r}} := \{(C^{m_1}, C^{m_2}, F^{m_1}, F^{m_2}, U_1^{m_1}, U_2^{m_2}) \in \mathbb{G}^6 \mid (m_1, m_2) \in \mathbb{Z}_p^2\}$ for the RMA-secure signature scheme defined below. For our generic framework to work, the tagged one-time signature schemes should have the same tag space.

Tagged one-time signature scheme. Basically, a tag in our scheme consists of a pair of elements in $\mathbb{G}$. However, due to a constraint from rSIG we show in the next section, the tags will have to be in an extended form. We therefore parameterize the one-time key generation function Update with a flag *mode* $\in \{\mathsf{normal}, \mathsf{extended}\}$ so that it outputs a key in the original or extended form. Although *mode* is given to Update as input, it should be considered as a fixed system-wide parameter that is common for every invocation of Update and the key space is fixed throughout the use of the scheme. Accordingly, this extension does not affect the security model at all.

$\mathsf{TOS.Key}(gk)$: Parse $gk = (\Lambda, G, C, F, U_1, U_2)$. Pick random x_r, y_r, x_s, y_s, x_t, y_t, x_1, y_1, $\ldots$, x_k, y_k in $\mathbb{Z}_p$ such that such that $x_r y_s \neq x_s y_r$ and compute $G_r := G^{x_r}, H_r := G^{y_r}, G_s := G^{x_s}, H_s := G^{y_s}, G_t := G^{x_t}, H_t := G^{y_t}, G_0 := G^{x_0}, H_0 := G^{y_0}, \ldots, G_k := G^{x_k}, H_k := G^{y_k}$. Output $pk := (G_r, G_s, G_t, H_r, H_s, H_t, G_0, \ldots, G_k, H_0, \ldots, H_k)$ and $sk := (x_r, x_s, x_t, y_r, y_s, y_t, x_0, \ldots, x_k, y_0, \ldots, y_k)$

$\mathsf{TOS.Tag}()$: Take generators G, C, F, U_1, U_2 from gk. Choose $w_1, w_2 \leftarrow \mathbb{Z}_p^*$ and compute $tag := (C^{w_1}, C^{w_2}, F^{w_1}, F^{w_2}, U_1^{w_1}, U_2^{w_2})$. Output tag.

$\mathsf{TOS.Sign}(sk, msg, tag)$: Parse msg to $(M_1, \ldots, M_k)$ and tag to $(T_1, T_2, \ldots)$. Parse sk accordingly. Choose random $m \leftarrow \mathbb{Z}_p$ and let value $M_0 := G^m \prod_{i=1}^k M_i^{-1}$. (This is uniformly distributed.) Compute $A := G^{-x_t} T_1^{-m} \prod_{i=0}^k M_i^{-x_i}$ and $B := G^{-y_t} T_2^{-m} \prod_{i=0}^k M_i^{-y_i}$. Since $x_r y_s \neq x_s y_r$ we can compute $\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = \begin{pmatrix} x_r & x_s \\ y_r & y_s \end{pmatrix}^{-1}$. (The determinant is nonzero.) Compute $Z := A^\alpha B^\beta$ and $W := A^\gamma B^\delta$. Output $\sigma := (Z, W, M_0)$.

$\mathsf{TOS.Vrf}(pk, tag, msg, \sigma)$: Accept if the following equalities hold:

$$e(G_r, Z) \cdot e(G_s, W) \cdot e(G_t, G) \prod_{i=0}^{k} e(G_i T_1, M_i) = 1$$
$$e(H_r, Z) \cdot e(H_s, W) \cdot e(H_t, G) \prod_{i=0}^{k} e(H_i T_2, M_i) = 1$$

We remark that the correctness of the extended tag $(T_3, \ldots, T_6)$ is not examined within this scheme. (We only need to show that the extended part is simulatable in the security proof.) Since the tag is given to SIGr as a message, it is the verification function of SIGr that verifies the correctness with respect to its message space, which is the same as the tag space. The scheme is obviously structure-preserving and the correctness is easily verified by simple calculation.

Theorem 23. *The above* TOS *scheme is OT-CMA under the SDP. In particular, for any* $\mathcal{A}$ *that makes at most* q_s *signing queries,* $\mathrm{Adv}^{\mathsf{ot\text{-}cma}}_{\mathsf{TOS},\mathcal{A}}(\lambda) \leq q_s \cdot \mathrm{Adv}^{\mathsf{sdp}}_{\mathcal{G},\mathcal{B}}(\lambda) + 1/p$ *holds.*

Proof. We show a reduction algorithm that simulates the one-time adaptive chosen message attack game for the adversary. The reduction gets an instance of the simultaneous double pairing assumption, $\Lambda, G_r, G_s, H_r, H_s$, and proceeds as follows.

Setup and Key Generation. It chooses ξ, η, μ and sets $G_t := G_r^\xi G_s^\eta$, and $H_t := H_r^\xi H_s^\mu$. It chooses $G \in \mathbb{G}$ and random $\omega, \nu, \nu_1, \nu_2$, and computes $gk = (\Lambda, C, F, U_1, U_2) = (\Lambda, G^\omega, G^{\omega\nu}, G^{\omega\nu_1}, G^{\omega\nu_2})$. It chooses random ρ_i, σ_i, τ_i, computes $G_i = G_r^{\rho_i} G_s^{\sigma_i} G_t^{\tau_i} = G_r^{\rho_i + \xi\tau_i} G_s^{\sigma_i + \eta\tau_i}$ and $H_i = H_r^{\rho_i} H_s^{\sigma_i} H_t^{\tau_i} = H_r^{\rho_i + \xi\tau_i} H_s^{\sigma_i + \mu\tau_i}$ for $i = 0 \ldots k$, and sets $pk = (G, G_r, G_s, G_t, H_r, H_s, H_t, G_0, \ldots G_k, H_0, \ldots, H_k)$. (Note that G_i, H_i are correctly distributed and give no information about τ_i.) It sends pk, gk to the adversary. The reduction will pick a random session j^*, and assume that the adversary will try to reuse tag from that session.

Queries to oracle $\mathcal{O}t$. When the adversary makes a query to the tag oracle $\mathcal{O}t$, choose the next new session index j.

- For session $j \neq j^*$: Pick random values $\rho, \sigma, \tau \leftarrow \mathbb{Z}_p$. Compute $(T_1, T_2) = (G_r^{\rho} G_s^{\sigma} G_t^{\tau}, H_r^{\rho} H_s^{\sigma} H_t^{\tau}) = (G_r^{\rho+\xi\tau} G_s^{\sigma+\eta\tau}, H_r^{\rho+\xi\tau} H_s^{\sigma+\mu\tau})$, and set $T = (T_1, T_2, T_1^{\nu}, T_2^{\nu}, T_1^{\nu_1}, T_2^{\nu_2})$. Store (j, ρ, σ, τ), and return T to the adversary.
- For session j^*. Pick random values $\rho, \sigma \leftarrow \mathbb{Z}_p$. Compute $(T_1, T_2) = (G_r^{\rho} G_s^{\sigma}, H_r^{\rho} H_s^{\sigma})$. Let $T = (T_1, T_2, T_1^{\nu}, T_2^{\nu}, T_1^{\nu_1}, T_2^{\nu_2})$. Store (j^*, ρ, σ), and return T to the adversary.

Queries to oracle $\mathcal{O}sig$. When the adversary queries $\mathcal{O}sig$ for message $M = (M_1, \ldots, M_k) \in G^k$ and session j, proceed as follows.

- If the $\mathcal{O}t$ has not yet produced a tag for session j, or $\mathcal{O}sig$ has already been queried for session j, return $\perp$.
- For session $j \neq j^*$: Look up the stored tuple (j, ρ, σ, τ). Compute $M_0 = (G \prod_{i=1}^{k} M_i^{\tau+\tau_i})^{-\frac{1}{\tau_0+\tau}}$. Note that for this choice of M_0, it will be the case that $e(G_t, G) \prod_{i=0}^{k} e(G_t^{\tau_i+\tau}, M_i) = e(G_t, M_0^{\tau_0+\tau} G \prod_{i=1}^{k} M_i^{\tau_i+\tau}) = 1$ and similarly $e(H_t, G) \prod_{i=0}^{k} e(H_t^{\tau_i+\tau}, M_i) = e(H_t, M_0^{\tau_0+\tau} G \prod_{i=1}^{k} M_i^{\tau_i+\tau}) = 1$. Note also that the tag is independent of τ, and since τ is uniformly distributed, then M_0 is independent of $\tau_0, \ldots, \tau_k$ even given tag. (To see this, let $m_0, \ldots, m_k$ be the discrete logarithms of $M_0, \ldots, M_k$ respectively and note that for any choice of $m_1, \ldots, m_k, \tau_0, \ldots, \tau_k$ and for any m_0 such that $m_0 \neq -\sum_{i=1}^{k} m_i$, there is a $\frac{1}{q}$ chance that we will choose $\tau = \frac{-1-\sum_{i=0}^{k} m_i \tau_i}{\sum_{i=0}^{k} m_i}$ which will yield $M_0 = (G \prod_{i=1}^{k} M_i^{\tau_i+\tau})^{-\frac{1}{\tau_0+\tau}}$.) Now compute $Z = \prod_{i=0}^{k} M_i^{-\rho_i-\rho}$ and $W = \prod_{i=0}^{k} M_i^{-\sigma_i-\sigma}$ and output the signature (Z, W, M_0).
  Note that these are the unique values such that $e(G_r, Z) \cdot e(G_s, W) \cdot e(G_t, G) \prod_{i=0}^{k} e(G_i T_1, M_i) = 1$ and similarly $e(H_r, Z) \cdot e(H_s, W) \cdot e(H_t, G) \prod_{i=0}^{k} e(H_i T_2, M_i) = 1$. Thus, Z, W are uniquely determined by $M_0, M_1, \ldots, M_k, tag$, and pk. $M_1, \ldots, M_k$ are provided by the adversary and, as we have argued, M_0, tag, pk are statistically independent of $\tau_0, \ldots, \tau_k$. We conclude that Z, W reveal no additional information about $\tau_0, \ldots, \tau_k$ even given the rest of the adversary's view.
- For session j^*: Look up the stored tuple (j, ρ, σ). Let $M_0 = (G \prod_{i=1}^{k} M_i^{\tau_i})^{-\frac{1}{\tau_0}})$. Note that for this choice of M_0, it will be the case that $e(G_t, G) \prod_{i=0}^{k} e(G_t^{\tau_i}, M_i) = e(G_t, M_0^{\tau_0} G \prod_{i=1}^{k} M_i^{\tau_i}) = 1$ and similarly $e(H_t, G) \prod_{i=0}^{k} e(H_t^{\tau_i}, M_i) = e(H_t, M_0^{\tau_0} G \prod_{i=1}^{k} M_i^{\tau_i}) = 1$. Note that T_1, T_2 are correctly distributed, that M_0 is statistically close to uniform since $\tau_0, \ldots, \tau_k$ are chosen at random, and furthermore that the only information revealed about $\tau_0, \ldots, \tau_k$ is that $G \prod_{i=0}^{k} M_i^{\tau_i} = 1$. Now, compute $Z = \prod_{i=0}^{k} M_i^{-\rho_i-\rho}$ and $W = \prod_{i=0}^{k} M_i^{-\sigma_i-\sigma}$, and output the signature (Z, W, M_0). Again all values are independent of $\tau_0, \ldots, \tau_k$ with the exception now of M_0, which is chosen so $G \prod_{i=0}^{k} M_i^{\tau_i} = 1$.

Processing the adversary's forgery. Now, suppose that the adversary produces $(M_1^{\dagger}, \ldots M_k^{\dagger})$ and $(Z^{\dagger}, W^{\dagger}, M_0^{\dagger}, T)$ for $T = (T_1, T_2, \ldots)$ used in the j^*th query. Look up the stored tuple (j^*, ρ, σ). Then with non-negligible probability (whenever the adversary succeeds) we have $\mathsf{TOS.Vrf}(pk, T, (M_1^{\dagger}, \ldots, M_k^{\dagger}), (Z^{\dagger}, W^{\dagger}, M_0^{\dagger})) = 1$. This means

$$1 = e(G_r, Z^{\dagger} G^{\xi} \prod_{i=0}^{k} (M_i^{\dagger})^{\rho_i+\rho+\xi\tau_i}) e(G_s, W^{\dagger} G^{\eta} \prod_{i=0}^{k} (M_i^{\dagger})^{\sigma_i+\sigma+\eta\tau_i}), \text{ and}$$

$$1 = e(H_r, Z^{\dagger} G^{\xi} \prod_{i=0}^{k} (M_i^{\dagger})^{\rho_i+\rho+\xi\tau_i}) e(H_s, W^{\dagger} G^{\mu} \prod_{i=0}^{k} (M_i^{\dagger})^{\sigma_i+\sigma+\mu\tau_i}).$$

So if $Z^\dagger G^\xi \prod_{i=0}^k (M_i^\dagger)^{\rho_i+\rho+\xi\tau_i} \neq 1$, then

$$(Z^\star,\ R^\star,\ S^\star) := (Z^\dagger G^\xi \prod_{i=0}^k (M_i^\dagger)^{\rho_i+\rho+\xi\tau_i}, W^\dagger G^\eta \prod_{i=0}^k (M_i^\dagger)^{\sigma_i+\sigma+\eta\tau_i}, W^\dagger G^\mu \prod_{i=0}^k (M_i^\dagger)^{\sigma_i+\sigma+\mu\tau_i})$$

is a valid solution for the simultaneous double pairing assumption.

$Z^\dagger G^\xi \prod_{i=0}^k (M_i^\dagger)^{\rho_i+\rho+\xi\tau_i} = Z^\dagger \prod_{i=0}^k (M_i^\dagger)^{\rho_i+\rho} (G \prod_{i=0}^k (M_i^\dagger)^{\tau_i})^\xi$, and a part of $Z^\dagger \prod_{i=0}^k (M_i^\dagger)^{\rho_i+\rho}$ is information theoretically hiding. Note that the only information that the adversary has about $\tau_0, \ldots, \tau_1$ is that in the j^*th session M_0 was chosen so that $G \prod_{i=0}^k M_i^{\tau_i} = 1$ (where $M = (M_1, \ldots, M_k)$ is the message signed in the j^*th session). If $M_i^\dagger \neq M_i$ for at least one i, then the probability that $G \prod_{i=0}^k (M_i^\dagger)^{\tau_i} = 1$ conditioned on the fact that $G \prod_{i=0}^k M_i^{\tau_i} = 1$ is $1/p$. As a result, the probability that $Z^\dagger G^\xi \prod_{i=0}^k (M_i^\dagger)^{\rho_i+\rho+\xi\tau_i} = 1$ is $1/p$.

Thus, if the guess for j^* is right, we succeed with all but probability $1/p$ whenever $\mathcal{A}$ does. We therefore have $\mathrm{Adv}^{\mathsf{ot\text{-}cma}}_{\mathsf{TOS},\mathcal{A}}(\lambda) \leq q_s \cdot \mathrm{Adv}^{\mathsf{sdp}}_{\mathcal{G},\mathcal{B}}(\lambda) + 1/p$.

RMA-secure signature scheme. For our random message signature scheme we will use a construction based on the dual system signature proposed in [41]. While the original scheme is CMA-secure under the DLIN assumption, the security proof makes use of a trapdoor commitment to elements in $\mathbb{Z}_p$ and consequently messages are elements in $\mathbb{Z}_p$ rather than $\mathbb{G}$. Our construction below resorts to RMA-security and removes this commitment to allows messages to be a sequence of random group elements satisfying a particular relation. As mentioned above, the message space $\mathcal{M}_\mathsf{x} := \{(C^{m_1}, C^{m_2}, F^{m_1}, F^{m_2}, U_1^{m_1}, U_2^{m_2}) \in \mathbb{G}^6 \mid (m_1, m_2) \in \mathbb{Z}_p^2\}$ is defined by generators (C, F, U_1, U_2) in gk.

$\mathsf{rSIG.Key}(gk)$: Given $gk := (\Lambda, G, C, F, U_1, U_2)$ as input, uniformly select V, V_1, V_2, H from $\mathbb{G}^*$ and a_1, a_2, b, α, and ρ from $\mathbb{Z}_p^*$. Then compute and output $vk := (B, A_1, A_2, B_1, B_2, R_1, R_2, W_1, W_2, V, V_1, V_2, H, X_1, X_2)$ and $sk := (vk, K_1, K_2)$ where

$$\begin{array}{llllll} B := G^b, & A_1 := G^{a_1}, & A_2 := G^{a_2}, & B_1 := G^{b\cdot a_1}, & B_2 := G^{b\cdot a_2} \\ R_1 := VV_1^{a_1}, & R_2 := VV_2^{a_2}, & W_1 := R_1^b, & W_2 := R_2^b, & \\ X_1 := G^\rho, & X_2 := G^{\alpha\cdot a_1\cdot b/\rho}, & K_1 := G^\alpha, & K_2 := G^{\alpha\cdot a_1}\ . & \end{array}$$

$\mathsf{rSIG.Sign}(sk, msg)$: Parse msg into $(M_1, M_2, M_3, M_4, M_5, M_6)$. Pick random $r_1, r_2, z_1, z_2 \in \mathbb{Z}_p$. Let $r = r_1 + r_2$. Compute and output signature $\sigma := (S_0, S_1, \ldots S_7)$ where

$$\begin{array}{llll} S_0 := (M_5M_6H)^{r_1}, & S_1 := K_2V^r, & S_2 := K_1^{-1}V_1^rG^{z_1}, & S_3 := B^{-z_1}, \\ S_4 := V_2^rG^{z_2}, & S_5 := B^{-z_2}, & S_6 := B^{r_2}, & S_7 := G^{r_1}. \end{array}$$

$\mathsf{rSIG.Vrf}(vk, \sigma, msg)$: Parse msg into $(M_1, M_2, M_3, M_4, M_5, M_6)$ and σ into $(S_0, S_1, \ldots, S_7)$. Also parse vk accordingly. Verify the following pairing product equations:

$$\begin{array}{l} e(S_7, M_5M_6H) = e(G, S_0) \\ e(S_1, B)\, e(S_2, B_1)\, e(S_3, A_1) = e(S_6, R_1)\, e(S_7, W_1) \\ e(S_1, B)\, e(S_4, B_2)\, e(S_5, A_2) = e(S_6, R_2)\, e(S_7, W_2)\, e(X_1, X_2) \\ e(F, M_1) = e(C, M_3),\ \ e(F, M_2) = e(C, M_4),\ \ e(U_1, M_1) = e(C, M_5),\ \ e(U_2, M_2) = e(C, M_6) \end{array}$$

The scheme is structure-preserving by construction and the correctness is easily verified.

Theorem 24. *The above* rSIG *scheme is UF-RMA under the DLIN assumption. In particular, for any p.p.t. adversary $\mathcal{A}$ against* rSIG *that makes at most q_s signing queries, there exists p.p.t. algorithm $\mathcal{B}$ for DLIN such that* $\mathrm{Adv}^{\mathsf{uf\text{-}rma}}_{\mathsf{rSIG},\mathcal{A}}(\lambda) \leq (q_s+2)\cdot \mathrm{Adv}^{\mathsf{dlin}}_{\mathcal{G},\mathcal{B}}(\lambda)$.

Proof. We refer to the signatures output by the signing algorithm as a *normal signature*. In the proof we will consider an additional type of signatures to which we refer to as *simulation-type signatures* that are computationally indistinguishable but easier to simulate. For $\gamma \in \mathbb{Z}_p$, simulation-type signatures are of the form $\sigma = (S_0, S_1' = S_1 \cdot G^{-a_1 a_2 \gamma}, S_2' = S_2 \cdot G^{a_2\gamma}, S_3, S_4' = S_4 \cdot G^{a_1\gamma}, S_5, \ldots, S_7)$. We give the outline of the proof using some lemmas.

Lemma 25. *Any signature that is accepted by the verification algorithm must be formed either as a normal signature, or a simulation-type signature.*

We consider a sequence of games. Let p_i be the probability that the adversary succeeds in **Game i**, and $p_i^{\mathrm{norm}}(\lambda)$ and $p_i^{\mathrm{sim}}(\lambda)$ that he succeeds with a normal-type respectively simulation-type forgery. Then by Lemma 25, $p_i(\lambda) = p_i^{\mathrm{norm}}(\lambda) + p_i^{\mathrm{sim}}(\lambda)$ for all i.

Game 0: The actual Unforgeability under Random Message Attacks game.

Lemma 26. *There exists an adversary $\mathcal{B}_1$ such that $p_0^{sim}(\lambda) = \mathrm{Adv}^{\mathsf{dlin}}_{\mathcal{G},\mathcal{B}_1}(\lambda)$.*

Game i: The real security game except that the first i signatures that are given by the oracle are simulation-type signatures.

Lemma 27. *There exists an adversary $\mathcal{B}_2$ such that $|p_{i-1}^{norm}(\lambda) - p_i^{norm}(\lambda)| = \mathrm{Adv}^{\mathsf{dlin}}_{\mathcal{G},\mathcal{B}_2}(\lambda)$.*

Game q: All sigantures that given by the oracle are simulation-type signatures.

Lemma 28. *There exists an adversary $\mathcal{B}_3$ such that $p_q^{norm}(\lambda) = \mathrm{Adv}^{\mathsf{cdh}}_{\mathcal{G},\mathcal{B}_3}(\lambda)$.*

We have shown that in **Game q**, $\mathcal{A}$ can output a normal-type forgery with at most negligible probability. Thus, by Lemma 27 we can conclude that the same is true in **Game 0** and it holds

$$\begin{aligned}\mathrm{Adv}^{\mathsf{uf\text{-}rma}}_{\mathsf{rSIG},\mathcal{A}}(\lambda) = p_0(\lambda) = p_0^{\mathrm{sim}}(\lambda) + p_0^{\mathrm{norm}}(\lambda) &\leq p_0^{\mathrm{sim}}(\lambda) + \sum_{i=1}^{q} |p_{i-1}^{\mathrm{norm}}(\lambda) - p_i^{\mathrm{norm}}(\lambda)| + p_q^{\mathrm{norm}}(\lambda) \\ &\leq \mathrm{Adv}^{\mathsf{dlin}}_{\mathcal{G},\mathcal{B}_1}(\lambda) + q\mathrm{Adv}^{\mathsf{dlin}}_{\mathcal{G},\mathcal{B}_2}(\lambda) + \mathrm{Adv}^{\mathsf{cdh}}_{\mathcal{G},\mathcal{B}_3}(\lambda) \leq (q+2)\cdot \mathrm{Adv}^{\mathsf{dlin}}_{\mathcal{G},\mathcal{B}}(\lambda)\ .\end{aligned}$$

Let MSGGen be an extended random message generator that first chooses $aux = (m_1, m_2)$ randomly from $\mathbb{Z}_p^2$ and then computes $msg = (C^{m_1}, C^{m_2}, F^{m_1}, F^{m_2}, U_1^{m_1}, U_2^{m_2})$. Note that this is what the reduction algorithm does in the proof of Theorem 24. Therefore, the same reduction algorithm works for the case of extended random message attacks with respect to message generator MSGGen. We thus have the following.

Corollary 29. *Under the DLIN assumption,* rSIG *scheme is UF-XRMA w.r.t. the message generator that provides $aux = (m_1, m_2)$ for every message $msg = (C^{m_1}, C^{m_2}, F^{m_1}, F^{m_2}, U_1^{m_1}, U_2^{m_2})$. In particular, for any p.p.t. adversary $\mathcal{A}$ against* rSIG *that is given at most q_s signatures, there exists p.p.t. algorithm $\mathcal{B}$ such that* $\mathrm{Adv}^{\mathsf{uf\text{-}xrma}}_{\mathsf{rSIG},\mathcal{A}}(\lambda) \leq (q_s+2)\cdot \mathrm{Adv}^{\mathsf{dlin}}_{\mathcal{G},\mathcal{B}}(\lambda)$.

Security and efficiency of resulting SIG1. Let SIG1 be the signature scheme obtained from TOS (with *mode* = extended) and rSIG by following the first generic construction in Section 4. From Theorem 17, 20, 23, and 24, the following is immediate.

Theorem 30. SIG1 *is a structure-preserving signature scheme that yields constant-size signatures, and is UF-CMA under the DLIN assumption. In particular, for any p.p.t. adversary $\mathcal{A}$ for* SIG1 *making at most q_s signing queries, there exists p.p.t. algorithm $\mathcal{B}$ such that* $\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SIG1},\mathcal{A}}(\lambda) \leq (q_s+3)\cdot \mathrm{Adv}^{\mathsf{dlin}}_{\mathcal{G},\mathcal{B}}(\lambda) + 1/p$.

6 Instantiating SIG2

We instantiate the POS and xSIG building blocks of our second generic construction to obtain our second SPS scheme. Here we choose the Type-III bilinear group setting. The resulting SIG2 scheme is an efficient structure-preserving signature scheme based on SXDH and XDLIN.

Setup for Type-III groups. The following setup procedure is common for all building blocks in this section. The global parameter gk is given to all functions implicitly.

- $\mathsf{Setup}(1^\lambda)$: Run $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$ and choose generators $G \in \mathbb{G}_1^*$ and $\hat{G} \in \mathbb{G}_2^*$. Also choose u, f_2, f_3 randomly from $\mathbb{Z}_p^*$, compute $F_2 := G^{f_2}$, $F_3 := G^{f_3}$, $\hat{F}_2 := \hat{G}^{f_2}$, $\hat{F}_3 := \hat{G}^{f_3}$, $U := G^u$, $\hat{U} := \hat{G}^u$, and output $gk := (\Lambda, G, \hat{G}, F_2, F_3, \hat{F}_2, \hat{F}_3, U, \hat{U})$.

A gk defines a message space $\mathcal{M}_{\mathsf{x}} = \{(\hat{F}_2^m, \hat{F}_3^m, \hat{U}^m) \in \mathbb{G}_2^* \mid m \in \mathbb{Z}_p\}$ for the signature scheme in this section. For our generic construction to work, the partial one-time signature scheme should have the same key space.

Partial one-time signatures for uniliteral messages. We construct a partial one-time signature scheme $\mathsf{POS}u2$ for messages in $\mathbb{G}_2^k$ for $k > 0$. The suffix "u2" indicates that the scheme is uniliteral and messages are taken from $\mathbb{G}_2$. Correspondingly, $\mathsf{POS}u1$ refers to the scheme whose messages belong to $\mathbb{G}_1$, which is obtained by swapping $\mathbb{G}_2$ and $\mathbb{G}_1$ in the following description. Our $\mathsf{POS}u2$ scheme is a minor refinement of the one-time signature scheme introduced in [4]. It comes, however, with a security proof for the new security model.

Basically, a one-time public-key in our scheme consists of one element in the base group $\mathbb{G}_1$ that is the opposite of the group $\mathbb{G}_2$ messages belong to. This property is very useful to construct a POS scheme for signing bilateral messages. As well as tags of TOS in Section 5, the one-time public-keys of POS will have to be in an extended form to meet the constraint from xSIG presented in the sequel. We use *mode* $\in \{\mathsf{normal}, \mathsf{extended}\}$ for this purpose again.

- $\mathsf{POS}u2.\mathsf{Key}(gk)$: Take generators U and $\hat{U}$ from gk. Choose w_r randomly from $\mathbb{Z}_p^*$ and compute $G_r := U^{w_r}$. For $i = 1, \ldots, k$, uniformly choose χ_i and γ_i from $\mathbb{Z}_p$ and compute $G_i := U^{\chi_i} G_r^{\gamma_i}$. Output $pk := (G_r, G_1, ..., G_k) \in \mathbb{G}_1^{k+1}$ and $sk := (\chi_1, \gamma_1, ..., \chi_k, \gamma_k, w_r)$.
- $\mathsf{POS}u2.\mathsf{Update}(\mathit{mode})$: Take F_2, F_3, U from gk. Choose $a \leftarrow \mathbb{Z}_p$ and output $opk := U^a \in \mathbb{G}_1$ if *mode* $=$ normal or $opk := (F_2^a, F_3^a, U^a) \in \mathbb{G}_1^3$ if *mode* $=$ extended. Also output $osk := a$.
- $\mathsf{POS}u2.\mathsf{Sign}(sk, msg, osk)$: Parse msg into $(\hat{M}_1, \ldots, \hat{M}_k) \in \mathbb{G}_2^k$. Take a and w_r from osk and sk, respectively. Choose ρ randomly from $\mathbb{Z}_p$ and compute $\zeta := a - \rho\, w_r \bmod p$. Then compute and output $\sigma := (\hat{Z}, \hat{R}) \in \mathbb{G}_2^2$ as the signature, where $\hat{Z} := \hat{U}^\zeta \prod_{i=1}^k \hat{M}_i^{-\chi_i}$ and $\hat{R} := \hat{U}^\rho \prod_{i=1}^k \hat{M}_i^{-\gamma_i}$
- $\mathsf{POS}u2.\mathsf{Vrf}(pk, \sigma, msg, opk)$: Parse σ as $(\hat{Z}, \hat{R}) \in \mathbb{G}_2^2$, msg as $(\hat{M}_1, \ldots, \hat{M}_k) \in \mathbb{G}_2^k$, and opk as (A_2, A_3, A) or A depending on *mode*. Return 1, if $e(A, \hat{U}) = e(U, \hat{Z})\, e(G_r, \hat{R})$ $\prod_{i=1}^k e(G_i, \hat{M}_i)$ holds. Return 0, otherwise.

Scheme $\mathsf{POS}u2$ is structure-preserving and has uniform one-time public-key property from the construction. We can easily verify that it is correct by simple calculation.

Theorem 31. $\mathsf{POS}u2$ *is strongly unforgeable against OT-CMA if DBP*$_1$ *holds. In particular,* $\mathrm{Adv}^{\mathsf{sot\text{-}cma}}_{\mathsf{POS}u2,\mathcal{A}}(\lambda) \le \mathrm{Adv}^{\mathsf{dbp1}}_{\mathcal{G},\mathcal{B}}(\lambda) + 1/p$.

Partial one-time signatures for bilateral messages. Using $\mathsf{POS}u1$ for $msg \in \mathbb{G}_1^{k_1+1}$ and $\mathsf{POS}u2$ for $msg \in \mathbb{G}_2^{k_2}$, we construct a $\mathsf{POS}b$ scheme for signing bilateral messages $(msg_1, msg_2) \in \mathbb{G}_1^{k_1} \times \mathbb{G}_2^{k_2}$. The scheme is a simple two-story construction where msg_2 is signed by $\mathsf{POS}u2$ with one-time secret-key $osk_2 \in \mathbb{G}_1$ and then the one-time public-key opk_2 is attached to msg_1 and signed by $\mathsf{POS}u1$. Public-key opk_2 is included in the signature, and opk_1 is output as a one-time public-key for $\mathsf{POS}b$.

- $\mathsf{POS}b.\mathsf{Key}(gk)$: Run $(pk_1, sk_1) \leftarrow \mathsf{POS}u1.\mathsf{Key}(gk)$ and $(pk_2, sk_2) \leftarrow \mathsf{POS}u2.\mathsf{Key}(gk)$. Set $pk := (pk_1, pk_2)$ and $sk := (sk_1, sk_2)$, and output (pk, sk).
- $\mathsf{POS}b.\mathsf{Update}(\mathit{mode})$: Run $(opk, osk) \leftarrow \mathsf{POS}u1(\mathit{mode})$ and output (opk, osk).
- $\mathsf{POS}b.\mathsf{Sign}(sk, msg, osk)$: Parse msg into $(msg_1, msg_2) \in \mathbb{G}_1^{k_1} \times \mathbb{G}_2^{k_2}$, and sk into (sk_1, sk_2). Run $(opk_2, osk_2) \leftarrow \mathsf{POS}u2.\mathsf{Update}(\mathsf{normal})$, and compute $\sigma_2 \leftarrow \mathsf{POS}u2.\mathsf{Sign}(sk_2, msg_2, osk_2)$ and $\sigma_1 \leftarrow \mathsf{POS}u1.\mathsf{Sign}(sk_1, (msg_1, opk_2), osk)$. Output $\sigma := (\sigma_1, \sigma_2, opk_2)$.
- $\mathsf{POS}b.\mathsf{Vrf}(pk, opk, \sigma, msg)$: Parse msg into $(msg_1, msg_2) \in \mathbb{G}_1^{k_1} \times \mathbb{G}_2^{k_2}$, and σ into $(\sigma_1, \sigma_2, opk_2)$. If $1 = \mathsf{POS}u1.\mathsf{Vrf}(pk_1, opk, \sigma_1, (msg_1, opk_2)) = \mathsf{POS}u2.\mathsf{Vrf}(pk_2, opk_2, \sigma_2, msg_2)$, output 1. Otherwise, output 0.

For a message in $\mathbb{G}_1^{k_1} \times \mathbb{G}_2^{k_2}$, the above $\mathsf{POS}b$ uses a public-key of size $(k+2, k+1)$, yields a one-time public-key of size $(0, 1)$ (for *mode* $=$ normal) or $(0, 3)$ (for *mode* $=$ $\mathsf{extended}$), and a signature of size $(3, 2)$. Verification requires 2 pairing product equations. A one-time public-key in $\mathsf{extended}$ mode, which is treated as a message to xSIG in this section, is of the form $opk = (\hat{F}_2^a, \hat{F}_3^a, \hat{U}^a) \in \mathbb{G}_2^3$. Structure-preservance and uniform public-key property are taken over from the underlying $\mathsf{POS}u1$ and $\mathsf{POS}u2$.

Theorem 32. *Scheme* $\mathsf{POS}b$ *is unforgeable against OT-CMA if SXDH holds. In particular,* $\mathrm{Adv}^{\mathsf{ot\text{-}cma}}_{\mathsf{POS}b,\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathsf{sxdh}}_{\mathcal{G},\mathcal{B}}(\lambda)$*+2/p.*

XRMA-secure signature scheme. Our construction bases on a variant of Waters' dual system encryption proposed by Ramanna, Chatterjee, and Sarkar [36]. Recall that $gk = (\Lambda, G, \hat{G}, F_2, F_3, \hat{F}_2, \hat{F}_3, U, \hat{U})$ with $\Lambda = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ is generated by $\mathsf{Setup}(1^\lambda)$ in advance.

$\mathsf{xSIG}.\mathsf{Gen}(gk)$**:** On input gk, select generators $V, V', H \leftarrow \mathbb{G}_1$, $\hat{V}, \hat{V}', \hat{H} \in \mathbb{G}_2$ such that $V \sim \hat{V}, V' \sim \hat{V}', H \sim \hat{H}, F_2 \sim \hat{F}_2, F_3 \sim \hat{F}_3$ and exponent $a, b, \alpha \leftarrow \mathbb{Z}_p$ and $\rho \leftarrow \mathbb{Z}_p^*$, compute $R := V(V')^a, \hat{R} := \hat{V}(\hat{V}')^a$, and set $vk := (gk, \hat{G}^b, \hat{G}^a, \hat{G}^{ba}, \hat{R}, \hat{R}^b$, $sk := (VK, G^\alpha, G^a, G^b)$.

$\mathsf{xSIG}.\mathsf{Sign}(sk, msg)$**:** On input message $msg = (\hat{M}_1, \hat{M}_2, \hat{M}_0) = (\hat{F}_2^m, \hat{F}_3^m, \hat{U}^m) \in \mathbb{G}_2^3$ ($m \in \mathbb{Z}_p$), select $r_1, r_2 \leftarrow \mathbb{Z}_p$, set $r := r_1 + r_2$, compute $\sigma_0 := (\hat{M}_0\hat{H})^{r_1}$, $\sigma_1 := G^\alpha V^r$, $\sigma_2 := (V')^r G^{-z}$, $\sigma_3 := (G^b)^z$, $\sigma_4 := (G^b)^{r_2}$, and $\sigma_5 := G^{r_1}$, and output $\sigma := (\sigma_0, \sigma_1, \ldots, \sigma_5) \in \mathbb{G}_2 \times \mathbb{G}_1^5$.

$\mathsf{xSIG}.\mathsf{Vrfy}(vk, \sigma, msg)$**:** On input $vk, msg = (\hat{M}_1, \hat{M}_2, \hat{M}_0)$, and signature σ, compute

$$e(F_2, \hat{M}_0) = e(U, \hat{M}_1),\ e(F_3, \hat{M}_0) = e(U, \hat{M}_2),\ e(\sigma_5, \hat{M}_0\hat{H}) = e(G, \sigma_0)$$
$$e(\sigma_1, \hat{G}^b)e(\sigma_2, \hat{G}^{ba})e(\sigma_3, \hat{G}^a) = e(\sigma_4, \hat{R})e(\sigma_5, \hat{R}^b)e(G^\rho, \hat{G}^{\alpha b/\rho}).$$

The scheme is structure-preserving by the construction. We can easily verify the correctness.

Theorem 33. *If the* DDH_2 *and* $XDLIN_1$ *assumptions hold, then above* xSIG *scheme is UF-XRMA with respect to the message generator that returns* $aux = m$ *for every random message* $msg = (\hat{F}_2^m, \hat{F}_3^m, \hat{U}^m)$*. In particular for any p.p.t. adversary* $\mathcal{A}$ *for* xSIG *making at most* q *signing queries, there exist p.p.t. algorithms* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ *such that* $\mathrm{Adv}^{\mathsf{uf\text{-}xrma}}_{\mathsf{xSIG},\mathcal{A}}(\lambda) < \mathrm{Adv}^{\mathsf{ddh2}}_{\mathcal{G},\mathcal{B}_1}(\lambda) + q\mathrm{Adv}^{\mathsf{xdlin1}}_{\mathcal{G},\mathcal{B}_2}(\lambda) + \mathrm{Adv}^{\mathsf{co\text{-}cdh}}_{\mathcal{G},\mathcal{B}_3}(\lambda)$.

Table 1. Efficiency of our schemes (SIG1 and SIG2) and comparison to other schemes with constant-size signatures. The top section is for the Type I variant, the middle section is for unilateral messages and the lower section is for bilateral messages. Notation (x, y) represents x elements in $\mathbb{G}_1$ and y in $\mathbb{G}_2$.

Schemes	$\|msg\|$	$\|gk\| + \|vk\|$	$\|\sigma\|$	#(PPE)	Assumptions
AHO10	k	$2k + 12$	7	2	q-SFP
SIG1	k	$2k + 25$	17	9	DLIN
AHO10	$(k_1, 0)$	$(4, 2k_1 + 8)$	$(5, 2)$	2	q-SFP
AGHO11	$(k_1, 0)$	$(1, k_1 + 4)$	$(3, 1)$	2	q-type
SIG2 : POSu1 + xSIG	$(k_1, 0)$	$(7, k_1 + 13)$	$(7, 4)$	5	SXDH, XDLIN$_1$
POSb + AHO10	(k_1, k_2)	$(k_2 + 5, k_1 + 12)$	$(10, 3)$	3	q-SFP
AGHO11	(k_1, k_2)	$(k_2 + 3, k_1 + 4)$	$(3, 3)$	2	q-type
SIG2 : POSb + xSIG	(k_1, k_2)	$(k_2 + 8, k_1 + 14)$	$(8, 6)$	6	SXDH, XDLIN$_1$

Security and efficiency of resulting SIG2. Let SIG2 be the scheme obtained from POSb (with *mode* = extended) and xSIG. SIG2 is structure-preserving as vk, σ, and msg consist of group elements from $\mathbb{G}_1$ and $\mathbb{G}_2$, and SIG2.Vrf evaluates pairing product equations. From Theorem 21, 32, and 33, we obtain the following theorem.

Theorem 34. SIG2 *is a structure-preserving signature scheme that is unforgeable against adaptive chosen message attacks if SXDH and XDLIN*$_1$ *hold for* $\mathcal{G}$.

7 Efficiency, Applications & Open Questions

Efficiency. Table 1 summarizes the efficiency of SIG1 and SIG2. For SIG2 we consider both uniliteral and biliteral messages. We count the number of group elements excluding a default generator for each group in gk, and distinguish between $\mathbb{G}_1$ and $\mathbb{G}_2$ and use k_1 and k_2 for the number of message elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. For comparison, we include the efficiency of the schemes in [4] and [2]. For bilateral messages, AHO10 is combined with POSb from Section 6.

Applications. Structure-preserving signatures (SPS) have become a mainstay in cryptographic protocol design in recent years. From the many applications that benefit from efficient SPS based on simple assumptions, we list only a few recent examples. Using our SIG1 scheme from Section 5 both the construction of a group signature scheme with efficient revocation by Libert, Peters and Yung [33] and the construction of compact verifiable shuffles by Chase et al. [16] can be proven purely under the DLIN assumption. All other building blocks already have efficient instantiations based on DLIN.

Hofheinz and Jager [31] construct a structure-preserving one-time signature scheme and use it to build a tree-based SPS scheme, say tSIG. Instead, we propose to use our partial one-time scheme to construct tSIG. As the resulting tSIG is secure against non-adaptive chosen message attacks, it is secure against extended random message attacks as well. We then combine the POSb scheme and the new tSIG scheme according to our second generic construction. As confirmed with the authors of [31], the resulting signature scheme is significantly more efficient than [31] and is a SPS scheme with a tight security reduction to SXDH. One can do the same in Type-I groups by using the tagged one-time signature scheme in Section 5 whose security tightly reduced to DLIN.

As also shown by [31], SPS schemes allow to implement simulation-sound NIZK proofs based on the Groth-Sahai proof system. Following the Naor-Yung-Sahai [35, 38] paradigm, one obtains structure-preserving CCA-secure public-key encryption in a modular fashion.

Open Questions. 1) Can we have (X)RMA-secure schemes with a message space that is a simple Cartesian product of groups without sacrificing on efficiency? 2) The RMA-secure signature schemes developed in this paper are in fact XRMA-secure. Can we have more efficient schemes by resorting to RMA-security? 3) Can we have tagged one-time signature schemes with tight reduction to the underlying simple assumptions? 4) What is the exact lower bound for the size of signatures under simple assumptions? Is it possible to show such a bound?

References

1. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *Advances in Cryptology - CRYPTO*, LNCS, pages 209–237, 2010.
2. M. Abe, J. Groth, K. Haralambiev, and M. Ohkubo. Optimal structure-prserving signatures in asymmetric bilinear groups. In *Advances in Cryptology — CRYPTO '11*, LNCS. Springer-Verlag, 2011.
3. M. Abe, J. Groth, and M. Ohkubo. Separating short structure preserving signatures from non-interactive assumptions. In *Advances in Cryptology – Asiacrypt 2011*, LNCS. Springer-Verlag, 2011.
4. M. Abe, K. Haralambiev, and M. Ohkubo. Signing on group elements for modular protocol designs. IACR ePrint Archive, Report 2010/133, 2010. `http://eprint.iacr.org`.
5. M. Abe and M. Ohkubo. A framework for universally composable non-committing blind signatures. *IJACT*, 2(3):229–249, 2012.
6. M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In S. Halevi, editor, *CRYPTO*, volume 5677 of *LNCS*, pages 108–125. Springer, 2009.
7. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements and a construction based on general assumptions. In E. Biham, editor, *Advances in Cryptology - EUROCRPYT '03*, volume 2656 of *LNCS*, pages 614–629, 2003.
8. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In A. Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *LNCS*, pages 136–154. Springer-Verlag, 2005. Full version available at IACR e-print 2004/077.
9. M. Bellare and S. Shoup. Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. In *Public-Key Cryptography*, volume 4450 of *LNCS*, pages 201–216, 2007.
10. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *Advances in Cryptology — CRYPTO '04*, volume 3152 of *LNCS*, pages 41–55. Springer-Verlag, 2004.
11. D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In E. Biham, editor, *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 416–432. Springer-Verlag, 2003.
12. J. Camenisch, M. Dubovitskaya, and K. Haralambiev. Efficiently signing group elements under simple assumptions. Unpublished Manuscript, available from the authors.
13. J. Camenisch, M. Dubovitskaya, and K. Haralambiev. Efficient structure-preserving signature scheme from standard assumptions. In *SCN*, volume 7485 of *LNCS*, pages 76–94. Springer, 2012.
14. J. Cathalo, B. Libert, and M. Yung. Group encryption: Non-interactive realization in the standard model. In M. Matsui, editor, *Advances in Cryptology - ASIACRYPT*, volume 5912 of *LNCS*, pages 179–196, 2009.
15. M. Chase and M. Kohlweiss. A new hash-and-sign approach and structure-preserving signatures from DLIN. In *SCN*, volume 7485 of *LNCS*, pages 131–148. Springer, 2012.
16. M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn. Malleable proof systems and applications. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT*, volume 7237 of *LNCS*, pages 281–300. Springer, 2012.
17. D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.
18. C. Dwork and M. Naor. An efficient existentially unforgeable signature scheme and its applications. *J. Cryptology*, 11(3):187–208, 1998.

19. S. Even, O. Goldreich, and S. Micali. On-line/off-line digital signatures. *J. Cryptology*, 9(1):35–67, 1996.
20. M. Fischlin. Round-optimal composable blind signatures in the common reference model. In C. Dwork, editor, *Advances in Cryptology — CRYPTO*, volume 4117 of *LNCS*, pages 60–77, 2006.
21. G. Fuchsbauer. Commuting signatures and verifiable encryption. In *Advances in Cryptology — Eurocrypt '11*, LNCS, pages 224–245. Springer-Verlag, 2011.
22. G. Fuchsbauer and D. Pointcheval. Anonymous proxy signatures. In R. Ostrovsky, R. D. Prisco, and I. Visconti, editors, *SCN*, volume 5229 of *LNCS*, pages 201–217. Springer, 2008.
23. G. Fuchsbauer, D. Pointcheval, and D. Vergnaud. Transferable constant-size fair e-cash. In J. A. Garay, A. Miyaji, and A. Otsuka, editors, *CANS*, volume 5888 of *LNCS*, pages 226–247, 2009.
24. G. Fuchsbauer and D. Vergnaud. Fair blind signatures without random oracles. In *AFRICACRYPT*, pages 16–33, 2010.
25. S. D. Galbraith, K. G. Peterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 12008.
26. S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.
27. M. Green and S. Hohenberger. Universally composable adaptive oblivious transfer. In J. Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT*, volume 5350 of *LNCS*, pages 179–197, 2008.
28. M. Green and S. Hohenberger. Practical adaptive oblivious transfer from simple assumptions. In Y. Ishai, editor, *TCC*, volume 6597 of *LNCS*, pages 347–363. Springer, 2011.
29. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In X. Lai and K. Chen, editors, *Advances in Cryptology - ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer-Verlag, 2006.
30. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Advances in Cryptology — Eurocrypt '08*, volume 4965 of *LNCS*, pages 415–432. Springer-Verlag, 2008.
31. D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. In *CRYPTO*, volume 7417 of *LNCS*, pages 590–607. Springer, 2012.
32. A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In *Advances in Cryptology – Eurocrypt 2005*, volume 3494 of *LNCS*, pages 198–214. Springer-Verlag, 2005.
33. B. Libert, T. Peters, and M. Yung. Scalable group signatures with revocation. In *Advances in Cryptology – Eurocrypt 2012*, LNCS. Springer-Verlag, 2012.
34. Y. Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. *J. Cryptology*, 19(3):359–377, 2006.
35. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC'90*, pages 427–437, 1990.
36. S. C. Ramanna, S. Chatterjee, and P. Sarkar. Variants of Waters' dual system primitives using asymmetric pairings - (extended abstract). In M. Fischlin, J. Buchmann, and M. Manulis, editors, *Public Key Cryptography*, volume 7293 of *LNCS*, pages 298–315. Springer, 2012.
37. M. Rückert and D. Schröder. Security of verifiably encrypted signatures and a construction without random oracles. In H. Shacham and B. Waters, editors, *Pairing*, volume 5671 of *LNCS*, pages 17–34. Springer, 2009.
38. A. Sahai. Non-malleable non-interactive zero-knowledge and chosen-ciphertext security. In *FOCS'99*, pages 543–553, 1999.
39. A. D. Santis, G. D. Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero knowledge. In J. Kilian, editor, *CRYPTO*, volume 2139 of *LNCS*, pages 566–598. Springer, 2001.
40. V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, volume 1233 of *LNCS*, pages 256–266. Springer-Verlag, 1997.
41. B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *Advances in Cryptology - CRYPTO 2009*, pages 619–636. Springer-Verlag, 2009.