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

Masayuki Abe* Melissa Chase** Bernardo David*** Markulf Kohlweiss†
Ryo Nishimaki* Miyako Ohkubo††

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

** Microsoft Research, Redmond, USA
melissac@microsoft.com

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

† Microsoft Research, Cambridge, UK
markulf@microsoft.com

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

Abstract

This paper presents efficient structure-preserving signature schemes based on assumptions as simple as Decision-Linear. We first give two general frameworks for constructing fully secure signature schemes from weaker building blocks such as two-tier 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

Contents

1 Introduction **1**
1.1 Our contribution . . . 1
1.2 Related Works . . . 2

2 Preliminaries **2**
2.1 Notation . . . 2
2.2 Bilinear groups . . . 2
2.3 Assumptions . . . 3

3 Definitions **4**
3.1 Common setup . . . 4
3.2 Signature schemes . . . 4
3.3 Partial one-time and tagged one-time signatures . . . 5
3.4 Structure-preserving signatures . . . 6

4 Generic Constructions **6**
4.1 SIG1: Combining tagged one-time and RMA-secure signatures . . . 6
4.2 SIG2: Combining partial one-time and XRMA-secure signatures . . . 7

5 Instantiating SIG1 **8**
5.1 Setup . . . 8
5.2 Tagged one-time signature scheme . . . 8
5.3 RMA-secure signature scheme . . . 11
5.4 Security and efficiency of resulting SIG1 . . . 17

6 Instantiating SIG2 **17**
6.1 Setup . . . 18
6.2 Partial one-time signatures for unilateral messages . . . 18
6.3 Partial one-time signatures for bilateral messages . . . 20
6.4 XRMA-secure signature scheme . . . 20
6.5 Security and efficiency of resulting SIG2 . . . 24

7 Other Instantiation **24**
7.1 Partial one-time signature scheme based on DLIN . . . 24

8 Applications and Open Questions **27**

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) [29]. 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 [27] showed how to construct an efficient simulation-sound zero-knowledge proof systems (ss-NIZK) building on generic constructions of [15, 38, 33]. Abe et al. [4] show how to obtain efficient round-optimal blind signatures by instantiating a framework by Fischlin [18]. SPS are also important building blocks for a wide range of cryptographic functionalities such as anonymous proxy signatures [20], delegatable anonymous credentials [6], transferable e-cash [21] and compact verifiable shuffles [14]. Most recently, [30] show how to construct a structure preserving tree-based signature scheme with a tight security reduction following the approach of [24, 16]. This signature scheme is then used to build a ss-NIZK which in turn is used with the Naor-Yung-Sahai [34, 37] 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, 31, 25, 5, 36, 22, 19, 26].

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 [27] (only partially answered by [13]) 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.

1.1 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 and 2 in Section 5.4 and 6.5, respectively, 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 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. The auxiliary information facilitates access to the one-time secret-key by the reduction.

- To instantiate $\mathsf{SIG1}$, 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 $\mathsf{SIG1}$ 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 unilateral 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 [40]. 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' 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 [30] to construct CCA-secure public-key encryption schemes with a proof of security that tightly reduces to DLIN or SXDH.

1.2 Related Works

Even, Goldreich and Micali [17] 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 [17], 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 [30] 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 [24]. Kohlweiss and Chase [13] construct a SPS scheme with security based on DLIN that improve the performance of Groth's scheme [27] 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.

2 Preliminaries

2.1 Notation

Appending element y to a sequence $X = (x_1, \dots, x_n)$ is denoted by (X, y), i.e., $(X, y) = (x_1, \dots, x_n, y)$.

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

2.2 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 $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Following the terminology in [23] 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}.

2.3 Assumptions

We first define computational and decisional Diffie-Hellman assumptions (co-CDH, DDH_1) and decision 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: co-CDH).
The co-CDH assumption holds if, for any polynomial-time algorithm $\mathcal{A}$, 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).
Let $\mathcal{G}_{\mathsf{ddh1}}(1^\lambda)$ be an algorithm that, on input security parameter 1^λ, runs group generator $\Lambda := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$, chooses $G \leftarrow \mathbb{G}_1$ and $x, y, z \leftarrow \mathbb{Z}_p$, and outputs $I_{\mathsf{ddh1}} := (\Lambda, G, G^x, G^y)$ and (x, y, z). The DDH_1 assumption holds if for polynomial-time adversary $\mathcal{A}$, advantage $\mathrm{Adv}^{\mathsf{ddh1}}_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\, 1 \leftarrow \mathcal{A}(I_{\mathsf{ddh1}}, G^{xy}) \mid (I_{\mathsf{ddh1}}, x, y, z) \leftarrow \mathcal{G}_{\mathsf{ddh1}}(1^\lambda)\,] - \Pr[\, 1 \leftarrow \mathcal{A}(I_{\mathsf{ddh1}}, G^z) \mid (I_{\mathsf{ddh1}}, x, y, z) \leftarrow \mathcal{G}_{\mathsf{ddh1}}(1^\lambda)\,]\,|$ is negligible in λ.

Definition 3 (Decision Linear Assumption in $\mathbb{G}_1$: DLIN_1).
Let $\mathcal{G}_{\mathsf{dlin1}}(1^\lambda)$ be an algorithm that on input security parameter λ, runs group generator $\Lambda := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$, selects $x, y, z \leftarrow \mathbb{Z}_p$ and $G_1, G_2, G_3 \leftarrow \mathbb{G}_1^*$, and outputs $I_{\mathsf{dlin1}} := (\Lambda, G_1, G_2, G_3, G_1^x, G_2^y)$ and (x, y, z). The DLIN_1 assumption holds if, for all polynomial-time adversary $\mathcal{A}$, advantage $\mathrm{Adv}^{\mathsf{dlin1}}_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\, 1 \leftarrow \mathcal{A}(I_{\mathsf{dlin1}}, {G_3}^{x+y}) \mid (I_{\mathsf{dlin1}}, x, y, z) \leftarrow \mathcal{G}_{\mathsf{dlin1}}(1^\lambda)\,] - \Pr[\, 1 \leftarrow \mathcal{A}(I_{\mathsf{dlin1}}, {G_3}^z) \mid (I_{\mathsf{dlin1}}, x, y, z) \leftarrow \mathcal{G}_{\mathsf{dlin1}}(1^\lambda)\,]\,|$ 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 $\mathrm{Adv}^{\mathsf{sxdh}}_{\mathcal{G},\mathcal{C}}(\lambda) \stackrel{\text{def}}{=} \mathrm{Adv}^{\mathsf{ddh1}}_{\mathcal{G},\mathcal{A}}(\lambda) + \mathrm{Adv}^{\mathsf{ddh2}}_{\mathcal{G},\mathcal{B}}(\lambda)$. We extend DLIN in a similar manner as DDH, and SXDH.

Definition 4 (External Decision Linear Assumption in $\mathbb{G}_1$: XDLIN_1).
Let $\mathcal{G}_{\mathsf{xdlin}}(1^\lambda)$ be an algorithm that on input security parameter λ, runs group generator $\Lambda := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$, selects $x, y, z \leftarrow \mathbb{Z}_p$ and $G_1, G_2, G_3 \leftarrow \mathbb{G}_1^*$, $\hat{G}_1, \hat{G}_2, \hat{G}_3 \in \mathbb{G}_2^*$ such that $(G_1, G_2, G_3) \sim (\hat{G}_1, \hat{G}_2, \hat{G}_3)$, and outputs $I_{\mathsf{xdlin}} := (\Lambda, G_1, G_2, G_3, \hat{G}_1, \hat{G}_2, \hat{G}_3, G_1^x, G_2^y, \hat{G}_1^x, \hat{G}_2^y)$ and (x, y, z). The XDLIN_1 assumption holds if, for all polynomial-time adversary $\mathcal{A}$, advantage $\mathrm{Adv}^{\mathsf{xdlin1}}_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\, 1 \leftarrow \mathcal{A}(I_{\mathsf{xdlin}}, {G_3}^{x+y}) \mid (I_{\mathsf{xdlin}}, x, y, z) \leftarrow \mathcal{G}_{\mathsf{xdlin}}(1^\lambda)\,] - \Pr[\, 1 \leftarrow \mathcal{A}(I_{\mathsf{xdlin}}, {G_3}^z) \mid (I_{\mathsf{xdlin}}, x, y, z) \leftarrow \mathcal{G}_{\mathsf{xdlin}}(1^\lambda)\,]\,|$ is negligible in λ.

The XDLIN_1 assumption is equivalent to the DLIN_1 assumption in the generic bilinear group model [39, 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 $\hat{G}_3^{x+y}$, respectively $\hat{G}_3^z$, to $\mathcal{A}$ instead. Then we define the simultaneous external decision Diffie-Hellman 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.).

Finally we recall two computational assumptions (tightly) reduced from one of the above basic assumptions.

Definition 5 (Double Pairing Assumption in $\mathbb{G}_1$ [4]:DBP_1).
The DBP_1 assumption holds if, for any polynomial-time $\mathcal{A}$, probability $\mathrm{Adv}^{\mathsf{dbp1}}_{\mathcal{G},\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \Pr[\, 1 = e(G_z, Z)\, e(G_r, R) \wedge Z \in \mathbb{G}_2^* \mid \Lambda \leftarrow \mathcal{G}(1^\lambda) \; ; \; (G_z, G_r) \leftarrow \mathbb{G}_1^* \times \mathbb{G}_1^* \; ; \; (Z, R) \leftarrow \mathcal{A}(\Lambda, G_z, G_r)\,]$ 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]. Thus the following holds.

Lemma 1. *SXDH* $\Rightarrow$ *DBP*$_1 \wedge$ *DBP*$_2$. *In particular,* $\mathrm{Adv}^{\textsf{dbp1}}_{\mathcal{G},\mathcal{A}}(\lambda) + \mathrm{Adv}^{\textsf{dbp2}}_{\mathcal{G},\mathcal{B}}(\lambda) \le \mathrm{Adv}^{\textsf{sxdh}}_{\mathcal{G},\mathcal{C}}(\lambda)$ *holds.*

Note that the double pairing assumption does not hold in Type-I groups since $Z = G_r$, $R = G_z$ is a trivial solution. The following analogous assumption will be useful in Type-I groups.

Definition 6 (Simultaneous Double Pairing Assumption [12]: SDP).
The SDP assumption holds if, for any polynomial-time $\mathcal{A}$, advantage $\mathrm{Adv}^{\textsf{sdp}}_{\mathcal{G},\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \Pr[\, 1 = e(G_z, Z)\; e(G_r, R) \wedge 1 = e(H_z, Z)\; e(H_s, S) \wedge Z \in \mathbb{G}^* \mid \Lambda \leftarrow \mathcal{G}(1^\lambda)\;;\; (G_z, G_r, H_z, H_s) \leftarrow \mathbb{G}^{*4}\;;\; (Z, R, S) \leftarrow \mathcal{A}(\Lambda, G_z, G_r, H_z, H_s)\,]$ is negligible in λ.

As shown in [12] for the Type-I setting, the simultaneous double pairing assumption holds for $\mathcal{G}$ if the decision linear assumption holds for $\mathcal{G}$.

Lemma 2. *DLIN* $\Rightarrow$ *SDP. In particular,* $\mathrm{Adv}^{\textsf{sdp}}_{\mathcal{G},\mathcal{A}}(\lambda) \le \mathrm{Adv}^{\textsf{dlin}}_{\mathcal{G},\mathcal{B}}(\lambda)$ *holds.*

3 Definitions

3.1 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.

3.2 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;

- $\mathsf{SIG.Key}(gk)$ is a probabilistic algorithm that generates a long-term public-key vk and a secret-key sk.
- $\mathsf{SIG.Sign}(sk, msg)$ is an algorithm that 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 default attack game that consists of the following sequence of algorithm calls:

$$\mathrm{ATK}(\mathcal{A}, \lambda) = \left[\begin{array}{l} gk \leftarrow \mathsf{Setup}(1^\lambda), \\ (vk, sk) \leftarrow \mathsf{SIG.Key}(gk), \\ (\sigma^\dagger, msg^\dagger) \leftarrow \mathcal{A}^{\mathcal{O}sig}(vk) \end{array}\right] \tag{1}$$

Let Q_m be messages observed by $\mathcal{A}$. The result 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 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)[17]). Random message attack security is defined by the attack game ATK where oracle $\mathcal{O}sig$ is as follows: on receiving a request, it chooses msg uniformly from $\mathcal{M}$ defined by gk, computes $\sigma \leftarrow \mathsf{SIG.Sign}(sk, msg)$, 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 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 (Non-Adaptive Chosen-Message Attack (NACMA)). Non-adaptive chosen message attack security is defined by the following attack game:

$$\mathrm{NACMA}(\mathcal{A}, \lambda) = \begin{bmatrix} gk \leftarrow \mathsf{Setup}(1^\lambda), \\ \vec{msg} \leftarrow \mathcal{A}(gk), \\ (vk, sk) \leftarrow \mathsf{SIG.Key}(gk), \vec{\sigma} \leftarrow \mathsf{SIG.Sign}(sk, \vec{msg}) \\ (\sigma^\dagger, msg^\dagger) \leftarrow \mathcal{A}(vk, \vec{\sigma}) \end{bmatrix} \tag{2}$$

For an attack ATK $\in \{\mathrm{CMA}, \mathrm{NACMA}, \mathrm{RMA}, \mathrm{XRMA}\}$, we define unforgeability as follows.

Definition 13 (Unforgeability against ATK). Signature scheme SIG is unforgeable against attack ATK (UF-ATK), if for all polynomial-time oracle algorithm $\mathcal{A}$ the advantage function $\mathrm{Adv}^{\mathsf{uf\text{-}atk}}_{\mathsf{SIG},\mathcal{A}}$ is negligible in λ, where

$$\mathrm{Adv}^{\mathsf{uf\text{-}atk}}_{\mathsf{SIG},\mathcal{A}}(\lambda) = \Pr\left[\begin{array}{l} msg^\dagger \notin Q_m \wedge \\ 1 = \mathsf{SIG.Vrf}(vk, \sigma^\dagger, msg^\dagger) \end{array} \middle| (vk, \sigma^\dagger, msg^\dagger, Q_m) \leftarrow \mathrm{ATK}(\mathcal{A}, \lambda) \right]. \tag{3}$$

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

3.3 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 14 (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 however, 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 secret-keys 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 15 (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 secret-key sk and tag 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 16 (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 polynomial-time oracle algorithm $\mathcal{A}$ the advantage function $\mathrm{Adv}^{\text{ot-cma}}_{\mathsf{POS},\mathcal{A}}$ is negligible in λ, where

$$\mathrm{Adv}^{\text{ot-cma}}_{\mathsf{POS},\mathcal{A}}(\lambda) = \Pr\left[\begin{array}{l|l} \exists(opk, msg) \in Q_m \text{ s.t.} & gk \leftarrow \mathsf{Setup}(1^\lambda), \\ opk^\dagger = opk \ \wedge \ msg^\dagger \neq msg \ \wedge & (pk, sk) \leftarrow \mathsf{POS.Key}(gk), \\ 1 = \mathsf{POS.Vrf}(pk, opk^\dagger, \sigma^\dagger, msg^\dagger) & (opk^\dagger, \sigma^\dagger, msg^\dagger) \leftarrow \mathcal{A}^{\mathcal{O}t, \mathcal{O}sig}(pk) \end{array}\right]. \quad (4)$$

$\mathcal{O}t$ 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) in the list Q_m, which is initially empty. $\mathcal{O}sig$ works only once for every session.

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. By $\mathrm{Adv}^{\text{ot-nacma}}_{\mathsf{POS},\mathcal{A}}(\lambda)$ 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 $\mathrm{Adv}^{\text{ot-cma}}_{\mathsf{TOS},\mathcal{A}}$ and $\mathrm{Adv}^{\text{ot-nacma}}_{\mathsf{TOS},\mathcal{A}}$ accordingly.

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 17 (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 $\mathcal{K}_{opk}$.

3.4 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, partial one-time signature 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 security of the above construction 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 1. *$\mathsf{SIG1}$ is unforgeable against adaptive chosen message attacks (UF-CMA) if TOS is uniform-tag and unforgeable against one-time non-adaptive chosen message attacks (OT-NACMA), and rSIG is unforgeable against random message attacks (UF-RMA). In particular,* $\mathrm{Adv}^{\textit{uf-cma}}_{\textit{SIG1},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\textit{ot-nacma}}_{\textit{TOS},\mathcal{B}}(\lambda) + \mathrm{Adv}^{\textit{uf-rma}}_{\textit{rSIG},\mathcal{C}}(\lambda)$.

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 $\mathsf{SIG1}$ 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 3. $|\Pr[\textbf{Game 0}] - \Pr[\textbf{Game 1}]| \leq \mathrm{Adv}^{\textit{ot-nacma}}_{\textit{TOS},\mathcal{B}}(\lambda)$

Proof. Attacker $\mathcal{A}$ wins in Game 0, but loses in Game 1, iff it produces a forgery that reuses a tag from a signing query. We describe a reduction $\mathcal{B}$ that use such an attacker to break the OT-NACMA-security of TOS The reduction $\mathcal{B}$ receives gk and pk_t from the challenger of TOS, sets up vk_r and sk_r honestly by running $\mathsf{rSIG.Key}(gk)$, and provides gk and $vk = (vk_r, pk_t)$ to $\mathcal{A}$.

To answer a signing query, $\mathcal{B}$ uses the signing oracle of TOS to get tag and σ_t, signs tag using sk_r to produce σ_r, and returns $(tag, \sigma_t, \sigma_r)$. When $\mathcal{A}$ produces a forgery $(tag^\dagger, \sigma_t^\dagger, \sigma_r^\dagger)$ on message $msg^\dagger$, $\mathcal{B}$ outputs $(msg^\dagger, tag^\dagger, \sigma_t^\dagger)$ as a forgery for TOS.

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

Lemma 4. $|\Pr[\textbf{Game 1}] - Pr[\textbf{Game 2}]| \leq \mathrm{Adv}^{\textit{uf-rma}}_{\textit{rSIG},\mathcal{C}}(\lambda)$

Proof. Attacker $\mathcal{A}$ wins in Game 1, iff it produces a forgery with a fresh tag. We describe a reduction $\mathcal{C}$ that use $\mathcal{A}$ to break the UF-RMA security of rSIG. Algorithm $\mathcal{C}$ receives gk and vk_r, runs $(pk_t, sk_t) \leftarrow \mathsf{TOS.Key}(gk)$, and provides gk and $vk = (vk_r, pk_t)$ to $\mathcal{A}$.

To answer signing query on message msg, $\mathcal{C}$ consults $\mathcal{O}sig$ and receives random message msg_r and signature σ_r. $\mathcal{C}$ then uses msg_r as a tag, i.e., $tag = msg_r$, and create signature σ_t on msg by running $\mathsf{TOS.Sign}(sk_t, msg, tag)$. It then returns $(tag, \sigma_t, \sigma_r)$. Note that for a uniform-tag TOS scheme these tags distribute uniformly over the tag space. Thus the reduction simulation is perfect. When $\mathcal{A}$ produces a forgery $(tag^\dagger, \sigma_t^\dagger, \sigma_r^\dagger)$ on $msg^\dagger$, reduction $\mathcal{C}$ outputs $(tag^\dagger, \sigma_r^\dagger)$ as a forgery.

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 2. *If $\mathsf{TOS.Tag}$ produces constant-size tags and signatures in the size of input messages, the resulting $\mathsf{SIG1}$ produces constant-size signatures as well. Furthermore, if TOS and rSIG are structure-preserving, so is $\mathsf{SIG1}$.* (We omit the proof 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 $\mathsf{SIG2}$ from xSIG and POS. Let gk be a global parameter generated by $\mathsf{Setup}(1^\lambda)$.

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

Theorem 3. *$\mathsf{SIG2}$ is unforgeable against adaptive chosen message attacks (UF-CMA) if POS is uniform-key and unforgeable against one-time non-adaptive chosen message attacks (OT-NACMA), and xSIG is unforgeable against extended random message attacks (UF-XRMA) with respect to $\mathsf{POS.Update}$ as the message generator. In particular,* $\mathrm{Adv}^{\textit{uf-cma}}_{\textit{SIG2},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\textit{ot-nacma}}_{\textit{POS},\mathcal{B}}(\lambda) + \mathrm{Adv}^{\textit{uf-xrma}}_{\textit{xSIG},\mathcal{C}}(\lambda)$.

Proof. The proof is almost the same as that for Theorem 1. The only difference appears in constructing $\mathcal{C}$ in the second step. Since $\mathsf{POS.Update}$ 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.

As for our first generic construction, the following theorem holds immediately from the construction.

Theorem 4. *If POS produces constant-size one-time public-keys and signatures in the size of input messages, resulting $\mathsf{SIG2}$ produces constant-size signatures as well. Furthermore, if POS and xSIG are structure-preserving, so is $\mathsf{SIG2}$.*

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 $\mathsf{SIG1}$ scheme is an efficient structure-preserving signature scheme based only on the DLIN assumption.

5.1 Setup

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 $\Lambda \leftarrow \mathcal{G}(1^\lambda)$ and choose 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 also 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.

5.2 Tagged one-time signature scheme

Our tagged one-time scheme is inspired by the commitment scheme in [28].

[Scheme TOS]

$\mathsf{TOS.Key}(gk)$: 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, …, 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 $M_0 := G^m \prod_{i=1}^k M_i^{-1}$. (Note that 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 $\left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right) = \left(\begin{smallmatrix} x_r & x_s \\ y_r & y_s \end{smallmatrix}\right)^{-1}$. (The determinant is nonzero.) Compute $Z := A^\alpha B^\beta$ and $W := A^\gamma B^\delta$. Output $\sigma := (Z, W, M_0)$.

$\mathsf{TOS.Vrf}(pk, tag, msg, \sigma)$: Parse the input accordingly. Accept if the following two 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 $(T_3, \dots, T_6)$ 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 verified by inspecting the following relations. First, observe that

$$\begin{aligned} e(G_r, Z) \cdot e(G_s, W) &= e(G^{x_r}, A^{\alpha} B^{\beta}) \cdot e(G^{x_s}, A^{\gamma} B^{\delta}) \\ &= e(G, A^{x_r\alpha + x_s\gamma}) e(G, B^{x_r\beta + x_s\delta}) \\ &= e(G, A). \end{aligned}$$

Second, observe that

$$\begin{aligned} \prod_{i=0}^{k} e(G_i T_1, M_i) &= e(T_1, \prod_{i=0}^{k} M_i) \prod_{i=0}^{k} e(G_i, M_i) \\ &= e(T_1, G^m \prod_{i=1}^{k} M_i^{-1} \prod_{i=1}^{k} M_i) \prod_{i=0}^{k} e(G_i, M_i) \\ &= e(T_1, G^m) \prod_{i=0}^{k} e(G_i, M_i). \end{aligned}$$

Third, observe that

$$\begin{aligned} e(G, A) &= e(G, G^{-x_t} T_1^{-m} \prod_{i=0}^{k} M_i^{-x_i}) \\ &= e(G_t, G^{-1}) e(T_1, G^{-m}) \prod_{i=0}^{k} e(G_i, M_i^{-1}). \end{aligned}$$

Using these three observations, one can check that

$$\begin{aligned} & e(G_r, Z) \cdot e(G_s, W) \cdot e(G_t, G) \prod_{i=0}^{k} e(G_i T_1, M_i) \\ &= e(G, A) \cdot e(G_t, G) \prod_{i=0}^{k} e(G_i T_1, M_i) \\ &= e(G, A) \cdot e(G_t, G) e(T_1, G^m) \prod_{i=0}^{k} e(G_i, M_i) \\ &= e(G_t, G^{-1}) e(T_1, G^{-m}) \prod_{i=0}^{k} e(G_i, M_i^{-1}) \cdot e(G_t, G) e(T_1, G^m) \prod_{i=0}^{k} e(G_i, M_i) \\ &= 1\ . \end{aligned}$$

The second verification equation is checked analogously.

Theorem 5. *The above TOS scheme is unforgeable against one-time adaptive chosen message attacks under the simultaneous double pairing assumption. In particular, for any $\mathcal{A}$ that makes at most q_s signing queries,* $\mathrm{Adv}^{\textsf{ot-cma}}_{\textsf{TOS},\mathcal{A}}(\lambda) \leq q_s \cdot \mathrm{Adv}^{\textsf{sdp}}_{\mathcal{G},\mathcal{B}}(\lambda) + 1/p$ *holds.*

Proof. Here, we will show that the Tagged One-Time signature scheme described in section 5 satisfies one-time chosen message attack (OT-CMA) security. The reduction gets $\Lambda, G_r, G_s, H_r, H_s$, 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 $i = 0 \dots k$, and sets $pk = (G, G_r, G_s, G_t, H_r, H_s, H_t, G_0, \dots G_k, H_0, \dots, H_k)$. (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 $M = (M_1, \dots, M_k) \in \mathbb{G}^k$ 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

  $$e(G_t, G) \prod_{i=0}^{k} e(G_t^{\tau_i+\tau}, M_i) = e(G_t, M_0^{\tau_0+\tau} G \prod_{i=1}^{k} M_i^{\tau_i+\tau}) = e(G_t, M_0^{\tau_0+\tau} G \prod_{i=1}^{k} M_i^{\tau_i+\tau}) = 1$$

  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 $\tau_0, \dots, \tau_k$ even given tag. (To see this, let $m_0, \dots, m_k$ be the discrete logarithms of $M_0, \dots, M_k$ respectively and note that for any choice of $m_1, \dots, m_k, \tau_0, \dots, \tau_k$ and for any m_0 such that $m_0 \neq -\sum_{i=1}^{k} m_i$, there is a $\frac{1}{q_s}$ 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

  $$Z = \prod_{i=0}^{k} M_i^{-\rho_i-\rho} \text{ and } W = \prod_{i=0}^{k} M_i^{-\sigma_i-\sigma}$$

  and output the signature (Z, W, M_0).

  Note that these are the unique values such that

  $$e(G_r, Z) \cdot e(G_s, W) \cdot e(G_t, G) \prod_{i=0}^{k} e(G_i T_1, M_i) = 1 \text{ and}$$

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

  Thus, Z, W are uniquely determined by $M_0, M_1, \dots, M_k, tag$, and pk. $M_1, \dots, M_k$ are provided by the adversary and, as we have argued, M_0, tag, pk are statistically independent of $\tau_0, \dots, \tau_k$. We conclude that Z, W reveal no additional information about $\tau_0, \dots, \tau_k$ 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

  $$e(H_t, G) \prod_{i=0}^{k} e(H_t^{\tau_i}, M_i) = e(H_t, M_0^{\tau_0} G \prod_{i=1}^{k} M_i^{\tau_i}) = 1.$$

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

$$Z = \prod_{i=0}^{k} M_i^{-\rho_i - \rho} \quad \text{and} \quad W = \prod_{i=0}^{k} M_i^{-\sigma_i - \sigma},$$

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

$$\begin{aligned} 1 &= e(G_r, Z^\dagger)\, e(G_s, W^\dagger)\, e(G_t, G) \prod_{i=0}^{k} e(G_i T_1, M_i^\dagger) \\ &= e(G_r, Z^\dagger)\, e(G_s, W^\dagger)\, e(G_r^\xi G_s^\eta, G) \prod_{i=0}^{k} e(G_r^{\rho_i + \rho + \xi\tau_i} G_s^{\sigma_i + \sigma + \eta\tau_i}, M_i^\dagger) \\ &= e(G_r, Z^\dagger G^\xi \prod_{i=0}^{k} (M_i^\dagger)^{\rho_i + \rho + \xi\tau_i}) e(G_s, W^\dagger G^\eta \prod_{i=0}^{k} (M_i^\dagger)^{\sigma_i + \sigma + \eta\tau_i}), \end{aligned}$$

and

$$\begin{aligned} 1 &= e(H_r, Z^\dagger)\, e(H_s, W^\dagger)\, e(H_t, G) \prod_{i=0}^{k} e(H_i T_2, M_i^\dagger) \\ &= e(H_r, Z^\dagger)\, e(H_s, W^\dagger)\, e(H_r^\xi H_s^\mu, G) \prod_{i=0}^{k} e(H_r^{\rho_i + \rho + \xi\tau_i} H_s^{\sigma_i + \sigma + \mu\tau_i}, M_i^\dagger) \\ &= e(H_r, Z^\dagger G^\xi \prod_{i=0}^{k} (M_i^\dagger)^{\rho_i + \rho + \xi\tau_i}) e(H_s, W^\dagger G^\mu \prod_{i=0}^{k} (M_i^\dagger)^{\sigma_i + \sigma + \mu\tau_i}). \end{aligned}$$

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 $\mathrm{Adv}_{\mathsf{TOS},\mathcal{A}}^{\mathsf{ot\text{-}cma}}(\lambda) \leq q_s \cdot \mathrm{Adv}_{\mathcal{G},\mathcal{B}}^{\mathsf{sdp}}(\lambda) + 1/p$.

5.3 RMA-secure signature scheme

For our random message signature scheme we will use a construction based on the dual system signature proposed in [40]. 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.

[Scheme rSIG]

$\mathsf{rSIG.Key}(gk)$: 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

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

$\mathsf{rSIG.Sign}(sk, msg)$: 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 $\sigma := (S_0, S_1, \dots S_7)$ where

$$\begin{array}{llll} S_0 := (M_5 M_6 H)^{r_1}, & S_1 := K_2 V^r, & S_2 := K_1^{-1} V_1^r G^{z_1}, & S_3 := B^{-z_1}, \\ S_4 := V_2^r G^{z_2}, & S_5 := B^{-z_2}, & S_6 := B^{r_2}, & S_7 := G^{r_1}. \end{array}$$

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

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

The scheme is structure-preserving by construction and the correctness is verified by inspecting the following relation.

$$\begin{aligned} e(S_1, G^b)\, e(S_2, G^{b\cdot a_1})\, e(S_3, G^{a_1}) &= e(K_2 V^r, G^b)\, e(K_1^{-1} V_1^r G^{z_1}, G^{b\cdot a_1})\, e(B^{-z_1}, G^{a_1}) \\ &= e(G^{\alpha\cdot a_1} V^r, G^b)\, e(G^{-\alpha} V_1^r G^{z_1}, G^{b\cdot a_1})\, e(G^{-b\cdot z_1}, G^{a_1}) \\ &= e(G^{\alpha\cdot a_1} V^r, G^b)\, e(G^{-\alpha} V_1^r, G^{b\cdot a_1}) \\ &= e(G^{\alpha\cdot a_1} V^r, G^b)\, e(G^{-\alpha\cdot a_1} V_1^{r\cdot a_1}, G^b) \\ &= e(V^r, G^b)\, e(V_1^{r\cdot a_1}, G^b) \\ &= e(G, VV_1^{a_1})^{b\cdot r} \end{aligned}$$

$$\begin{aligned} e(S_6, VV_1^{a_1})\, e(S_7, R_1^b) &= e(B^{r_2}, VV_1^{a_1})\, e(G^{r_1}, V^b V_1^{b\cdot a_1}) \\ &= e(G^{b\cdot r_2}, VV_1^{a_1})\, e(G^{r_1}, V^b V_1^{b\cdot a_1}) \\ &= e(G, VV_1^{a_1})^{b\cdot r} \end{aligned}$$

Thus, the second equation holds since $r = r_1 + r_2$. The third equation can be verified analogously, and the remaining equations are easily verified.

Theorem 6. *The above rSIG scheme is secure against random message attacks under the DLIN assumption. In particular, for any polynomial-time adversary $\mathcal{A}$ against rSIG that makes at most q_s signing queries, there exists polynomial-time 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

$$\sigma = (S_0, S_1' = S_1 \cdot G^{-a_1 a_2 \gamma}, S_2' = S_2 \cdot G^{a_2 \gamma}, S_3, S_4' = S_4 \cdot G^{a_1 \gamma}, S_5, \dots, S_7)$$

We first give the outline of the proof using some lemmas. Proofs for the lemmas are given after the outline.

Lemma 5. *Any signature that is accepted by the verification algorithm must be formed either as a normal signature, or a simulation-type signature.*

Based on the notion of simulation-type signatures, we consider a sequence of games. Let p_i be the probability that the adversary succeeds in **Game i**, and $p_i^{\text{norm}}(\lambda)$ and $p_i^{\text{sim}}(\lambda)$ that he succeeds with a normal-type respectively simulation-type forgery. Then by Lemma 5, $p_i(\lambda) = p_i^{\text{norm}}(\lambda) + p_i^{\text{sim}}(\lambda)$ for all i.

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

> **Lemma 6.** *In Game 0, the adversary produces a valid forgery which is a simulation-type signature only with negligible probability $p_0^{sim}(\lambda)$ under the DLIN assumption. More concretely, there exists an adversary $\mathcal{B}_1$ such that $p_0^{sim}(\lambda) = \mathrm{Adv}_{\mathcal{G},\mathcal{B}_1}^{\mathsf{dlin}}(\lambda)$.*

Game i: The real security game except that the first i signing queries are answered with simulation-type signatures.

> **Lemma 7.** *The probability that $\mathcal{A}$ outputs a normal-type forgery is the same (up to a negligible amount) in Game $i-1$ as in Game i: $p_{i-1}^{norm}(\lambda) \leq p_i^{norm}(\lambda) + \Delta_i(\lambda)$ for some negligible $\Delta_i(\lambda)$ under the DLIN assumption. More concretely, there exists an adversary $\mathcal{B}_2$ such that $|p_{i-1}^{norm}(\lambda) - p_i^{norm}(\lambda)| = \mathrm{Adv}_{\mathcal{G},\mathcal{B}_2}^{\mathsf{dlin}}(\lambda)$.*

Game q: All private key queries are answered with simulation-type signatures.

> **Lemma 8.** *In Game q, $\mathcal{A}$ outputs a normal-type forgery with at most negligible probability $p_q^{norm}(\lambda)$ under the CDH assumption. More concretely, there exists an adversary $\mathcal{B}_3$ such that $p_q^{norm}(\lambda) = \mathrm{Adv}_{\mathcal{G},\mathcal{B}_3}^{\mathsf{cdh}}(\lambda)$.*

We have shown that in **Game q**, $\mathcal{A}$ can output a normal-type forgery with at most negligible probability. Thus, by Lemma 7 we can conclude that the same is true in **Game 0**. Since we have already shown that in **Game 0** the adversary can output simulation-type forgeries only with negligible probability, and that any signature that is accepted by the verification algorithm is either normal or simulation-type, we conclude that the adversary can produce valid forgeries with only negligible probability

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

Proof. (of Lemma 5)
We have to show that only normal and simulation-type signatures can fulfil these equations. We ignore the first row of verification equations that establish that M is well-formed. A signature has 4 random exponents, r_1, r_2, z_1, z_2. A simulation-type signatures has additional exponent γ.

We interpret S_7 as G^{r_1}, and it follows from the first verification equation that S_0 is $(M_5 M_6 H)^{r_1}$. We interpret S_3 as G^{-bz_1}, S_5 as G^{-bz_2}, and S_6 as $G^{r_2 b}$. Now we have fixed all exponents of a normal signature. The remaining two verification equations tell us that

$$
\begin{aligned}
e(G^b, S_1) \cdot e(G^{ba_1}, S_2) &= e(VV_1^{a_1}, G^{r_2 b}) \cdot e((VV_1^{a_1})^b, G^{r_1}) \cdot e(G^{a_1}, G^{bz_1}), \\
e(G^b, S_1) \cdot e(G^{ba_2}, S_4) &= e(VV_2^{a_2}, G^{r_2 b}) \cdot e((VV_2^{a_2})^b, G^{r_1}) \cdot e(G^{a_2}, G^{bz_2}) \cdot e(G, G)^{\alpha a_1 b}.
\end{aligned}
$$

We interpret S_1 as $G^{\alpha \cdot a_1} V^r G^{-a_1 a_2 \gamma}$. Now we have two equations and two unknowns that fix S_2 to $G^{-\alpha} V_1^r G^{z_1} G^{a_2 \gamma}$ and S_4 to $V_2^r G^{z_2} G^{a_1 \gamma}$ respectively. For $\gamma = 0$ we have a normal signature otherwise a simulation-type signature.

Proof. (of Lemma 6).
Suppose for contradiction that there is an adversary $\mathcal{A}$, which, when playing Game Real (and thus receiving only normal signatures), produces forgeries which are formed like simulation-type signatures. Then we can create an adversary $\mathcal{B}$ for DLIN as follows:

Let $I_{\mathsf{dlin}} = (\Lambda, G_1, G_2, G_3, X, Y, Z)$ be an instance of DLIN where there exist random $x, y, z \in \mathbb{Z}_p$ such that $X = G_1^x$, $Y = G_2^y$ and $Z = G_3^z$ or G_3^{x+y}. Given I_{dlin}, adversary $\mathcal{B}$ works as follows. It first sets $G := G_3$, and $A_1 := G_1$ and $A_2 := G_2$. Then it chooses random $b, \alpha, \rho \in \mathbb{Z}_p$ and computes $B_1 := G_1^b$ and $B_2 := G_2^b$, $K_1 := G_3^\alpha$, $K_2 := G_1^\alpha$ and

$X_1 := G_3^{\rho}$, $X_2 := G_1^{\alpha b/\rho}$. It also chooses random v, v_1, v_2 and sets $V = G_3^{v}$, $V_1 = G_3^{v_1}$, and $V_2 = G_3^{v_2}$. (This way we know the discrete log of these values w.r.t. G_3.) $\mathcal{B}$ further computes $R_1 = VV_1^{a_1} = G_3^{v}G_1^{v_1}$ and $R_2 := VV_2^{a_2} = G_3^{v}G_2^{v_2}$. It then chooses U, H at random from $\mathbb{G}$. Our final public-key is

$$vk = (G_3^b, G_1, G_2, G_1^b, G_2^b, G_3^v G_1^{v_1}, G_3^v G_2^{v_2}, G_3^{vb} G_1^{v_1 b}, G_3^{vb} G_2^{v_2 b}, G_3^v, G_3^{v_1}, G_3^{v_2}, U, H, X_1, X_2)$$

and our final secret key is

$$sk = (vk, G_3^{\alpha}, G_1^{\alpha}).$$

Note that both the distribution of the public and secret keys is statistically close to that in the real DLIN game. Moreover, to sign random messages, $\mathcal{B}$ can follow the real signing algorithm by using sk.

Suppose that $\mathcal{A}$ produces a valid forgery $\sigma^{\dagger}$ and $msg^{\dagger}$. Then $\mathcal{B}$ proceeds as follows: It parses $\sigma^{\dagger}$ as $(S_0, \ldots, S_7)$. Recall that in Lemma 5, it is shown that if the verification equations hold, then we must have $S_1 = G^{\alpha a_1} V^r G^{-a_1 a_2 \gamma}$, $S_2 = G^{-\alpha} V_1^r G^{z_1} G^{a_2 \gamma}$, and $S_4 = V_2^r G^{z_2} G^{a_1 \gamma}$. If this is a simulation-type signature, we will have $\gamma \neq 0$. Rewritten according to our choice of public-key, this means that $S_1 = G_1^{\alpha} V^r G_2^{-f\gamma}$, $S_2 = G_3^{-\alpha} V_1^r V^{z_1} G_2^{\gamma}$, and $S_4 = V_2^r G_3^{z_2} G_1^{\gamma}$, where f is the discrete log of G_1 w.r.t. G_3. Thus, if we can extract $G_2^{-f\gamma}, G_2^{\gamma}, G_1^{\gamma}$, we can easily break the DLIN instance by testing whether $e(Z, G_2^{-f\gamma}) = e(G_2^{\gamma}, X)e(G_1^{\gamma}, Y)$. But, recall that the signature includes $S_3 = G_3^{-bz_1}$, $S_5 = G_3^{-bz_2}$, $S_6 = G_3^{br_2}$, and $S_7 = G_3^{r_1}$, and we know b, α and the discrete logarithms of V, V_1, V_2 w.r.t. G_3. Thus, it will be straightforward to extract the above values.

Proof. (of Lemma 7).
Suppose for contradiction that there exists an adversary $\mathcal{A}$ such that the probabilities that $\mathcal{A}$ outputs a normal-type forgery in Game i and Game $i+1$ differ by a non-negligible amount. Then we will use $\mathcal{A}$ to construct an algorithm $\mathcal{B}$ that breaks the DLIN assumption.

We are given an instance of DLIN; $I_{\mathsf{dlin}} = (G_1, G_2, G_3, X, Y, Z)$. Note that determining whether a signature is of normal-type or simulation-type naturally corresponds to a DLIN problem: each signature contains $S_7 = G^{r_1}$, $S_6 = (G^b)^{r_2}$, and S_1 which will include $V^{r_1+r_2}$ or $V^{r_1+r_2}G^{-a_1 a_2 \gamma}$ depending on whether this is a normal- or simulation-type signature. (Recall that we define $r = r_1 + r_2$.) If we set $G = G_2$, $G^b = G_1$, and $V = G_3$, then it seems fairly straightforward to argue based on the DLIN assumption that it will be impossible for the adversary to distinguish normal and simulation-type signatures. However, we cannot tell whether $\mathcal{A}$'s forgery is normal- or simulation-type in this simulation. Thus, there will be no way for $\mathcal{B}$ to take advantage of a change in $\mathcal{A}$'s success probability to solve the DLIN challenge.

The solution is to set things up so that, with high probability we can take S_0 from the adversary's forgery and extract something that looks like G^{r_1} (which will allow us to distinguish DLIN tuples and consequently detect simulation-type signatures), but at the same time we are guaranteed that for the i-th message, the G component of S_0 will cancel out, leaving only an $G_2^{r_1}$ component which will not allow the challenger itself to know whether a simulated signature is normal-type or simulation-type.

More specifically, the idea will be to choose some secret values $\xi_1, \xi_2, \beta, \chi_1, \chi_2, \chi_3$ and embed them in the parameters so that $U_1^{w_1} U_2^{w_2} H = G_2^{\chi_1 w_1 + \chi_2 w_2 + \chi_3} G_3^{\xi_1 w_1 + \xi_2 w_2 + \beta}$. Then $S_0 = (U_1^{w_1} U_2^{w_2} H)^{r_1} = G_2^{(\chi_1 w_1 + \chi_2 w_2 + \chi_3) r_1} G_3^{(\xi_1 w_1 + \xi_2 w_2 + \beta) r_1}$. If $\xi_1 w_1 + \xi_2 w_2 + \beta \neq 0$, this gives useful information on $G_3^{r_1}$ (in particular it will allow us to test candidate values), while if $\xi_1 w_1 + \xi_2 w_2 + \beta = 0$, this has no G_3 component and thus doesn't help at all with finding $G_3^{r_1}$. We choose ξ_1, ξ_2, β so that $\xi_1 w_1 + \xi_2 w_2 + \beta = 0$ for the (w_1, w_2) used to generate the ith message. Furthermore, we will guarantee that ξ_1, ξ_2, β are information theoretically hidden even given this pair (w_1, w_2), so the adversary has only negligible chance of producing another message with $U_1^{w_1^*}, U_2^{w_2^*}$ such that $\xi_1 w_1^* + \xi_2 w_2^* + \beta = 0$ as well.

Message space setup and key generation: Set (C, F), used to define message space $\mathcal{M}$, to (G_1^{φ}, G_3). We choose random $\xi_1, \xi_2, \beta, \chi_1, \chi_2, \chi_3 \leftarrow \mathbb{Z}_p$, and compute $U_1 = G_2^{\chi_1} G_3^{\xi_1}$, $U_2 = G_2^{\chi_2} G_3^{\xi_2}$, and $H = G_2^{\chi_3} G_3^{\beta}$. These values will be uniformly distributed, and independent of ξ_1, ξ_2, β.

$$gk \;\; = \;\; (\mathbb{G}, C, F, U_1, U_2) = (\mathbb{G}, G_1^{\varphi}, G_3, G_2^{\chi_1} G_3^{\xi_1}, G_2^{\chi_2} G_3^{\xi_2})$$

We set $G = G_2$, $B = G_1$. We choose random $a_1, a_2, \alpha, \rho \leftarrow \mathbb{Z}_p$, and compute G^{a_1}, G^{a_2}, $G^{a_1 b}$, $G^{a_2 b}$, G^{ρ},and $G^{\alpha a_1 b/\rho}$ using these values.

Next, we choose V, V_1, V_2. We must choose these values carefully so that we can compute both R_i and R_i^b, and at the same time so that the component V^r of a signature-value S_1 gives us some useful information (in particular it will allow us to derive G_3^r). We do this by choosing v_1, v_2, δ, and computing $V = G_3^{-a_1 a_2 \delta}$, $V_1 = G_2^{v_1} G_3^{a_2 \delta}$, and $V_2 = G_2^{v_2} G_3^{a_1 \delta}$. Now, this means $R_1 = G_2^{a_1 v_1}$ and $R_2 = G_2^{a_2 v_2}$, and we can easily compute $R_1^b = G_1^{a_1 v_1}$ and $R_2^b = G_1^{a_2 v_2}$. At the same time, note that these values are all distributed identically to the corresponding values in the real public and secret key. Store $a_1, a_2, \alpha, v_1, v_2, \delta$ and

$$sk = (vk, G^{\alpha}, G^{\alpha a_1}) = (vk, G_2^{\alpha}, G_2^{\alpha a_1}).$$

Publish

$$\begin{aligned} vk &= (G^b, G^{a_1}, G^{a_2}, G^{ba_1}, G^{ba_2}, R_1, R_2, R_1^b, R_2^b, V, V_1, V_2, H, G^{\rho}, G^{\alpha a_1 b/\rho}) \\ &= (G_1, G_2^{a_1}, G_2^{a_2}, G_1^{a_1}, G_1^{a_2}, G_2^{a_1 v_1}, G_2^{a_2 v_2}, G_1^{a_1 v_1}, G_1^{a_2 v_2}, G_3^{-a_1 a_2 \delta}, G_2^{v_1} G_3^{a_2 \delta}, G_2^{v_2} G_3^{a_1 \delta}, \\ &\quad G_2^{\chi_3} G_3^{\beta}, G_2^{\rho}, G_1^{\alpha a_1/\rho}). \end{aligned}$$

Note that both of these tuples are distributed statistically close to those produced by Setup and SIGr.Key.

Signatures for j-th message where $j < i$. Pick w_{j1},w_{j2} at random and compute $(M_1, M_2, M_3, M_4, M_5, M_6) = (C^{w_{j1}}, C^{w_{j2}}, F^{w_{j1}}, F^{w_{j2}}, U_1^{w_{j1}}, U_2^{w_{j2}})$. $\mathcal{B}$ can compute a simulation-type signatures for this message since it has sk and $G^{a_1 a_2} = G_2^{a_1 a_2}$.

Signatures for i-th message: Pick w_1,w_2 such that $\xi_1 w_1 + \xi_2 w_2 + \beta = 0$ and compute $(M_1, M_2, M_3, M_4, M_5, M_6) = (C^{w_1}, C^{w_2}, F^{w_1}, F^{w_2}, U_1^{w_1}, U_2^{w_2})$. Note that since no information about ξ_1, ξ_2, β is revealed this message will look appropriately random to the adversary. We will implicitly set $r_1 = y$ and $r_2 = x$. We compute $S_6 = G^{br_2} = G_1^x = X$ and $S_7 = G^{r_1} = G_2^y = Y$. Recall that we chose U_1, U_2, H such that $U_1^{w_1} U_2^{w_2} H = G_2^{\chi_1 w_1 + \chi_2 w_2 + \chi_3}$. Thus, we can compute $S_0 = (M_5 M_6 H)^{r_1} = Y^{\chi_1 w_1 + \chi_2 w_2 + \chi_3}$.

What remains is to compute S_1, S_2, S_4. Note that this involves computing V^r, V_1^r, and V_2^r respectively. This is where we will embed our challenge. Recall that $V = G_3^{-a_1 a_2 \delta}$. Thus, we will compute $V^r = (G_3^{r_1+r_2})^{-a_1 a_2 \delta}$ as $Z^{-a_1 a_2 \delta}$. If $Z = G_3^{x+y}$ this will be correct; if $Z = G_3^z$ for random z, then there will be an extra factor of $G_3^{-a_1 a_2 \delta (z-(x+y))}$. If we let $G^{\gamma} = G_3^{\delta(z-(x+y))}$ (which is uniformly random from the adversary's point of view), then this is distributed exactly as it should be in a simulation-type signature. Thus, we compute S_1 which should be either $G^{\alpha a_1} V^r$ or $G^{\alpha a_1} V^r G^{-a_1 a_2 \gamma}$ as $G_2^{\alpha a_1} Z^{-a_1 a_2 \delta}$.

We can try to apply the same approach to compute V_1^r to get S_1. However, recall that we set $V_1 = G_2^{v_1} G_3^{a_2 \delta}$. Thus, computing V_1^r involves computing G_2^r, which we cannot do. (If we could we could use that to break the DLIN assumption.) To get around this, we use z_1, z_2: choose random s_1, s_2 and implicitly set $G^{z_1} = G_2^{-v_1 r_2 + s_1}$ and $G^{z_2} = G_2^{-v_2 r_2 + s_2}$. While we cannot compute these values, we can compute $G^{-z_1 b} = G_1^{v_1 r_2 - s_1} = X^{v_1} G_1^{-s_1}$ and $G^{-z_2 b} = X^{v_2} G_1^{-s_2}$. Then to generate S_2, we can compute

$$\begin{aligned} G_2^{-\alpha} Y^{v_1} Z^{a_2 \delta} G_2^{s_1} &= G^{-\alpha} G_2^{r_1 v_1} Z^{a_2 \delta} G_2^{s_1} G_2^{r_2 v_1} G_2^{-r_2 v_1} \\ &= G^{-\alpha} G_2^{(r_1+r_2) v_1} Z^{a_2 \delta} G_2^{s_1 - r_2 v_1} \\ &= G^{-\alpha} G_2^{r v_1} Z^{a_2 \delta} G^{z_1} \ . \end{aligned}$$

If $Z = G_3^{x+y} = G_3^r$, then this will be

$$\begin{aligned} G^{-\alpha} G_2^{r v_1} G_3^{r a_2 \delta} G^{z_1} &= G^{-\alpha} (G_2^{v_1} G_3^{a_2 \delta})^r G^{z_1} \\ &= G^{-\alpha} V_1^r G^{z_1} \ . \end{aligned}$$

If $Z = G_3^{z \neq x+y}$, then this will be:

$$\begin{aligned} G^{-\alpha} G_2^{r v_1} G_3^{z a_2 \delta} G^{z_1} &= G^{-\alpha} G_2^{r v_1} G_3^{r a_2 \delta} G_3^{a_2 \delta (z-(x+y))} G^{z_1} \\ &= G^{-\alpha} G_2^{r v_1} G_3^{r a_2 \delta} G^{a_2 \gamma} G^{z_1} \\ &= G^{-\alpha} V_1^r G^{a_2 \gamma} G^{z_1} \end{aligned}$$

where the second to last equality follows from our choice of γ above. By a similar argument, we compute S_4 as $Y^{v_2} Z^{a_1 \delta} G_2^{s_2}$ and argue that this will be either $V_2^r G^{z_2}$ or $V_2^r G^{z_2} G^{a_1 \gamma}$ as desired. Let $S := (S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7)$ where

$$\begin{aligned} &S_0 = Y^{\chi_1 w_{i1} + \chi_2 w_{i2} + \chi_3} && S_1 = G_2^{\alpha a_1} Z^{-a_1 a_2 \delta} && S_2 = G_2^{-\alpha} Y^{v_1} Z^{a_2 \delta} G_2^{s_1} \\ &S_3 = X^{v_1} G_1^{-s_1} && S_4 = Y^{v_2} Z^{a_1 \delta} G_2^{s_2} && S_5 = X^{v_2} G_1^{-s_2} \\ &S_6 = X && S_7 = Y. \end{aligned}$$

Signatures for j-th message where $j > i$: Pick w_1, w_2 and compute $m_j = (M_1, M_2, M_3, M_4, M_5, M_6) = (C^{w_1}, C^{w_2}, F^{w_1}, F^{w_2}, U_1^{w_1}, U_2^{w_2})$ and a signature σ according to $\mathsf{SIGr.Sign}(sk, m_j)$. Output σ, m_j.

On receiving $\mathcal{A}$'s forgery: $\mathcal{A}$ sends a signature $S = (S_0, S_1, \ldots, S_7)$ and $(M_1, M_2, M_3, M_4, M_5, M_6) = (C^{w_1}, C^{w_2}, F^{w_1}, F^{w_2}, U_1^{w_1}, U_2^{w_2})$ for some message w_1, w_2. $\mathcal{B}$ outputs 1 if and only if

$$\begin{aligned}&e(S_0, G_1) \cdot e(M_3^{\xi_1} M_4^{\xi_2} G_3^{\beta}, S_6)\\&= e((S_1 G_2^{-\alpha a_1})^{-1/(-a_1 a_2 \delta)}, (M_1^{1/\varphi})^{\xi_1} (M_2^{1/\varphi})^{\xi_2} G_1^{\beta}) \cdot e(S_7, (M_1^{1/\varphi})^{\chi_1} (M_2^{1/\varphi})^{\chi_2} G_1^{\chi_3})\ .\end{aligned}$$

By Lemma 5, we are guaranteed that if the signature S verifies, then there must exist $w_1, w_2, r_1, r_2, \gamma$ such that $S_0 = (U_1^{w_1} U_2^{w_2} H)^{r_1}$, $S_1 = G^{\alpha a_1} V^r G^{-a_1 a_2 \gamma}$, $S_6 = G^{b r_2}$, and $S_7 = G^{r_1}$ where $r = r_1 + r_2$. We are also guaranteed that $M_1 = (G_1^{\varphi})^{w_1}$, $M_2 = (G_1^{\varphi})^{w_2}$ and $M_3 = G_2^{w_1}$, $M_4 = G_2^{w_2}$.

Rephrased in terms of our parameters, this means

$$\begin{aligned}S_0 &= (G_2^{\chi_1 w_1 + \chi_2 w_2 + \chi_3} G_3^{\xi_1 w_1 + \xi_2 w_2 + \beta})^{r_1} & S_1 &= G_2^{\alpha a_1} G_3^{-a_1 a_2 \delta r} G_2^{-a_1 a_2 \gamma}\\ S_6 &= G_1^{r_2} & S_7 &= G_2^{r_1}\ .\end{aligned}$$

Plugging this into the above computation we get that $\mathcal{B}$ will output 1 if and only if

$$\begin{aligned}&e((G_2^{\chi_1 w_1 \chi_2 w_2 + \chi_3} G_3^{\xi_1 w_1 + \xi_2 w_2 + \beta})^{r_1}, G_1) \cdot e((G_3^{w_1})^{\xi_1} (G_3^{w_2})^{\xi_2} G_3^{\beta}, G_1^{r_2})\\&= e((G_2^{\alpha a_1} G_3^{-a_1 a_2 \delta r} G_2^{-a_1 a_2 \gamma} G_2^{-\alpha a_1})^{1/(-a_1 a_2 \delta)}, (G_1^{w_1})^{\xi_1} (G_1^{w_1})^{\xi_2} G_1^{\beta}) \cdot e(G_2^{r_1}, (G_1^{w_1})^{\chi_1} (G_1^{w_2})^{\chi_2} G_1^{\chi_3})\ .\end{aligned}$$

Simplifying the left side to

$$\begin{aligned}&e(G_2^{\chi_1 w_1 + \chi_2 w_2 + \chi_3} G_3^{\xi_1 w_1 + \xi_2 w_2 + \beta})^{r_1}, G_1) \cdot e(G_3^{\xi_1 w_1 + \xi_2 w_2 + \beta}, G_1^{r_2})\\ =&e(G_2, G_1)^{(\chi_1 w_1 + \chi_2 w_2 + \chi_3) r_1} \cdot e(G_3, G_1)^{(\xi_1 w_1 + \xi_2 w_2 + \beta) r_1} \cdot e(G_3, G_1)^{(\xi_1 w_1 + \xi_2 w_2 + \beta) r_2}\\ =&e(G_2, G_1)^{(\chi_1 w_1 + \chi_2 w_2 + \chi_3) r_1} \cdot e(G_3, G_1)^{(\xi_1 w_1 + \xi_2 w_2 + \beta) r}\end{aligned}$$

and the right side to

$$\begin{aligned}&e((G_3^{-a_1 a_2 \delta r} G_2^{-a_1 a_2 \gamma})^{1/(-a_1 a_2 \delta)}, G_1^{\xi_1 w_1 + \xi_2 w_2 + \beta}) \cdot e(G_2^{r_1}, G_1^{\chi_1 w_1 + \chi_2 w_2 + \chi_3})\\ =&e(G_3^{r} G_2^{\gamma/\delta}, G_1^{\xi_1 w_1 + \xi_2 w_2 + \beta}) \cdot e(G_2^{r_1}, G_1^{\chi_1 w_1 + \chi_2 w_2 + \chi_3})\\ =&e(G_2, G_1)^{(\chi_1 w_1 + \chi_2 w_2 + \chi_3) r_1} \cdot e(G_3, G_1)^{(\xi_1 w_1 + \xi_2 w_2 + \beta) r} \cdot e(G_2, G_1)^{\gamma/\delta(\xi_1 w_1 + \xi_2 w_2 + \beta)}\end{aligned}$$

and by dividing out all the pairings of the left side we obtain the simplified equation

$$1 = e(G_2, G_1)^{\gamma/\delta(\xi_1 w_1 + \xi_2 w_2 + \beta)}$$

which is true if and only if either $\xi_1 w_1 + \xi_2 w_2 + \beta = 0$ or $\gamma = 0$. Since the only information that the adversary has on ξ_1, ξ_2, β is that $\xi_1 w_{i1} + \xi_2 w_{i2} + \beta = 0$, we are guaranteed that $\xi_1 w_1 + \xi_2 w_2 + \beta = 0$ happens with negligible probability. Thus, we conclude that $\mathcal{B}$ outputs 1 iff $\gamma = 0$ and this was a normal-type signature, and $\mathcal{B}$ outputs 0 iff $\gamma \neq 0$ and this was a simulation-type signature.

Proof. (of Lemma 8).
Suppose that there exists an adversary $\mathcal{A}$ that outputs normal-type forgeries with non-negligible probability in Game q. Then we construct an adversary $\mathcal{B}$ for the CDH problem as follows:

$\mathcal{B}$ is given $X = G^x$, $Y = G^y$ and must compute G^{xy}. $\mathcal{B}$ will proceed as follows:

Message space setup and key generation: We will implicitly set $\alpha = xy$ and $a_2 = y$. We choose b, a_1 at random from $\mathbb{Z}_p$. We need to be able to compute $V_2^{a_2}$, so we choose random v_2 and set $V_2 = G^{v_2}$. We also want to know the discrete logarithm of V_1, so we will choose random v_1 and set $V_1 = G^{v_1}$. We choose U_1, U_2, H, V at random from $\mathbb{G}$. We compute $V V_2^{a_2} = V Y^{v_2}$ and $G^{a_2} = Y$. We choose random ρ' and set $G^{\rho} = X^{\rho'}$ and $G^{\alpha a_1 b} = Y^{a_1 b/\rho'}$. The rest of the parameters can be constructed honestly.

Signature queries: On a signature query, we pick w_1,w_2 at random and compute $(M_1, M_2, M_3, M_4, M_5, M_6) = (C^{w_1}, C^{w_2}, F^{w_1}, F^{w_2}, U_1^{w_1}, U_2^{w_2})$ and generate a simulation-type signature as follows: Choose random r_1, r_2, z_1, z_2, and random s. Implicitly set $\gamma = (x - s)$.

Compute $S_1 = Y^{sa_1}V^r = G^{ysa_1}V^r = G^{ysa_1+xya_1-xya_1}V^r = G^{xya_1}V^rG^{(s-x)ya_1} = G^{\alpha a_1}V^rG^{-\gamma a_2 a_1}$, and $S_2 = Y^{-s}V_1^rG^{z_1} = G^{-ys}V_1^rG^{z_1} = G^{-ys+xy-xy}V_1^rG^{z_1} = G^{-xy}V_1^rG^{z_1}G^{(x-s)y} = G^{-\alpha}V_1^rG^{z_1}G^{\gamma a_2}$, and $S_4 = V_2^rG^{z_2}X^{a_1}G^{-sa_1} = V_2^rG^{z_2}G^{xa_1}G^{-sa_1} = V_2^rG^{z_2}G^{(x-s)a_1} = V_2^rG^{z_2}G^{a_1\gamma}$

The rest of the signature can be computed honestly.

Adversary's forgery: When the adversary outputs a normal-type forgery, there exists r_1, r_2, z_1 such that $S_2 = G^{-\alpha}V_1^{r_1+r_2}G^{z_1}$, $S_3 = (G^b)^{-z_1}$, $S_6 = G^{r_2 b}$, and $S_7 = G^{r_1}$. Thus, we can compute

$$\begin{aligned} S_2^{-1} \cdot S_7^{v_1} S_6^{v_1/b} S_3^{-1/b} &= G^{\alpha}V_1^{-(r_1+r_2)}G^{-z_1} \cdot (G^{r_1})^{v_1}(G^{r_2 b})^{v_1/b}((G^b)^{-z_1})^{-1/b} \\ &= G^{\alpha}V_1^{-r_1-r_2}G^{-z_1} \cdot (G^{v_1})^{r_1}(G^{v_1})^{r_2}G^{z_1} \\ &= G^{\alpha}V_1^{-r_1-r_2}G^{-z_1} \cdot V_1^{r_1}V_1^{r_2}G^{z_1} \\ &= G^{\alpha} \ . \end{aligned}$$

$\mathcal{B}$ will output this value. By our choice of parameters, $\alpha = xy$, so $G^{\alpha} = G^{xy}$ as desired.

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 6. 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 1. *Under the DLIN assumption, the above rSIG scheme is secure against extended random message attacks with respect to 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 polynomial-time adversary $\mathcal{A}$ against rSIG that makes at most q_s signing queries, there exists polynomial-time algorithm $\mathcal{B}$ such that $\mathrm{Adv}^{\textit{uf-xrma}}_{\mathsf{rSIG},\mathcal{A}}(\lambda) \leq (q_s + 2) \cdot \mathrm{Adv}^{\textit{dlin}}_{\mathcal{G},\mathcal{B}}(\lambda)$.*

5.4 Security and efficiency of resulting SIG1

Let $\mathsf{SIG1}$ be the signature scheme obtained from TOS and rSIG by following the first generic construction in Section 4. From Theorem 1, 2, 5, 6, and Lemma 2, the following is immediate.

Theorem 7. *$\mathsf{SIG1}$ is a structure-preserving signature scheme that yields constant-size signatures, and is unforgeable against adaptive chosen message attacks under the DLIN assumption. In particular, for any polynomial-time adversary $\mathcal{A}$ for $\mathsf{SIG1}$ making at most q_s signing queries, there exists polynomial-time algorithm $\mathcal{B}$ such that $\mathrm{Adv}^{\textit{uf-cma}}_{\mathsf{SIG1},\mathcal{A}}(\lambda) \leq (2q_s+2) \cdot \mathrm{Adv}^{\textit{dlin}}_{\mathcal{G},\mathcal{B}}(\lambda)$.*

The efficiency is summarised in Table 1 and compared to the scheme in [4]. We measure efficiency by counting the number of group elements and the number of pairing product equations for verifying a singature. The figures do not count the default a generator for each group in gk.

Table 1: Efficiency of $\mathsf{SIG1}$.

Scheme	$\lvert msg \rvert$	$\lvert gk \rvert + \lvert vk \rvert$	$\lvert \sigma \rvert$	#(PPE)	Assmp.
AHO10	k	$2k + 12$	7	2	q-SFP
SIG1	k	$2k + 23$	17	9	DLIN

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 $\mathsf{SIG2}$ scheme is an efficient structure-preserving signature scheme based on SXDH and XDLIN.

6.1 Setup

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 $\Lambda \leftarrow \mathcal{G}(1^\lambda)$ and choose generators $G \in \mathbb{G}_1^*$ and $\hat{G} \in \mathbb{G}_2^*$. Also choose u, f_2, f_3 randomly from $\mathbb{Z}_p^*$ and compute $F_2 := G^{f_2}$, $F_3 := G^{f_3}$, $\hat{F}_2 := \hat{G}^{f_2}$, $\hat{F}_3 := \hat{G}^{f_3}$, $U := G^u$, and $\hat{U} := \hat{G}^u$. 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 Section 6.4. For our generic construction to work, the partial one-time signature scheme should have the same key space.

6.2 Partial one-time signatures for unilateral messages

We construct a partial one-time signature scheme $\mathsf{POSu2}$ for messages in $\mathbb{G}_2^k$ for $k > 0$. The suffix "u2" indicates that the scheme is unilateral and messages are taken from $\mathbb{G}_2$. Correspondingly, $\mathsf{POSu1}$ 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 $\mathsf{POSu2}$ 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 $\mathbb{G}_2^*$ that messages belong to. This structure is very useful for constructing a POS scheme for signing bilateral messages. Just like our rSIG scheme in Section 5.3, our instantiation of xSIG signs random messages from message space $\mathcal{M}_{\mathsf{x}}$ consisting of several related group elements and we extend the key space of $\mathsf{POSu2}$ to fit $\mathcal{M}_{\mathsf{x}}$.

To be able to certify one-time public keys both using our $\mathsf{POSu1}$ and xSIG schemes, we parameterize the one-time key generation function Update with a flag *mode* $\in \{\mathsf{normal}, \mathsf{extended}\}$. If *mode* $= \mathsf{normal}$, it outputs a key in the original form, and if *mode* $= \mathsf{extended}$, it outputs the 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.

[Scheme POSu2]

- $\mathsf{POSu2.Key}(gk)$: 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 $pk := (G_r, G_1, \ldots, G_k) \in \mathbb{G}_1^{k+1}$ and $sk := (\chi_1, \gamma_1, \ldots, \chi_k, \gamma_k, w_r)$.

- $\mathsf{POSu2.Update}(mode)$: 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* $= \mathsf{normal}$ or $opk := (F_2^a, F_3^a, U^a) \in \mathbb{G}_1^3$ if *mode* $= \mathsf{extended}$. Also output $osk := a$.

- $\mathsf{POSu2.Sign}(sk, msg, osk)$: 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
$$\hat{Z} := \hat{U}^{\zeta} \prod_{i=1}^{k} \hat{M}_i^{-\chi_i} \quad \text{and} \quad \hat{R} := \hat{U}^{\rho} \prod_{i=1}^{k} \hat{M}_i^{-\gamma_i}\ . \tag{5}$$

- $\mathsf{POSu2.Vrf}(pk, \sigma, msg, opk)$: 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 $\mathsf{POSu2}$ is structure-preserving and has uniform one-time public-key property from the construction. It is correct as the following relation holds for the verification equation and the computed signatures:

$$\begin{aligned} e(U, \hat{Z})\, e(G_r, \hat{R}) \prod_{i=1}^{k} e(G_i, \hat{M}_i) &= e(U, \hat{U}^{\zeta} \prod_{i=1}^{k} \hat{M}_i^{-\chi_i})\, e(G_r, \hat{U}^{\rho} \prod_{i=1}^{k} \hat{M}_i^{-\gamma_i}) \prod_{i=1}^{k} e(U^{\chi_i} G_r^{\gamma_i}, \hat{M}_i) \\ &= e(U, \hat{U}^{\zeta})\, e(U^{w_r}, \hat{U}^{\rho}) = e(U^{\zeta + w_r \rho}, \hat{U}) = e(A, \hat{U})\ . \end{aligned}$$

Theorem 8. *POSu2 is unforgeable against one-time adaptive chosen message attacks (OT-CMA) if* DBP_1 *holds. In particular,* $\mathrm{Adv}^{\textsf{ot-cma}}_{\textsf{POSu2},\mathcal{A}}(\lambda) \le \mathrm{Adv}^{\textsf{dbp1}}_{\mathcal{G},\mathcal{B}}(\lambda) + 1/p$.

Proof. Making use of successful forger $\mathcal{A}$ against POSu2 as a black-box, we construct $\mathcal{B}$ that is successful in breaking DBP$_1$. We consider the case *mode* $=$ extended in the following. The other case, *mode* $=$ normal can be automatically obtained by dropping A_2 and A_3. Given instance $I_{\mathsf{dbp1}} = (\Lambda, G_z, G_r)$ of DBP$_1$, algorithm $\mathcal{B}$ simulates the attack game against POSu2 as follows. It first build gk by $U := G_z$, $\hat{U} \leftarrow \mathbb{G}_2^*$, and $gk := (\Lambda, U^g, \hat{U}^g, U^{f_2}, U^{f_3}, \hat{U}^{f_2}, \hat{U}^{f_3}, U, \hat{U})$ for $g, f_2, f_3 \leftarrow \mathbb{Z}_p^*$. This yields a gk from the same distribution as produced by Setup. Next $\mathcal{B}$ simulates POSu2.Key by following the original prescription except that Λ and G_r are taken from I_{dbp1}. Note that $w_r = \log_U G_r$ is not known to the simulator but it is not needed in the original POSu2.Key. On receiving one-time key query, algorithm $\mathcal{B}$ simulates POSu2.Update by returning $A := U^{\zeta} G_r^{\rho}$, $A_2 := A^{f_2}$, $A_3 := A^{f_3}$ for $\zeta, \rho \leftarrow \mathbb{Z}_p$, and f_2 and f_3 generated in Setup.

On receiving signing query $msg^{(j)}$ from $\mathcal{A}$, algorithm $\mathcal{B}$ simulates $\mathcal{O}sig$ by simulating POSu2.Sign without having w_r. It is done by using ζ and ρ used in POSu2.Update instead of computing ζ from a. For each signing, transcript (opk, σ, msg) is recorded. When $\mathcal{A}$ outputs a forgery $(opk^{\dagger}, \sigma^{\dagger}, msg^{\dagger})$, algorithm $\mathcal{B}$ searches the records for (opk, σ, msg) such that $opk^{\dagger} = opk$ and $msg^{\dagger} \neq msg$. If no such entry exists, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ computes

$$\hat{Z}^{\star} := \frac{\hat{Z}^{\dagger}}{\hat{Z}} \prod_{i=1}^{k} \left(\frac{\hat{M}_i^{\dagger}}{\hat{M}_i} \right)^{\chi_i}, \quad \text{and} \quad \hat{R}^{\star} := \frac{\hat{R}^{\dagger}}{\hat{R}} \prod_{i=1}^{k} \left(\frac{\hat{M}_i^{\dagger}}{\hat{M}_i} \right)^{\gamma_i}, \tag{6}$$

where $(\hat{Z}, \hat{R}, \hat{M}_1, \ldots, \hat{M}_k)$ and its dagger counterpart are taken from (σ, msg) and $(\sigma^{\dagger}, msg^{\dagger})$, respectively. $\mathcal{B}$ finally outputs $(\hat{Z}^{\star}, \hat{R}^{\star})$. This completes the description of $\mathcal{B}$.

We first claim that the simulation by $\mathcal{B}$ is perfect; the parameters and keys generated in Setup and POSu2.Key due to the uniform choice of $I_{\mathsf{dbp1}} = (\Lambda, G_z, G_r)$, and the distribution of (a, ζ, ρ) is uniform over $\mathbb{Z}_p^3$ under constraint $a = \zeta + \rho w_r$ as well as the original procedure. Accordingly, $\mathcal{A}$ outputs successful forgery with noticeable probability and $\mathcal{B}$ finds a corresponding record (opk, σ, msg).

We next claim that each χ_i is independent of the view of $\mathcal{A}$. Concretely, we show that, if coins $\chi_1, \ldots, \chi_k$ distribute uniformly over $(\mathbb{Z}_p)^k$, other coins $\gamma_1, \ldots, \gamma_k, \zeta^{(1)}, \rho^{(1)}, \ldots, \zeta^{(q_s)}, \rho^{(q_s)}$ distribute uniformly as well retaining consistency with the view of $\mathcal{A}$. Observe that the view of $\mathcal{A}$ making q signing queries consists of independent group elements $(U, \hat{U})$, (G, F_2, F_3), $(G_r, G_1, \ldots, G_k)$ and $(A^{(j)}, \hat{Z}^{(j)}, \hat{M}_1^{(j)}, \ldots, \hat{M}_k^{(j)})$ for $j = 1, \ldots, q_s$. (Note that $\hat{G}$, $\hat{F}_2$, $\hat{F}_3$, and $A_1^{(j)}$, $A_2^{(j)}$, and $\hat{R}^{(j)}$ for all j are uniquely determined from other group elements.) We represent the view by the discrete-logarithms of these group elements with respect to bases U and $\hat{U}$ in each group. Namely, the view is $(g, f_2, f_3, w_r, w_1, \ldots, w_k)$ and $(a^{(j)}, z^{(j)}, m_1^{(j)}, \ldots, m_k^{(j)})$ for $j = 1, \ldots, q_s$. To be consistent, the view and the coins satisfy relations

$$w_i = \chi_i + w_r \gamma_i \quad \text{for } i = 1, \ldots, k, \text{ and} \tag{7}$$

$$a^{(j)} = \zeta^{(j)} + w_r \rho^{(j)}, \quad \text{and} \quad z^{(j)} = \zeta^{(j)} - \sum_{i=1}^{k} m_i^{(j)} \chi_i \quad \text{for } j = 1, \ldots, q_s. \tag{8}$$

From relation (7), $(\gamma_1, \ldots, \gamma_k)$ distributes uniformly according to the uniform distribution of $(\chi_1, \ldots, \chi_k)$. From the second relation in (8) for every j, if $(m_1, \ldots, m_k) \neq (0, \ldots, 0)$ then $\zeta^{(j)}$ distributes uniformly according to the uniform distribution of $(\chi_1, \ldots, \chi_k)$. Then, from first relation of (8), $\rho^{(j)}$ distributes uniformly, too. If $(m_1, \ldots, m_k) = (0, \ldots, 0)$, then $\zeta^{(j)}$ and $\rho^{(j)}$ are independent of $(\chi_1, \ldots, \chi_k)$ and can be uniformly assigned by following the first relation in (8).

Finally, we claim that $(\hat{Z}^{\star}, \hat{R}^{\star})$ is a valid solution to the given instance of DBP$_1$. Since both forged and recorded signatures fulfill the verification equation,

dividing the equations results in

$$\begin{aligned}
1 &= e\left(U, \frac{\hat{Z}^{\dagger}}{\hat{Z}}\right) \; e\left(G_r, \frac{\hat{R}^{\dagger}}{\hat{R}}\right) \prod_{i=1}^{k} e\left(U^{\chi_i} G_r^{\gamma_i}, \frac{\hat{M}_i^{\dagger}}{\hat{M}_i}\right) \\
&= e\left(U, \frac{\hat{Z}^{\dagger}}{\hat{Z}} \prod_{i=1}^{k} \left(\frac{\hat{M}_i^{\dagger}}{M_i}\right)^{\chi_i}\right) \; e\left(G_r, \frac{\hat{R}^{\dagger}}{\hat{R}} \prod_{i=1}^{k} \left(\frac{\hat{M}_i^{\dagger}}{M_i}\right)^{\gamma_i}\right) \\
&= e\left(U, \hat{Z}^{\star}\right) \; e\left(G_r, \hat{R}^{\star}\right).
\end{aligned}$$

What remains is to prove that $\hat{Z}^{\star} \neq 1$. Since $msg^{\dagger} \neq msg^{(j)}$, there exists $\ell \in \{1, \ldots, k\}$ such that $\frac{\hat{M}_{\ell}^{\dagger}}{M_{\ell}} \neq 1$. As already proven, χ_{ℓ} is independent of the view of $\mathcal{A}$. Thus $\left(\frac{M_{\ell}^{\dagger}}{M_{\ell}}\right)^{\chi_{\ell}}$ distributes uniformly over $\mathbb{G}_2$ and so does $\hat{Z}^{\star}$. Accordingly, $Z^{\star} = 1$ holds only if $Z^{\dagger} = \hat{Z} \prod (M_i^{\dagger}/M_i)^{-\chi_i}$, which happens only with probability $1/p$ over the choice of χ_{ℓ}. We thus have $\mathsf{Adv}^{\mathsf{ot\text{-}cma}}_{\mathsf{POSu2},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{dbp1}}_{\mathcal{G},\mathcal{B}}(\lambda) + 1/p$.

6.3 Partial one-time signatures for bilateral messages

Using $\mathsf{POSu1}$ for $msg \in \mathbb{G}_1^{k_1+1}$ and $\mathsf{POSu2}$ for $msg \in \mathbb{G}_2^{k_2}$, we construct a POSb 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 $\mathsf{POSu2}$ 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 $\mathsf{POSu1}$. Public-key opk_2 is included in the signature, and opk_1 is output as a one-time public-key for POSb.

[Scheme POSb]

- $\mathsf{POSb.Key}(gk)$: Run $(pk_1, sk_1) \leftarrow \mathsf{POSu1.Key}(gk)$ and $(pk_2, sk_2) \leftarrow \mathsf{POSu2.Key}(gk)$. Set $pk := (pk_1, pk_2)$ and $sk := (sk_1, sk_2)$, and output (pk, sk).

- $\mathsf{POSb.Update}(mode)$: Run $(opk, osk) \leftarrow \mathsf{POSu1}(mode)$ and output (opk, osk).

- $\mathsf{POSb.Sign}(sk, msg, osk)$: Parse msg into $(msg_1, msg_2) \in \mathbb{G}_1^{k_1} \times \mathbb{G}_2^{k_2}$, and sk accordingly. Run $(opk_2, osk_2) \leftarrow \mathsf{POSu2.Update}(\mathsf{normal})$, and compute signatures $\sigma_2 \leftarrow \mathsf{POSu2.Sign}(sk_2, msg_2, osk_2)$ and $\sigma_1 \leftarrow \mathsf{POSu1.Sign}(sk_1, (msg_1, opk_2), osk)$. Output $\sigma := (\sigma_1, \sigma_2, opk_2)$.

- $\mathsf{POSb.Vrf}(pk, opk, \sigma, msg)$: 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 $1 = \mathsf{POSu1.Vrf}(pk_1, opk, \sigma_1, (msg_1, opk_2)) = \mathsf{POSu2.Vrf}(pk_2, opk_2, \sigma_2, msg_2)$, output 1. Otherwise, output 0.

For a message in $\mathbb{G}_1^{k_1} \times \mathbb{G}_2^{k_2}$, the above POSb uses a public-key of size $(k+2, k+1)$, yields a one-time public-key of size $(0,1)$ (for $mode = \mathsf{normal}$) 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 Section 6.4, 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 $\mathsf{POSu1}$ and $\mathsf{POSu2}$.

Theorem 9. *Scheme POSb is unforgeable against one-time adaptive chosen message attacks if SXDH holds. In particular,* $\mathrm{Adv}^{\textit{ot-cma}}_{\textit{POSb},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\textit{sxdh}}_{\mathcal{G},\mathcal{B}}(\lambda) + 2/p$.

Proof. Suppose an adversary $\mathcal{A}$ outputs a forgery $(opk^\dagger, \sigma^\dagger, msg^\dagger)$. Then there exists a triple (σ, opk, msg) observed by the signing oracle such that $opk^\dagger = opk$ and $msg^\dagger \neq msg$. Let $msg^\dagger = (msg_1^\dagger, msg_2^\dagger)$ and $\sigma^\dagger = (\sigma_1^\dagger, \sigma_2^\dagger, opk_2^\dagger)$. Similarly, let $msg = (msg_1, msg_2)$ and $\sigma = (\sigma_1, \sigma_2, opk_2)$. Then there are two cases; either $(msg_1, opk_2) \neq (msg_1^\dagger, opk_2^\dagger)$, or $(msg_1, opk_2) = (msg_1^\dagger, opk_2^\dagger)$ and $msg_2 \neq msg_2^\dagger$. In the first case we have $1 = \mathsf{POSu1.Vrf}(pk_1, opk, \sigma_1, (msg_1, opk_2)) = \mathsf{POSu1.Vrf}(pk_1, opk, \sigma_1^\dagger, (msg_1^\dagger, opk_2^\dagger))$, which breaks the unforgeability of $\mathsf{POSu1}$ and contradicts the DBP_2 assumption. In the second case we have $1 = \mathsf{POSu2.Vrf}(pk_2, opk_2, \sigma_2, msg_2) = \mathsf{POSu2.Vrf}(pk_2, opk_2, \sigma_2, msg_2^\dagger)$, which breaks the unforgeability of $\mathsf{POSu2}$ and contradicts the DBP_1 assumption. Accordingly, we have $\mathrm{Adv}^{\mathsf{ot\text{-}cma}}_{\mathsf{POSb},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathsf{dbp1}}_{\mathcal{G},\mathcal{A}}(\lambda) + 1/p + \mathrm{Adv}^{\mathsf{dbp2}}_{\mathcal{G},\mathcal{B}}(\lambda) + 1/p \leq \mathrm{Adv}^{\mathsf{sxdh}}_{\mathcal{G},\mathcal{B}}(\lambda) + 2/p$.

6.4 XRMA-secure signature scheme

We use a variant of Waters' dual system encryption proposed by Ramanna, Chatterjee, and Sarkar [35]. 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.

[Scheme xSIG]

$\mathsf{xSIG.Gen}(gk)$**:** 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

$$\begin{aligned} vk &:= (gk, \hat{G}^b, \hat{G}^a, \hat{G}^{ba}, \hat{R}, \hat{R}^b, H, \hat{H}, V, \hat{V}, V', \hat{V}', G^\rho, \hat{G}^{\alpha b/\rho}) \\ sk &:= (VK, G^\alpha, G^a, G^b)\,. \end{aligned}$$

$\mathsf{xSIG.Sign}(sk, msg)$**:** 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 $\sigma := (\sigma_0, \sigma_1, \ldots, \sigma_5) \in \mathbb{G}_2 \times \mathbb{G}_1^5$.

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

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

The scheme is structure-preserving due to the construction. It is correct as the following relations hold for the verification equation and the computed signatures.

$$
\begin{aligned}
e(\sigma_1, \hat{G}^b)\ e(\sigma_2, \hat{G}^{ba})\ e(\sigma_3, \hat{G}^a) &= e(G^\alpha V^r, \hat{G}^b)\ e((V')^r \hat{G}^{-z}, \hat{G}^{ba})\ e(G^{bz}, \hat{G}^a) \\
&= e(G, \hat{G})^{\alpha b}\ e(V, \hat{G})^{br}\ e(V', \hat{G})^{abr} \\
&= e(G, \hat{G})^{\alpha b}\ e(V(V')^a, \hat{G})^{br}
\end{aligned}
$$

$$
\begin{aligned}
e(\sigma_4, \hat{R})\ e(\sigma_5, \hat{R}^b)\ e(G^\rho, \hat{G}^{\alpha b/\rho}) &= e(G^{br_2}, \hat{V}(\hat{V}')^a)\ e(G^{r_1}, \hat{V}^b(\hat{V}')^{ba})\ e(G, \hat{G})^{\alpha b} \\
&= e(G, \hat{V}(\hat{V}')^a)^{br_2}\ e(G, \hat{V}(\hat{V}')^a)^{br_1}\ e(G, \hat{G})^{\alpha b} \\
&= e(G, \hat{G})^{\alpha b}\ e(G, \hat{V}(\hat{V}')^a)^{br}
\end{aligned}
$$

Thus, the second euqation holds since $G \sim \hat{G}$, $V \sim \hat{V}$, $V' \sim \hat{V}'$, and $r = r_1 + r_2$. The first, third, fourth equations are easily verified.

Theorem 10. *If the DDH$_2$ and XDLIN$_1$ assumptions hold, then our* xSIG *scheme is unforgeable against extended random chosen message attacks 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* $\mathrm{Adv}^{\textit{uf-xrma}}_{\textit{xSIG},\mathcal{A}}(\lambda) < \mathrm{Adv}^{\textit{ddh2}}_{\mathcal{G},\mathcal{B}_1}(\lambda) + q\mathrm{Adv}^{\textit{xdlin1}}_{\mathcal{G},\mathcal{B}_2}(\lambda) + \mathrm{Adv}^{\textit{co-cdh}}_{\mathcal{G},\mathcal{B}_3}(\lambda)$.

Proof. The outline of the proof follows that of Water's dual signature scheme and quite similar to the proof of Theorem 6. We start with the following lemma.

Lemma 9. *Any accepted signature by the verification algorithm must be formed either as a normal-type signature or a simulation-type signature.*

Proof. (of Lemma 9) For a signature element σ_5, there exists some $r_1 \in \mathbb{Z}_p$ such that $\sigma_5 = G^{r_1}$, so the first verification equation implies that $\sigma_0 = (\hat{U}^m\hat{H})^{r_1}$. For fixed $b \in \mathbb{Z}_p$ ($\hat{G}^b$ is included in vk), there exists $r_2, z \in \mathbb{Z}_p$ such that $\sigma_3 = G^{bz}$, $\sigma_4 = G^{br_2}$. If we fix $\sigma_1 = G^\alpha V^r G^{-a\gamma}$, then a remaining unknown value is σ_2. The verification equation is

$$
e(\sigma_1, \hat{G}^b)e(\sigma_2, \hat{G}^{ba})e(\sigma_3, \hat{G}^a) \quad = \quad e(\sigma_4, \hat{R})e(\sigma_5, \hat{R}^b)e(G, \hat{G})^{\alpha b}
$$

so we can fix $\sigma_2 = (V')^r G^{-z} G^\gamma$.

Based on the notion of simulation-type signatures, 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 9, $p_i(\lambda) = p_i^{\mathrm{norm}}(\lambda) + p_i^{\mathrm{sim}}(\lambda)$ for all i.

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

> **Lemma 10.** *In Game 0, the adversary produces a valid forgery which is a simulation-type signature only with negligible probability $p_0^{sim}(\lambda)$ under the DDH$_2$ assumption. More concretely, there exists an adversary $\mathcal{B}_1$ such that* $p_0^{sim}(\lambda) \le \mathrm{Adv}^{\textit{ddh2}}_{\mathcal{G},\mathcal{B}_1}(\lambda)$.

Game i: The real security game except that the first i signing queries are answered with simulation-type signatures.

> **Lemma 11.** *The probability that $\mathcal{A}$ outputs a normal-type forgery is the same (up to a negligible amount) in Game $i-1$ as in Game i: $p_{i-1}^{norm}(\lambda) \le p_i^{norm}(\lambda) + \Delta_i(\lambda)$ for some negligible $\Delta_i(\lambda)$ under the XDLIN$_1$ assumption. More concretely, there exists an adversary $\mathcal{B}_2$ such that* $|p_{i-1}^{norm}(\lambda) - p_i^{norm}(\lambda)| \le \mathrm{Adv}^{\textit{xdlin1}}_{\mathcal{G},\mathcal{B}}(\lambda)$.

Game q: All private key queries are answered with simulation-type signatures.

Lemma 12. *In Game q, $\mathcal{A}$ outputs a normal-type forgery with at most negligible probability $p_q^{norm}(\lambda)$ under the co-CDH assumption. More concretely, there exists an adversary $\mathcal{B}_2$ such that $p_q^{norm}(\lambda) \leq \mathrm{Adv}_{\mathcal{G},\mathcal{B}_2}^{\textsf{co-cdh}}(\lambda)$.*

We have shown that in **Game q**, $\mathcal{A}$ can output a normal-type forgery with at most negligible probability. Thus, by Lemma 11 we can conclude that the same is true in **Game 0**. Since we have already shown that in **Game 0** the adversary can output simulation-type forgeries only with negligible probability, and that any signature that is accepted by the verification algorithm is either normal or simulation-type, we conclude that the adversary can produce valid forgeries with only negligible probability

$$\begin{aligned}
\mathrm{Adv}_{\mathsf{xSIG},\mathcal{A}}^{\textsf{uf-xrma}}(\lambda) &= p_0(\lambda) = p_0^{\mathrm{sim}}(\lambda) + p_0^{\mathrm{norm}}(\lambda) = p_0^{\mathrm{sim}}(\lambda) + \sum_{i=1}^{q} |p_{i-1}^{\mathrm{norm}}(\lambda) - p_i^{\mathrm{norm}}(\lambda)| + p_q^{\mathrm{norm}}(\lambda) \\
&\leq \mathrm{Adv}_{\mathcal{G},\mathcal{B}_2}^{\textsf{co-cdh}}(\lambda) + q\mathrm{Adv}_{\mathcal{G},\mathcal{B}}^{\textsf{xdlin1}}(\lambda) + \mathrm{Adv}_{\mathcal{G},\mathcal{B}_1}^{\textsf{ddh2}}(\lambda)
\end{aligned}$$

as stated.

Proof. (of Lemma 10) We show that, if adversary outputs a simulation-type forgery, then we can construct algorithm $\mathcal{B}_1$ that solves the DDH_2 problem. Algorithm $\mathcal{B}_1$ is given instance $(\Lambda, \hat{G}, \hat{G}^s, \hat{G}^a, \hat{Z} \in \mathbb{G}_2)$ of DDH_2, and simulates the verification key and the signing oracle for the signature scheme ($\mathcal{B}_1$ does not have value a, s).

$\mathcal{B}_1$ generates gk and vk as follows. It selects $G \leftarrow \mathbb{G}_1$, and selects exponents $b, \alpha, v, v', u, h, f_2, f_3 \leftarrow \mathbb{Z}_p$ and $\rho \leftarrow \mathbb{Z}_p^*$, computes $\hat{G}^a := \hat{G}^a$, $\hat{G}^b := \hat{G}^b$, $\hat{G}^{ba} := (\hat{G}^a)^b$, $V := G^v$, $V' := G^{v'}$, $\hat{V} := \hat{G}^v$, $\hat{V}' := \hat{G}^{v'}$, $\hat{R} := \hat{V}(\hat{V}')^a = \hat{G}^v(\hat{G}^a)^v$, $U := G^u$, $H := G^h$, $\hat{U} := \hat{G}^u$, $\hat{H} := \hat{G}^h$, $\hat{F}_2 := \hat{G}^{f_2}$, $\hat{F}_3 := \hat{G}^{f_3}$, $G^\rho := G^\rho$, $\hat{G}^{\alpha b/\rho} := \hat{G}^{\alpha b/\rho}$, and sets

$$\begin{aligned}
gk &:= (\Lambda, G, \hat{G}, F_2, \hat{F}_2, F_3, \hat{F}_3, U, \hat{U}), \\
vk &:= (\hat{G}^b, \hat{G}^a, \hat{G}^{ab}, \hat{G}^v\hat{G}^{va}, \hat{G}^{vb}\hat{G}^{vab}, H, \hat{H}, V, \hat{V}, V', \hat{V}', G^\rho, \hat{G}^{\alpha b/\rho}), \\
sk &:= (VK, G^\alpha, G^a, G^b).
\end{aligned}$$

$\mathcal{B}_1$ can generate normal-type signatures by using the (normal) signing algorithm since $\mathcal{B}_1$ has α, b and V, V'.

If adversary $\mathcal{A}$ outputs a simulation-type forgery $\sigma_1 := (G^\alpha V^r) \cdot G^{-a\gamma}$, $\sigma_2 := ((V')^r G^{-z}) \cdot G^\gamma$, $\sigma_3 := (G^b)^{-z}$, $\sigma_4 := (G^b)^{r_2}$, $\sigma_5 := G^{r_1}$, and $\sigma_0 := (\hat{M}_0\hat{H})^{r_1}$, for some $r_1, r_2, z, \gamma \in \mathbb{Z}_p$ ($r = r_1 + r_2$) for message $F(m) = (\hat{F}_2^m, \hat{F}_3^m, \hat{U}^m)$, then $\mathcal{B}_1$ can compute $(G^{a\gamma}, G^\gamma)$ from σ_1, σ_2 respectively. The reason is as follows:

$\mathcal{B}_1$ has b, so can compute G^z, G^{r_1}, G^{r_2} from $\sigma_3 = G^{bz}$, $\sigma_5 = G^{r_1}$, $\sigma_4 = G^{br_2}$, respectively and obtains $G^r = G^{r_1+r_2}$, $V^r = G^{rv}$, $(V')^r = G^{rv'}$ ($\mathcal{B}_1$ has v, v'). Thus, $\mathcal{B}_1$ can extract $(G^{-a\gamma}, G^\gamma)$ from σ_1 and σ_2. $\mathcal{B}_1$ can solve the DDH_2 problem by checking whether

$$e(G^\gamma, \hat{Z}) = e(G^{a\gamma}, \hat{G}^s)$$

or not because $e(G^{a\gamma}, \hat{G}^s) = e(G, \hat{G})^{as\gamma} = e(G^\gamma, \hat{G}^{as})$. If $\hat{Z} = \hat{G}^{as}$ (DDH tuple), then the equation holds. Thus, $\mathcal{B}_1$ solves the DDH_2 problem whenever the adversary outputs a valid simulation-type forgery, i.e., $p_0^{\mathrm{sim}}(\lambda) \leq \mathrm{Adv}_{\mathcal{G},\mathcal{B}_1}^{\textsf{ddh2}}(\lambda)$ as claimed.

Proof. (of Lemma 11) Given access to $\mathcal{A}$ playing $p_{i-1}^{\mathrm{norm}}(\lambda)$ and $p_i^{\mathrm{norm}}(\lambda)$, we construct algorithm $\mathcal{B}_2$ that solves the XDLIN_1 problem with advantage $|p_{i-1}^{\mathrm{norm}}(\lambda) - p_i^{\mathrm{norm}}(\lambda)|$.

$\mathcal{B}_2$ is given instance $(\Lambda, G_1, G_2, G_3, \hat{G}_1, \hat{G}_2, \hat{G}_3, X, Y, \hat{X}, \hat{Y}, Z \in \mathbb{G}_1)$ of the XDLIN_1 problem. It implicitly holds that $G_1 = G_3^b, \hat{G}_1 = \hat{G}_3^b, X = G_1^x, Y = G_2^y, \hat{X} = \hat{G}_1^x, \hat{Y} = \hat{G}_2^y$. $\mathcal{B}_2$ generates the group elements in gk and vk as follows: It selects exponents $\alpha, a, v', u, h, \xi, \beta, \chi_1, \chi_2, \delta \leftarrow \mathbb{Z}_p$ and $\rho \leftarrow \mathbb{Z}_p^*$, such that $\xi m + \beta = 0$ where $m \in \mathbb{Z}_p$ is the exponent of the i-th random message and will be used to answer i-th signature, computes $G := G_2$, $\hat{G} := \hat{G}_2$, $\hat{G}^b := \hat{G}_1$, $\hat{G}^{ba} := \hat{G}_1^a$, $V := G_3^{-a\delta}$, $\hat{V} := \hat{G}_3^{-a\delta}$, $V' := G_3^\delta G_2^{v'}$, $\hat{V}' := \hat{G}_3^\delta \hat{G}_2^{v'}$, $G^\rho := G_2^\rho$, $\hat{G}^{\alpha b/\rho} := (\hat{G}_1)^{\alpha/\rho}$, $R := V(V')^a = G_2^{v'a}$, $\hat{R} := \hat{V}(\hat{V}')^a = \hat{G}_2^{v'a}$, $R^b := (V(V')^a)^b = G_1^{v'a}$, $\hat{R}^b := (\hat{V}(\hat{V}')^a)^b = \hat{G}_1^{v'a}$, $U := G_2^{\chi_1}G_3^\xi$, $\hat{U} := \hat{G}_2^{\chi_1}\hat{G}_3^\xi$, $H := G_2^{\chi_2}G_3^\beta$, $\hat{H} := \hat{G}_2^{\chi_2}\hat{G}_3^\beta$.

If $\xi m + \beta = 0$, then it holds that $(\hat{U}^m\hat{H}) = \hat{G}_2^{m\chi_1+\chi_2}\hat{G}_3^{\xi m+\beta} = \hat{G}_2^{m\chi_1+\chi_2}$. Note that ξ and β are information theoretically hidden even given m, so the adversary has only negligible chance of producing another message $\hat{U}^{m^*}$ such that $\xi m^* + \beta = 0$. We choose $\varphi \leftarrow \mathbb{Z}_p$, set $F_2 := G_1^\varphi$, $F_3 := G_3$, $\hat{F}_2 := \hat{G}_1^\varphi$, $\hat{F}_3 := \hat{G}_3$.

$\mathcal{B}_2$ sets

$$\begin{aligned}
gk &:= (\Lambda, G_2, \hat{G}_2, G_1^\varphi, \hat{G}_1^\varphi, G_3, \hat{G}_3, G_2^{\chi_1}G_3^\xi, \hat{G}_2^{\chi_1}\hat{G}_3^\xi), \\
vk &:= (\hat{G}_1, \hat{G}^a, \hat{G}_1^a, G_2^{v'a}, G_1^{v'a}, \hat{G}_2^{v'a}, \hat{G}_1^{v'a}, G_2^{\chi_2}G_3^\beta, \hat{G}_2^{\chi_2}\hat{G}_3^\beta, G_3^{-a\delta}, \hat{G}_3^{-a\delta}, G_3^\delta G_2^{v'}, \hat{G}_3^\delta\hat{G}_2^{v'}, G_2^\rho, (\hat{G}_1)^{\alpha/\rho}), \\
sk &:= (VK, G^\alpha, G^b = G_1).
\end{aligned}$$

$\mathcal{B}_2$ has G^a since it has a, thus $\mathcal{B}_2$ can generate simulation-type signatures. $\mathcal{B}_2$ gives signatures as follows: For the j-th random message,

Case $j > i$**:** Returns normal-type signature by using $SK = (VK, G_2^\alpha, G_2^b)$.

Case $j < i$**:** Returns simulation-type signature by using SK and G_2^a.

Case $j = i$**:** Embeds the instance as follows. For i-th randomly chosen message m by $\mathcal{B}_2$, $\mathcal{B}_2$ implicitly sets $r_1 := y, r_2 := x$ and computes $\sigma_4 := G^{br_2} = G_1^x$, $\sigma_5 := G^{r_1} = G_2^y$. $\mathcal{B}_2$ can compute $\sigma_0 := (\hat{G}_2^y)^{m\chi_1+\chi_2} = (\hat{U}^m\hat{H})^{r_1}$. Next, in order to compute V^r and $(V')^r$, $\mathcal{B}_2$ computes $(G_3^{r_1+r_2})^{-a\delta}$ as $Z^{-a\delta}$. If $Z = G_3^{x+y}$, then this will be correct. If $Z = G_3^\zeta$ for $\zeta \leftarrow \mathbb{Z}_p$, then we let $G^\gamma := G_3^{\delta(\zeta-(x+y))}$ and this will be a simulation-type signature. $\mathcal{B}_2$ chooses $s \leftarrow \mathbb{Z}_p$ and implicitly sets $G^{-z} := G_2^{-v'r_2+s}$. These value are not computable but $\mathcal{B}_2$ can compute $G^{zb} = G_1^{xv'-s}$. $\sigma_2 := (G_2^y)^{v'}Z^\delta G_2^s = G_2^{r_1v'+r_2v'}Z^\delta G_2^{s-r_2v'} = G_2^{rv'}Z^\delta G^{-z}$. $\mathcal{B}_2$ generates a signature as follows:

$$\sigma_0 := (G_2^y)^{m\chi_1+\chi_2} \qquad \sigma_1 := G_2^\alpha Z^{-a\delta} \qquad \sigma_2 := (G_2^y)^{v'}Z^\delta G_2^s$$
$$\sigma_3 := (G_1^x)^{v'}G_1^{-s} \qquad \sigma_4 := G_1^x \qquad \sigma_5 := G_2^y$$

$\mathcal{B}_2$ can generate σ_0 correctly since $\mathcal{B}_2$ set $\xi m + \beta = 0$.

- If $Z = G_3^{x+y} \in \mathbb{G}_1$, the above signature is a normal-type with $Z = G_3^r$, $\sigma_1 = G_2^\alpha G_3^{-a\delta r} = G_2^\alpha V^r$, and $\sigma_2 = (G_2^{v'}G_3^\delta)^r G^{-z} = (V')^r G^{-z}$.
- If $Z \leftarrow \mathbb{G}_1$, the above signature is a simulation-type since $Z = G_3^z$ for some $z \leftarrow \mathbb{Z}_p$, $\sigma_1 = G_2^\alpha G_3^{-a\delta r}G_3^{-a\delta\zeta}$ $G_3^{a\delta r} = G_2^\alpha V^r G_3^{-a\delta(\zeta-(x+y))} = G^\alpha V^r G^{-a\gamma}$ since $G_3^{\delta(\zeta-(x+y))} = G^\gamma$, and $\sigma_2 = G_2^{rv'}G_3^{r\delta}G_3^{\delta(\zeta-(x+y))}G^{-z}$ $= (V')^r G^\gamma G^{-z}$.

That is, if $Z = G_3^{x+y}$ (linear), then $\mathcal{A}$ is in $p_{i-1}^{\text{norm}}(\lambda)$, otherwise $\mathcal{A}$ is in $p_i^{\text{norm}}(\lambda)$. For all messages, $\mathcal{B}_2$ can return $\mu(M_i) = m_i$.

At some point, $\mathcal{A}$ outputs forgery $(\sigma_1^*, \ldots, \sigma_7^*, \sigma_0^*)$ and message $F(m) = (\hat{Q}_1, \hat{Q}_2, \hat{Q}_0) = (\hat{F}_2^m, \hat{F}_3^m, \hat{U}^m)$. $\mathcal{B}_2$ outputs 1 if and only if

$$e(G_1, \sigma_0) \cdot e(\sigma_6, \hat{Q}_2^\xi \hat{G}_3^\beta) = e((\sigma_1 G_2^{-\alpha a_1})^{1/(-a\delta)}, (\hat{Q}_1^{1/\varphi})^\xi \hat{G}_1^\beta) \cdot e(\sigma_7, (\hat{Q}_1^{1/\varphi})^{\chi_1}\hat{G}_1^{\chi_2}).$$

By the lemma, there exist $m^*, r_1^*, r_2^*, \gamma^*, r^* = r_1^* + r_2^*$ such that $\sigma_0 = (\hat{U}^{m^*}\hat{H})^{r_1^*}$, $\sigma_1 = G_2^\alpha V^{r^*}G_2^{-a\gamma^*}$, $\sigma_4 = G_1^{r_2^*}$, $\sigma_5 = G_2^{r_1^*}$, $\hat{Q}_1 = (\hat{G}_1^\varphi)^{m^*}$, $\hat{Q}_2 = \hat{G}_3^{m^*}$. Since $\sigma_0 = (\hat{G}_2^{m\chi_1+\chi_2}\hat{G}_3^{\xi x+\beta})^{r_1^*}$, $\sigma_1 = G_2^\alpha G_3{}^{-a\delta r^*}G_2^{-a\gamma^*}$, $\sigma_4 = G_1{}^{r_2^*}$, $\sigma_5 = G_2^{r_1^*}$, we have

$$\begin{aligned}
e(G_1, \sigma_0) \cdot e(\sigma_4, \hat{Q}_2^\xi \hat{G}_3^\beta) &= e(G_1, (\hat{G}_2^{m^*\chi_1+\chi_2}\hat{G}_3^{\xi m^*+\beta})^{r_1^*}) \cdot e(G_1^{r_2^*}, (\hat{G}_3^{m^*})^\xi \hat{G}_3^\beta) \\
&= e(G_1, \hat{G}_2)^{(m^*\chi_1+\chi_2)r_1^*} e(G_1, \hat{G}_3)^{(\xi m^*+\beta)r_1^*} e(G_1, \hat{G}_3)^{(\xi m^*+\beta)r_2^*}
\end{aligned}$$

and

$$\begin{aligned}
&e((\sigma_1 G_2^{-\alpha})^{1/(-a\delta)}, (\hat{Q}_1^{1/\varphi})^\xi \hat{G}_1^\beta) \cdot e(\sigma_5, (\hat{Q}_1^{1/\varphi})^{\chi_1}\hat{G}_1^{\chi_2}) \\
&\quad = e(G_3^{r^*}G_2^{\gamma^*/\delta}, \hat{G}_1^{\xi m^*+\beta}) \cdot e(G_2^{r_1^*}, \hat{G}_1^{m^*\chi_1+\chi_2}) \\
&\quad = e(G_3, \hat{G}_1)^{(\xi m^*+\beta)r^*} e(G_2, \hat{G}_1)^{\gamma^*/\delta(\xi m^*+\beta)} e(G_2, \hat{G}_1)^{(m^*\chi_1+\chi_2)r_1^*}.
\end{aligned}$$

A simplified equation is $1 = e(G_2, \hat{G}_1)^{\gamma^*/\delta(\xi m^*+\beta)}$.

Thus, the difference of $\mathcal{A}$'s advantage in two games gives the advantage of $\mathcal{B}_2$ in solving the XDLIN$_1$ problem as stated.

Proof. (of Lemma 12) Observe that, in $p_q^{\text{norm}}(\lambda)$, $\mathcal{A}$ is given simulation-type signatures only. We show that if $\mathcal{A}$ outputs a normal-type forgery in $p_q^{\text{norm}}(\lambda)$ then we can construct algorithm $\mathcal{B}_3$ that solves the co-CDH problem.

$\mathcal{B}_3$ is given instance $(\Lambda, G, \hat{G}, G^x, G^y, \hat{G}^x, \hat{G}^y)$ of the co-CDH problem. $\mathcal{B}_3$ generates the verification key as follows: Selects exponents $b, v, v', u, h, f_2, f_3 \leftarrow \mathbb{Z}_p$ and $\rho' \leftarrow \mathbb{Z}_p^*$, computes $\hat{G}^b := \hat{G}^b$, $G^a := G^y$, $\hat{G}^y := \hat{G}^a$, $\hat{G}^{ba} := (\hat{G}^y)^b$, $V := G^v$, $\hat{V} := \hat{G}^v$, $V' := G^{v'}$, $\hat{V}' := \hat{G}^{v'}$, $u := G^u$, $\hat{U} := \hat{G}^u$, $H := G^h$, $\hat{H} := \hat{G}^h$, $F_2 := G^{f_2}$, $\hat{F}_2 := \hat{G}^{f_2}$, $F_3 := G^{f_3}$, $\hat{F}_3 := \hat{G}^{f_3}$, $G^\rho := (G^x)^{\rho'}$, $\hat{G}^{\alpha b/\rho} := (\hat{G}^y)^{b/\rho'}$ where $\rho = \rho' x$ (it implicitly holds $\alpha = xy$ though $\mathcal{B}_3$ does not have α), $R := V(G^y)^{v'}$, R^b, $\hat{R} := \hat{V}(\hat{G}^y)^{v'}$, and $\hat{R}^b$, and sets

$$\begin{aligned}
gk &:= (\Lambda, G, \hat{G}, F_2, \hat{F}_2, F_3, \hat{F}_3, U, \hat{U}), \\
vk &:= (\hat{G}^b, \hat{G}^y, (\hat{G}^y)^b, V(G^y)^{v'}, V^b(G^y)^{bv'}, \hat{V}(\hat{G}^y)^{v'}, \hat{V}^b(\hat{G}^y)^{bv'}, H, \hat{H}, V, \hat{V}, V', \hat{V}', (G^x)^{\rho'}, (\hat{G}^y)^{b/\rho'}).
\end{aligned}$$

Note that $\mathcal{B}_3$ does not have $G^{\alpha} = G^{xy}$, so $\mathcal{B}_3$ cannot compute normal-type signature. $\mathcal{B}_3$ outputs simulation-type signatures for i-th random message m as follows:

Selects $r_1, r_2, z, \gamma' \leftarrow \mathbb{Z}_p$, sets $r := r_1 + r_2$ (we want to set $\gamma := x + \gamma'$), and computes:

$$\sigma_1 := (G^y)^{-\gamma'} \cdot V^r = (G^{\alpha} V^r) \cdot G^{-a\gamma} \; (a = y, xy = \alpha)$$

$$\sigma_2 := G^{\gamma'} G^x (V')^r G^{-z} = ((V')^r G^{-z}) \cdot G^{\gamma}$$

$$\sigma_3 := (G^b)^z \qquad \sigma_4 := G^{r_2 b} \qquad \sigma_5 := G^{r_1} \qquad \sigma_0 := (\hat{U}^m \hat{H})^{r_1}$$

Outputs signature $(\sigma_0, \sigma_1, \ldots, \sigma_5)$ for $F(m) = (\hat{M}_1, \hat{M}_2, \hat{M}_0) = (\hat{F}_2^m, \hat{F}_3^m, \hat{U}^m)$.

At some point, $\mathcal{A}$ outputs a normal-type forgery, $\sigma_1^* = G^{\alpha} V^{r^*}$, $\sigma_2^* = (V')^{r^*} G^{-z^*}$, $\sigma_3^* = (G^b)^{z^*}$, $\sigma_4^* = G^{r_2^* b}$, $\sigma_5^* = G^{r_1^*}$, and $\sigma_0^* = (\hat{U}^{m^*} \hat{H})^{r_1^*}$, for some $r_1^*, r_2^*, z^*, \in \mathbb{Z}_p$ for message $F(m^*) = (\hat{F}_2^{m^*}, \hat{F}_3^{m^*}, \hat{U}^{m^*})$.

By using these values, $\mathcal{B}_3$ can compute $G^{r_2^*} = (\sigma_4^*)^{1/b}$, $G^{r_1^*} = \sigma_5^*$, $G^{z^*} = (\sigma_3^*)^{1/b}$, $V^{r^*} = (G^{r_1^*} \cdot G^{r_2^*})^v$ since $V = G^v$. Thus, $\mathcal{B}_3$ can compute $\sigma_1^*/V^{r^*} = G^{\alpha} = G^{xy}$. That is, $\mathcal{B}_3$ can solve the co-CDH problem and it holds $p_q^{\mathsf{norm}}(\lambda) \leq \mathsf{Adv}_{\mathcal{G},\mathcal{B}_3}^{\mathsf{co\text{-}cdh}}(\lambda)$ as claimed.

6.5 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 3, 9, and 10, we obtain the following theorem.

Theorem 11. *SIG2 is a structure-preserving signature scheme that is unforgeable against adaptive chosen message attacks if SXDH and XDLIN$_1$ hold for $\mathcal{G}$.*

Table 2 summarises the efficiency of SIG2 for both unileteral and bileteral 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 6.3.

Table 2: Efficiency of SIG2 and comparison to other schemes with constant-size signatures. Upper half is for unilateral messages and the lower half is for bilateral messages. Notation (x, y) represents x elements in $\mathbb{G}_1$ and y in $\mathbb{G}_2$.

Scheme	$\lvert msg \rvert$	$\lvert gk \rvert + \lvert vk \rvert$	$\lvert \sigma \rvert$	#(PPE)	Assumptions
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	$(k_1, 0)$	$(7, k_1 + 13)$	$(7, 4)$	4	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)	$(k_2 + 8, k_1 + 14)$	$(8, 6)$	5	SXDH, XDLIN$_1$

7 Other Instantiation

7.1 Partial one-time signature scheme based on DLIN

This section presents partial one-time signature scheme over Type-I bilinear groups. It is coupled with Setup in Section 5.1, and can further be combined with XRMA-secure rSIG in Section 5.3 to instantiate SIG2 over Type-I groups.

[Scheme POS]

- POS.Key(gk): Parse $gk = (\Lambda, G, C, F, U_1, U_2)$ Choose τ_1, τ_2 randomly from $\mathbb{Z}_p^*$ and compute $G_t := G^{\tau_1}$ and $H_t := G^{\tau_2}$. Choose w_r, μ_s randomly from $\mathbb{Z}_p^*$ and compute $G_r := U_1^{w_r}$, and $H_s := U_2^{\mu_s}$. For $i = 1, \ldots, k$, uniformly choose χ_i, γ_i, δ_i from $\mathbb{Z}_p$ and compute

$$G_i := U_1^{\chi_i} G_r^{\gamma_i}, \qquad H_i := U_2^{\chi_i} H_s^{\delta_i}. \tag{9}$$

  Output $pk := (G_r, H_s, G_1, \ldots, G_k, H_1, \ldots, H_k) \in \mathbb{G}^{2k+2}$ and $sk := (\tau_1, \tau_2, \chi_1, \gamma_1, \delta_1, \ldots, \chi_k, \gamma_k, \delta_k, w_r, \mu_s)$.

- POS.Update(*mode*): Take (G, C, F, U_1, U_2) from gk. Choose $(b_1, b_2) \leftarrow \mathbb{Z}_p$ and output $opk := (U_1^{b_1}, U_2^{b_2}) = (B_1, B_2) \in \mathbb{G}^2$ if *mode* = normal or $opk := (C^{b_1}, C^{b_2}, F^{b_1}, F^{b_2}, U_1^{b_1}, U_2^{b_2}) = (C_1, C_2, A_1, A_2, B_1, B_2) \in \mathbb{G}^6$ if *mode* = extended. Also output $osk := (b_1, b_2)$.
- POS.Sign(sk, msg, osk): Parse msg into $(M_1, \ldots, M_k) \in \mathbb{G}^k$, and other inputs accordingly. Take b_1, b_2, τ_1, τ_2 and w_r, μ_s from osk and sk, respectively. Choose ζ randomly from $\mathbb{Z}_p$ and compute

$$\rho := (b_1\, \tau_1 - \zeta)\, / w_r \bmod p \quad \text{and} \quad \varphi := (b_2\, \tau_2 - \zeta)\, / \mu_s \bmod p. \tag{10}$$

  Then compute

$$Z := G^{\zeta} \prod_{i=1}^{k} M_i^{-\chi_i} \quad \text{and} \quad R := G^{\rho} \prod_{i=1}^{k} M_i^{-\gamma_i} \quad \text{and} \quad S := G^{\varphi} \prod_{i=1}^{k} M_i^{-\delta_i}. \tag{11}$$

  Output $\sigma := (Z, R, S) \in \mathbb{G}^3$ as a signature.
- POS.Vrf(pk, σ, msg, opk): Parse σ as $(Z, R, S) \in \mathbb{G}^3$, msg as $(M_1, \ldots, M_k) \in \mathbb{G}^k$, and opk as $(C_1, C_2, A_1, A_2, B_1, B_2)$ or (B_1, B_2) depending on *mode*. Return 1 if

$$e(B_1, G_t) = e(U_1, Z)\; e(G_r, R)\; \prod_{i=1}^{k} e(G_i, M_i) \tag{12}$$

$$e(B_2, H_t) = e(U_2, Z)\; e(H_s, S)\; \prod_{i=1}^{k} e(H_i, M_i) \tag{13}$$

  holds. Return 0, otherwise.

The scheme is correct as the following relation holds for the verification equation and the computed signatures.

$$\begin{aligned} e(U_1, Z)\; e(G_r, R)\; \prod_{i=1}^{k} e(G_i, M_i) &= e(U_1, G^{\zeta} \prod_{i=1}^{k} M_i^{-\chi_i})\; e(G_r, G^{\rho} \prod_{i=1}^{k} M_i^{-\gamma_i})\; \prod_{i=1}^{k} e(U_1^{\chi_i} G_r^{\gamma_i}, M_i) \\ &= e(U_1, G^{\zeta})\; e(U_1^{w_r}, G^{\rho}) \\ &= e(U_1^{\zeta + w_r \rho}, G) \\ &= e(B_1, G_t) \end{aligned}$$

and

$$\begin{aligned} e(U_2, Z)\; e(H_s, S)\; \prod_{i=1}^{k} e(H_i, M_i) &= e(U_2, G^{\zeta} \prod_{i=1}^{k} M_i^{-\chi_i})\; e(H_r, G^{\varphi} \prod_{i=1}^{k} M_i^{-\delta_i})\; \prod_{i=1}^{k} e(U_2^{\chi_i} H_s^{\delta_i}, M_i) \\ &= e(U_2, G^{\zeta})\; e(U_2^{\mu_s}, G^{\varphi}) \\ &= e(U_2^{\zeta + \mu_s \varphi}, G) \\ &= e(B_2, H_t) \end{aligned}$$

Theorem 12. *POS is unforgeable against one-time adaptive chosen message attacks (OT-CMA) if the SDP assumption holds. In particular,* $\mathrm{Adv}^{\textit{ot-cma}}_{\textit{POS},\mathcal{A}} \leq \mathrm{Adv}^{\textit{sdp}}_{\mathcal{G},\mathcal{B}} + 1/p$.

Proof. The following proof is almost a copy of that for Theorem 8 except for trivial modifications.

Given successful forger $\mathcal{A}$ against POS as a black-box, we construct $\mathcal{B}$ that is successful in breaking SDP. We consider the case *mode* = extended in the following. The other case, *mode* = normal can be automatically obtained by dropping (C_1, C_2, A_1, A_2).

Given instance $I_{\mathsf{sdp}} = (\Lambda, G_z, G_r, H_z, H_s)$ of SDP, algorithm $\mathcal{B}$ simulates the attack game against POS as follows. It first build gk. Choose randomly G from $\mathbb{G}^*$ and $c,\ f$ from $\mathbb{Z}_p^*$, and compute $C := G^c$, $F := G^f$, then set $U_1 := G_z$, $U_2 := G_z$, and $gk := (\Lambda, G, C, F, U_1, U_2)$. This yields a gk from the same distribution as produced by Setup. Next $\mathcal{B}$ simulates POS.Key by following the original prescription except that Λ and G_r, H_s are taken from I_{sdp}. Note that w_r, μ_s, i.e., the discrete-logs of G_r, H_s with respect to base U_1, U_2, are not known to the simulator, but they are not needed in simulating. On receiving one-time key query, algorithm $\mathcal{B}$ simulates $\mathsf{POS.Update}$ by returning $B_1 := U_1^{\zeta} G_r^{\rho},\ B_2 := U_2^{\zeta} H_s^{\varphi},\ C_1 := B_1^c,\ C_2 := B_2^c,\ \ A_1 := B_1^f,\ \ A_2 := B_2^f$ for $\zeta, \rho, \varphi \leftarrow \mathbb{Z}_p$, and $c,\ f$ generated in Setup.

On receiving signing query $msg^{(j)}$ from $\mathcal{A}$, algorithm $\mathcal{B}$ simulates $\mathcal{O}sig$ by simulating $\mathsf{POS.Sign}$ without having w_r and μ_s. It is done by using ζ, ρ, and φ used in $\mathsf{POS.Update}$ instead of computing $\rho,\ \varphi$ from (b_1, b_2). For each signing, transcript (opk, σ, msg) is recorded. When $\mathcal{A}$ outputs a forgery $(opk^\dagger, \sigma^\dagger, msg^\dagger)$, algorithm $\mathcal{B}$ searches the records for (opk, σ, msg) such that $opk^\dagger = opk$ and $msg^\dagger \neq msg$. If no such entry exists, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ computes

$$Z^\star := \frac{Z^\dagger}{Z} \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\chi_i}, \quad \text{and} \quad R^\star := \frac{R^\dagger}{R} \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\gamma_i}, \quad \text{and} \quad S^\star := \frac{S^\dagger}{S} \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\delta_i}, \tag{14}$$

where $(Z, R, S, M_1, \ldots, M_k)$ and its dagger counterpart are taken from (σ, msg) and $(\sigma^\dagger, msg^\dagger)$, respectively. $\mathcal{B}$ finally outputs $(Z^\star, R^\star, S^\star)$. This completes the description of $\mathcal{B}$.

We first claim that the simulation by $\mathcal{B}$ is perfect; the parameters and keys generated in Setup and $\mathsf{POS.Key}$ due to the uniform choice of $I_{\mathsf{sdp}} = (\Lambda, G_z, G_r, H_z, H_s)$, and the distribution of $(b_1, b_2, \zeta, \rho, \varphi)$ is uniform over $\mathbb{Z}_p^5$ under constraint $b_1 = (\zeta + \rho w_r)/\tau_1$ and $b_2 = (\zeta + \varphi \mu_s)/\tau_2$ as well as the original procedure. Accordingly, $\mathcal{A}$ outputs successful forgery with noticeable probability and $\mathcal{B}$ finds a corresponding record (opk, σ, msg).

We next claim that each χ_i is independent of the view of $\mathcal{A}$. Concretely, we show that, if coins $\chi_1, \ldots, \chi_k$ distribute uniformly over $(\mathbb{Z}_p)^k$, other coins $\gamma_1, \ldots, \gamma_k$, $\delta_1, \ldots, \delta_k, (\zeta^{(1)}, \rho^{(1)}, \varphi^{(1)}), \ldots, (\zeta^{(q_s)}, \rho^{(q_s)}, \varphi^{(q_s)})$ distribute uniformly as well retaining consistency with the view of $\mathcal{A}$. Observe that the view of $\mathcal{A}$ making q_s signing queries consists of independent group elements $(G, C, F, U_1, U_2, G_r, H_s, G_t, H_t, G_1, H_1, \ldots, G_k, H_k)$ and $(B_1^{(j)}, B_2^{(j)}, Z^{(j)}, M_1^{(j)}, \ldots, M_k^{(j)})$ for $j = 1, \ldots, q_s$. We represent the view by the discrete-logarithms of these group elements with respect to base G. Namely, the view is $(1, c, f, u_1, u_2, w_r, \mu_s, \tau_1, \tau_2, w_1, \mu_1, \ldots, w_k, \mu_k)$ and $(b_1^{(j)}, b_2^{(j)}, z^{(j)}, m_1^{(j)}, \ldots, m_k^{(j)})$ for $j = 1, \ldots, q_s$. To be consistent, the view and the coins satisfy relations

$$w_i = \chi_i + w_r \gamma_i, \quad \mu_i = \chi_i + \mu_s \delta_i \quad \text{for } i = 1, \ldots, k, \text{ and} \tag{15}$$

$$b_1^{(j)} = (\zeta^{(j)} + w_r \rho^{(j)})/\tau_1, \quad b_2^{(j)} = (\zeta^{(j)} + \mu_s \varphi^{(j)})/\tau_2, \quad \text{and} \tag{16}$$

$$z^{(j)} = \zeta^{(j)} - \sum_{i=1}^{k} m_i^{(j)} \chi_i \quad \text{for } j = 1, \ldots, q_s. \tag{17}$$

The equation (15), (16) and (17) corresponding in (9), (10) and the first one in (11) respectively.

Consider χ_ℓ for fixed ℓ in $\{1, \ldots, k\}$. For every value of χ_ℓ, the linear equations in (15) determine $\gamma_1, \ldots, \gamma_k$ and $\delta_1, \ldots, \delta_k$. Similarly, if $m_\ell \neq 0$, equations in (16), and (17) determine $\zeta^{(j)}$, $\rho^{(j)}$, and $\varphi^{(j)}$ for $j = 1, \ldots, q_s$. If $m_\ell = 0$, then χ_ℓ is independent of $\zeta^{(j)}, \rho^{(j)}$, and $\varphi^{(j)}$ for $j = 1, \ldots, q_s$. The above holds for every ℓ in $\{1, \ldots, k\}$. Thus, if $\chi_1, \ldots, \chi_k$ distributes uniformly over $(\mathbb{Z}_p)^k$, then other coins distribute uniformly as well retaining consistency with the view of $\mathcal{A}$.

Finally, we claim that $(Z^\star, R^\star, S^\star)$ is a valid solution to the given instance of SDP. Since both forged and recorded signatures fulfill equation (12) (13), dividing the equations results in

$$1 = e\left(U_1, \frac{Z^\dagger}{Z}\right)\ e\left(G_r, \frac{R^\dagger}{R}\right) \prod_{i=1}^{k} e\left(U_1^{\chi_i} G_r^{\gamma_i}, \frac{M_i^\dagger}{M_i}\right) \tag{18}$$

$$= e\left(U_1, \frac{Z^\dagger}{Z} \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\chi_i}\right)\ e\left(G_r, \frac{R^\dagger}{R} \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\gamma_i}\right) \tag{19}$$

$$= e\left(U_1, Z^\star\right)\ e\left(G_r, R^\star\right), \tag{20}$$

and

$$1 = e\left(U_2, \frac{Z^\dagger}{Z}\right)\ e\left(H_s, \frac{S^\dagger}{S}\right) \prod_{i=1}^{k} e\left(U_2^{\chi_i} H_s^{\delta_i}, \frac{M_i^\dagger}{M_i}\right) \tag{21}$$

$$= e\left(U_2, \frac{Z^\dagger}{Z} \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\chi_i}\right)\ e\left(H_s, \frac{S^\dagger}{S} \prod_{i=1}^{k} \left(\frac{M_i^\dagger}{M_i}\right)^{\delta_i}\right) \tag{22}$$

$$= e\left(U_2, Z^\star\right)\ e\left(H_s, S^\star\right). \tag{23}$$

What remains is to prove that $Z^\star \neq 1$. Since $msg^\dagger \neq msg^{(j)}$, there exists $\ell \in \{1, \ldots, k\}$ such that $\frac{M_\ell^\dagger}{M_\ell} \neq 1$. As already proven, χ_ℓ is independent of the view of $\mathcal{A}$. Thus $\left(\frac{M_\ell^\dagger}{M_\ell}\right)^{\chi_\ell}$ distributes uniformly over $\mathbb{G}$ and $Z^\star = 1$ holds only if $Z^\dagger = Z \prod (M_i^\dagger / M_i)^{-\chi_i}$, which happens only with probability $1/p$ over the choice of χ_ℓ. We thus have $\mathrm{Adv}^{\mathsf{ot\text{-}cma}}_{\mathsf{TOS},\mathcal{A}} \leq \mathrm{Adv}^{\mathsf{sdp}}_{\mathcal{G},\mathcal{B}} + 1/p$. This concludes poof of Theorem 12.

Finally, from Theorem 12 and Lemma 2, we have $\mathrm{Adv}^{\mathsf{ot\text{-}cma}}_{\mathsf{POS},\mathcal{A}} \leq \mathrm{Adv}^{\mathsf{dlin}}_{\mathcal{G},\mathcal{B}} + 1/p$.

8 Applications and Open Questions

Structure-preserving signatures (SPS) have become a mainstay in cryptographic protocol design in recent years. From the many applications that benefit from efficient SPS bases on simple assumptions, we list only a few recent examples. Using our $\mathsf{SIG1}$ scheme from Section 5 both the construction of a group signature scheme with efficient revocation by Libert, Peters and Yung [32] and the construction of compact verifiable shuffles by Chase et al. [14] can be proven purely under the DLIN assumption. All other building blocks already have efficient instantiations based on DLIN.

Hofheinz and Jager [30] construct a structure-preserving one-time signature scheme. They use it to build a tree-based SPS scheme, denoted by tSIG, that is secure against non-adaptive chosen message attacks. Instead, we propose to use our partial one-time scheme to construct tSIG. Since the latter 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 [30], the resulting signature scheme is significantly more efficient than [30] and is a SPS scheme with a tight security reduction to SXDH. One can do the same in Type-I groups by using the partial one-time signature scheme in Section 7.1 whose security tightly reduced to DLIN.

As also shown by [30], SPS schemes allow to implements simulation-sound NIZK proof systems based on the Groth-Sahai proof system. Following the Naor-Yung-Sahai [34, 37] paradigm, one obtains structure-preserving CCA-secure public-key encryption in a modular fashion.

Open Questions. 1) Can we have RMA or XRMA-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 scheme 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 indeed 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 '10*, LNCS, pages 209–237. Springer-Verlag, 2010. (Cited on page 1.)

[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. (Cited on page 1, 24.)

[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. (Cited on page 2.)

[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`. (Cited on page 1, 3, 17, 18, 24.)

[5] M. Abe and M. Ohkubo. A framework for universally composable non-committing blind signatures. *IJACT*, 2(3):229–249, 2012. (Cited on page 1.)

[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 *Lecture Notes in Computer Science*, pages 108–125. Springer, 2009. (Cited on page 1.)

[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. Springer-Verlag, 2003. (Cited on page 1.)

[8] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. IACR e-print 2004/077, 2004. (Cited on page 1.)

[9] M. Bellare and S. Shoup. Two-tier signatures, strongly unforgeable signatures, and fiat-shamir without random oracles. In *Proceedings of the 10th International Conference on Theory and Practice of Public-Key Cryptography - PKC 2007*, volume 4450 of *LNCS*, pages 201–216. Springer-Verlag, 2007. (Cited on page 1, 5.)

[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. (Cited on page 3.)

[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. (Cited on page 1.)

[12] J. Cathalo, B. Libert, and M. Yung. Group encryption: Non-interactive realization in the standard model. In M. Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 179–196. Springer-Verlag, 2009. (Cited on page 4.)

[13] M. Chase and M. Kohlweiss. A domain transformation for structure-preserving signatures on group elements. IACR ePrint Archive, Report 2011/342, 2011. (Cited on page 1, 2.)

[14] 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 *Lecture Notes in Computer Science*, pages 281–300. Springer, 2012. (Cited on page 1, 27.)

[15] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000. (Cited on page 1.)

[16] C. Dwork and M. Naor. An efficient existentially unforgeable signature scheme and its applications. *J. Cryptology*, 11(3):187–208, 1998. (Cited on page 1.)

[17] S. Even, O. Goldreich, and S. Micali. On-line/off-line digital signatures. *J. Cryptology*, 9(1):35–67, 1996. (Cited on page 2, 4.)

[18] M. Fischlin. Round-optimal composable blind signatures in the common reference model. In C. Dwork, editor, *Advances in Cryptology — CRYPTO '06*, volume 4117 of *LNCS*, pages 60–77. Springer-Verlag, 2006. (Cited on page 1.)

[19] G. Fuchsbauer. Commuting signatures and verifiable encryption. In *Advances in Cryptology — Eurocrypt '11*, LNCS, pages 224–245. Springer-Verlag, 2011. (Cited on page 1.)

[20] G. Fuchsbauer and D. Pointcheval. Anonymous proxy signatures. In R. Ostrovsky, R. D. Prisco, and I. Visconti, editors, *SCN*, volume 5229 of *Lecture Notes in Computer Science*, pages 201–217. Springer, 2008. (Cited on page 1.)

[21] 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 *Lecture Notes in Computer Science*, pages 226–247. Springer, 2009. (Cited on page 1.)

[22] G. Fuchsbauer and D. Vergnaud. Fair blind signatures without random oracles. In D. J. Bernstein and T. Lange, editors, *AFRICACRYPT*, volume 6055 of *Lecture Notes in Computer Science*, pages 16–33. Springer, 2010. (Cited on page 1.)

[23] S. D. Galbraith, K. G. Peterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 12008. (Cited on page 2.)

[24] 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. (Cited on page 1, 2.)

[25] M. Green and S. Hohenberger. Universally composable adaptive oblivious transfer. In J. Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 179–197. Springer-Verlag, 2008. Preliminary version: IACR ePrint Archive 2008/163. (Cited on page 1.)

[26] M. Green and S. Hohenberger. Practical adaptive oblivious transfer from simple assumptions. In Y. Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 347–363. Springer, 2011. (Cited on page 1.)

[27] 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. (Cited on page 1, 2.)

[28] J. Groth. Homomorphic trapdoor commitments to group elements. IACR ePrint Archive, Report 2009/007, January 2009. Update version available from the author's homepage. (Cited on page 8.)

[29] 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. Full version available: IACR ePrint Archive 2007/155. (Cited on page 1.)

[30] D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. In *CRYPTO*. Springer, 2012. (Cited on page 1, 2, 27.)

[31] 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. (Cited on page 1.)

[32] B. Libert, T. Peters, and M. Yung. Scalable group signatures with revocation. In *Advances in Cryptology – Eurocrypt 2012*, LNCS. Springer-Verlag, 2012. (Cited on page 27.)

[33] Y. Lindell. A simpler construction of cca2-secure public-keyencryption under general assumptions. *J. Cryptology*, 19(3):359–377, 2006. (Cited on page 1.)

[34] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing*, pages 427–437, 1990. (Cited on page 1, 27.)

[35] S. C. Ramanna, S. Chatterjee, and P. Sarkar. Variants of waters' dual-system primitives using asymmetric pairings. *IACR Cryptology ePrint Archive*, 2012:24, 2012. (Cited on page 20.)

[36] 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 *Lecture Notes in Computer Science*, pages 17–34. Springer, 2009. (Cited on page 1.)

[37] A. Sahai. Non-malleable non-interactive zero-knowledge and chosen-ciphertext security. In *Proceedings of the 40th IEEE Annual Symposium on Foundations of Computer Science*, pages 543–553, 1999. (Cited on page 1, 27.)

[38] 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 *Lecture Notes in Computer Science*, pages 566–598. Springer, 2001. (Cited on page 1.)

[39] 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. (Cited on page 3.)

[40] 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. (Cited on page 2, 11.)