Full version of [10].

Fully Structure-Preserving Signatures and Shrinking Commitments

Masayuki Abe[1] Markulf Kohlweiss[2] Miyako Ohkubo[3] Mehdi Tibouchi[1]

[1] Information Sharing Platform Laboratories, NTT Corporation, Japan
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585 Japan
{abe.masayuki,tibouch.mehdi}@lab.ntt.co.jp

[2] Microsoft Research, Cambridge, Microsoft, UK
markulf@microsoft.com

[3] Security Fundamentals Laboratory, NSR, NICT, Japan
m.ohkubo@nict.or.jp

Abstract

Structure-preserving signatures are schemes in which public keys, messages, and signatures are all collections of source group elements of some bilinear groups. In this paper, we introduce fully structure-preserving signature schemes, with the additional requirement that even secret keys should be group elements. This new type of structure-preserving signatures allows for efficient non-interactive proofs of knowledge of the secret key and is useful in designing cryptographic protocols with strong security guarantees based on the simulation paradigm where the simulator has to extract the secret keys on-line.

To gain efficiency, we construct shrinking structure-preserving trapdoor commitments. This is by itself an important primitive and of independent interest as it appears to contradict a known impossibility result. We argue that a relaxed binding property lets us circumvent the impossibility result while still retaining the usefulness of the primitive in important applications as mentioned above.

Keywords: Structure-preserving signatures, Secret key extraction, Structure-preserving commitments

Contents

1 Introduction **1**

2 Preliminaries **3**
2.1 Notations . . . 3
2.2 Bilinear Groups . . . 3
2.3 Digital Signatures . . . 3

3 Building Blocks **5**
3.1 Common Setup Function . . . 5
3.2 Partially One-time Signatures . . . 5
3.3 xRMA-secure Fully Structure-Preserving Signature Scheme . . . 6

4 Trapdoor Commitment Schemes **8**
4.1 Definitions . . . 8
4.2 γ-Binding Commitment Scheme . . . 9
4.3 Structure-Preserving Shrinking Trapdoor Commitment Scheme . . . 11

5 Fully Structure-Preserving Signatures **13**
5.1 Warm-Up . . . 13
5.2 Main Construction . . . 16
5.3 Efficiency . . . 18
5.4 Lower Bound on Signature Size and Verification key size . . . 21

6 Variations **22**
6.1 xSIG + POS . . . 23
6.2 xSIG + SPS . . . 23
6.3 xSIG + TCγ + OTS . . . 24
6.4 xSIG + TCγ + SPS . . . 24

7 Conclusion **24**

1 Introduction

In pairing-based cryptography, cryptographic primitives are often designed to have algorithms in which messages and public materials consist only of source group elements and correctness can be proved using pairing-product equations to allow smooth coupling with other primitives. This interest in so called structure-preserving primitives [3] led to the study of algebraic algorithms with many positive but also negative results, e.g., [5, 6, 19, 2, 36, 8, 1, 7, 25, 12].

In structure-preserving signature schemes, all components but secret keys are group elements. This raises a natural question: "*Can secret keys consist entirely of source group elements as well*?" Having messages and signatures in the same group prevents us from relying on the one-wayness of exponentiation (or of the isomorphism from one source group to the other in the case of asymmetric bilinear groups) to blend messages into signatures, and it is a major difficulty in designing structure-preserving signatures. In existing schemes, this is overcome by having secret keys in the exponent. Thus, it is quite unclear how messages and secret keys can blend into signatures if even secret keys are group elements.

Besides the above question being a fascinating fundamental question in its own right, it is connected to practical protocol design since group secret keys combined with the Groth-Sahai proof system [33] allow straight-line (i.e., no rewinding) extraction of the secret keys when necessary. While there are solutions in the random oracle model, e.g. [28, 20], secret key extraction without random oracles is currently prohibitively expensive. Meiklejohn [37] demonstrates how to extract a secret key in the exponent using the Groth-Sahai proofs. It requires bit-by-bit decomposition of secret x, and the proof consists of $20 \log_2 x + 18$ group elements. For instance, applying it to a structure-preserving signature scheme [2] whose secret key consists of $4 + 2\ell$ scalar values for signing messages of ℓ group elements, proving secret keys for signing 10 group elements at 128-bit security, requires more than 61,000 group elements.

Our contribution. This paper contains one main result and one important by-product that is of independent interest. First, we present a *fully structure-preserving signature (FSPS)* scheme all of whose components, including secret keys, consist of source group elements of bilinear groups. This result demonstrates that the paradigm of structure-preserving cryptography can be extended to cover private key material. The security against adaptive chosen message attacks is proved based on static (i.e., not q-type) assumptions. Its secret key consists only of four group elements, and a witness indistinguishable proof of knowledge about the secret key consists of 18 group elements (see Section 5.3). These are huge savings compared to the current solution mentioned above.

A price to pay is the signature size $O(\sqrt{\ell})$ for messages consisting of ℓ elements. A precise performance analysis shows that this remains relatively practical for short messages, e.g., a signature consists of 23 elements for messages of 9 elements (see Table 1). We show a non-trivial trade-off between the size of verification keys and signatures that implies that an order of $\sqrt{\ell}$ elements in signatures is inherent, at least for the type of modular constructions considered in this paper.

The investigation of efficient instantiations lead us to our second contribution: a *shrinking structure-preserving trapdoor commitment scheme* (SPTC). We present an SPTC scheme that produces constant-size commitments consisting of a single group element regardless of the message size. In addition to being an important primitive in itself, it is a remarkable construction in light of the well-known impossibility result [9] stating that SPTC schemes that yield shorter commitments than messages cannot be binding (or collision resistant, equivalently). We get around the impossibility by making two exclusive relaxations in the requirements. One is to weaken the security from collision resistance to what we call chosen-message target collision resistance (CM-TCR). In the proof of impossibility in [9], it is essential that the adversary finding a collision knows the randomness used to create the commitment. In CMTCR, it is still the adversary who chooses the messages to commit to, but it is the challenger who creates the target commitment from the given message. Therefore the random coins used for the target are hidden from the adversary. We show that CMTCR is sufficient to construct secure signature schemes that achieve existential unforgeability against adaptive chosen message attacks [31] in combination with a weak signature scheme that is secure only against extended random message attacks [2].

Despite the first relaxation, it is still not easy to achieve CMTCR security. As a stepping stone we make the second relaxation and allow the commitment function to take exponents as input while mapping it to group elements for verification. The resulting scheme is no longer structure-preserving but does preserve the group structure with respect to verification. As we require a bijection γ between the message

space for commitment and that for verification, we call such schemes γ-binding commitments. Finding a concrete construction satisfying the shrinking property is another challenge. There are commitment schemes whose messages can be scalar values in a bilinear group setting, e.g. [40, 23, 3, 33, 35], but none are γ-binding and shrinking. We present a concrete scheme whose commitment consists of a single group element and achieves collision resistance. We then use the shrinking γ-binding commitments to compress verification keys of a (not necessarily fully) structure-preserving partially one-time signature scheme (POS), and prove that it constitutes a shrinking SPTC with the CMTCR property. Thus, we argue "group to group commitments do shrink, sometimes."

Related work. At least one FSPS scheme already exists [2] but with constraints on both security and usability. Namely, it only meets the weak security guarantee (unforgeable against extended random message attacks), and the signing function takes messages of the form (G^m, F^m, U^m) that essentially requires knowledge of m [34, 15]. Nevertheless, the UF-XRMA-secure FSPS scheme is a reasonable starting point and we overcome its shortcomings by combining it with structure-preserving trapdoor commitments or one-time structure-preserving signatures.

Regarding SPTC, the study by Abe et al.[9], is an important piece of context. It presents a concrete attack against all shrinking SPTC schemes. In fact, all existing SPTCs, e.g. [3], are rather expanding. The way we circumvent the impossibility, namely the γ-binding property, resembles the F-unforgeability notion [14] for signature schemes.

The use of trapdoor commitments and chameleon hashing has also been explored in the construction of on-line off-line signatures [27, 22]. The work of Even, Goldreich, and Micali already formed the basis for the generic construction of SPS [2]. In addition, Catalano et al.[22], and Mohassel [39] observed an interesting relationship between one-time signatures and chameleon hashing. In [4], a security notion for hash functions that can be seen as a special case of CMTCR is introduced.

We discuss potential applications of FSPS in the context of efficient secret key extraction from concrete to more high-level as follows.

Public-key infrastructure. On the very applied side, the question is connected with the timely problem of public-key infrastructures. Few protocols have been designed with the goal of being secure against adversarial keys, and few real-world certificate authorities validate that registrees provide valid public keys or prove knowledge of the corresponding secret keys. The availability of schemes with efficient non-interactive proofs-of-knowledge of secret key possession can only improve this situation. In the provable security literature, this *knowledge of secret key* solution to *rogue-key attacks* appeared early on in the study of multi-signatures by Micali et al. [38, Problem 4 and Fix 4].

Protocol design in strong security model. More generally, these obstacles to secret key extraction have hindered modular composable protocol design. Camenisch et al. [20] developed a framework for practical universally composable (UC) zero-knowledge proofs, in which they identify proofs-of-knowledge of exponents as a major bottleneck. Dubovitskaya [26] constructed unlinkable redactable signatures and anonymous credentials that are UC-secure. Their construction requires proofs-of-knowledge of the signing key of a structure-preserving signature scheme, which in turn, as studied by Chase et al. [24], is an instance of a general transformation for making signature schemes simulatable [11]. Given these examples, we conjecture that fully structure-preserving signature schemes help build UC-secure privacy preserving protocols.

Strengthening privacy in group and ring signatures. In classical group and ring signatures, e.g. [16, 32, 41, 18], the goal of the adversary against privacy is to distinguish signatures from two *honest* members whose keys are actually generated and registered by the challenger. The attack game aborts if either of the targets is a corrupted member registered with an adversarially generated key. Instead of excluding such corrupt members from the scope of security, stronger privacy in the presence of adversarial keys can be guaranteed, if the challenger can extract the secret key to create group or ring signatures on their behalf. Such a model is meaningful when some keys are generated incorrectly, e.g., because of multiple potentially flawed implementations, but their owners nevertheless use them with the correct signing algorithm. Note that this requires a trusted common reference string that puts mild assumptions on the trust model to retain other security properties such as unforgeability and non-frameability: the extraction trapdoor must be inaccessible for the adversary.

Other applications of FSPS are settings in which the signing keys need to be verifiably encrypted,

for instance when extending delegatable anonymous credential systems [13, 29, 24] with all-or-nothing non-transferability [21].

Organization. After introducing notations and definitions in Section 2, we review POS and xRMA-secure FSPS in Section 3. We then construct a shrinking SPTC scheme in Section 4 where we first present a γ-binding scheme in Section 4.2 and use it to construct an CMTCR-secure SPTC in Section 4.3. We present FSPS schemes in Section 5. Starting from a simple construction in Section 5.1 that identifies problems, we present our main construction in Section 5.2. We then discuss their performance in Section 5.3 and a lower bound for signature and public key sizes in Section 5.4. Finally in Section 6, we discuss variations in our constructions.

2 Preliminaries

2.1 Notations

By $|X|$ we denote the size of X (in some implicit unit). In particular, if X consists of group elements of some groups, it counts the number of elements in X. For x representing an object, $\vec{x}$ denotes an ordered set of x and is understood as $\vec{x} = (x_1, \ldots, x_n)$ for some positive integer n that is limited by a polynomial in the security parameter. The size n will be implicit if it is not very important in the context. By $y \leftarrow A(x)$, we denote that algorithm A takes x as input and outputs y. When it is clear from the context, we abuse notation like $\vec{y} \leftarrow A(\vec{x})$ to denote repetition of execution $y_i \leftarrow A(x_i)$ for $x_i \in \vec{x}$ and $y_i \in \vec{y}$.

2.2 Bilinear Groups

Let $\mathcal{G}$ be a generator of bilinear groups that takes security parameter 1^λ as input and outputs $\Lambda := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G})$, where p is a λ-bit prime, $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are groups of prime order p with efficiently computable group operations, membership tests, and bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Elements G and $\tilde{G}$ are default random generators of $\mathbb{G}_1$, $\mathbb{G}_2$, and $e(G, \tilde{G})$ generates $\mathbb{G}_T$. We use the multiplicative notation for group operations in $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$. The pairing operation e satisfies that ${}^\forall A \in \mathbb{G}_1$, ${}^\forall B \in \mathbb{G}_2$, ${}^\forall x, y \in \mathbb{Z}$: $e(A^x, B^y) = e(A, B)^{xy}$. An equation of the form $\prod_i \prod_j e(A_i, B_j)^{a_{ij}} = 1$ for constants $a_{ij} \in \mathbb{Z}_p$, and constants or variables $A_i \in \mathbb{G}_1$, $B_j \in \mathbb{G}_2$ is called a pairing product equation (PPE). By $\mathbb{G}_1^*$, we denote $\mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$, and similar for $\mathbb{G}_2^*$ and $\mathbb{Z}_p^*$.

Throughout the paper, we work over asymmetric bilinear groups (so-called Type-III setting [30]) where no efficient isomorphisms exist between $\mathbb{G}_1$ and $\mathbb{G}_2$. Some building blocks in our construction rely on the double pairing assumption [3].

Assumption 1 (Double Pairing Assumption: DBP). *The double pairing assumption holds in $\mathbb{G}_2$ relative to $\mathcal{G}$ if, for all probabilistic polynomial-time algorithms $\mathcal{A}$, probability*

$$\Pr\left[\begin{array}{ll} \Lambda \leftarrow \mathcal{G}(1^\lambda); & \\ \tilde{G}_z \leftarrow \mathbb{G}_2^*; & : \quad (Z, R) \in \mathbb{G}_1^* \times \mathbb{G}_1^*, \wedge \\ (Z, R) \leftarrow \mathcal{A}(\Lambda, \tilde{G}_z) & \quad 1 = e(Z, \tilde{G}_z)\, e(R, \tilde{G}) \end{array}\right] \tag{1}$$

is negligible in security parameter λ.

The DBP assumption in $\mathbb{G}_1$ is defined by swapping $\mathbb{G}_1$ and $\mathbb{G}_2$ in the above definition. Note that the DBP assumption (in $\mathbb{G}_1$ and $\mathbb{G}_2$) is implied by the Decision Diffie-Hellman assumption (in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively) which is often assumed in Type-III setting.

We also use a building block that requires more assumptions such as DDH_2, XDLIN_1, and co-CDH that we refer to [2] for definitions.

2.3 Digital Signatures

In this section we recall definitions of digital signatures, one-time signatures and their security notions. On top of the standard notions, we define structure-preserving and fully structure-reserving signatures.

Definition 1 (Digital Signature Scheme). *A digital signature scheme is a set of algorithms* $\{\mathsf{Setup}, \mathsf{Key}, \mathsf{Sign}, \mathsf{Vrf}\}$. $\mathsf{Setup}(1^\lambda) \to gk$ *is a setup function that, given a security parameter* λ*, generates common parameter* gk*, which defines message space* $\mathcal{M}$. $\mathsf{Key}(gk) \to (vk, sk)$ *is a key generation algorithm that takes common parameter* gk *and generates a verification key* vk *and a signing key* sk. $\mathsf{Sign}(sk, m) \to \sigma$ *is a signature generation algorithm that computes a signature* σ *for input message* $m \in \mathcal{M}$ *by using signing key* sk. $\mathsf{Vrf}(vk, m, \sigma) \to 1/0$ *is a verification algorithm that outputs 1 for acceptance or 0 for rejection according to the input.*

For any legitimately generated gk*,* vk*,* sk *and any* $m \in \mathcal{M}$*, it must hold that* $1 = \mathsf{Vrf}(vk, m, \mathsf{Sign}(sk, m))$*. A key pair* (vk, sk) *is correct with respect to* gk *if it is in the range of* $\mathsf{Key}(gk)$*.*

Definition 2 (Unforgeability against Adaptive Chosen-Message Attacks). *A signature scheme,* $\mathsf{SIG} = \{\mathsf{Setup}, \mathsf{Key}, \mathsf{Sign}, \mathsf{Vrf}\}$*, is unforgeable against adaptive chosen message attacks (UF-CMA) if the following advantage function is negligible against any polynomial-time adversary* $\mathcal{A}$*.*

$$\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SIG},\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{l|l} gk \leftarrow \mathsf{Setup}(1^\lambda), & m^\dagger \notin Q_m \wedge \\ (vk, sk) \leftarrow \mathsf{Key}(gk), & 1 = \mathsf{Vrf}(vk, m^\dagger, \sigma^\dagger) \\ (\sigma^\dagger, m^\dagger) \leftarrow \mathcal{A}^{\mathcal{O}_{sk}}(vk) & \end{array}\right], \tag{2}$$

where $\mathcal{O}_{sk}$ *is an oracle that, given* m*, executes* $\sigma \leftarrow \mathsf{Sign}(sk, m)$*, records* m *to* Q_m*, and returns* σ*.*

A non-adaptive chosen message attack is defined by letting adversary $\mathcal{A}$ commit to the messages to query before seeing vk. ($\mathcal{A}$ is given gk that defines the message space.) Existential unforgeability against non-adaptive chosen message attack is denoted by UF-NACMA.

A one-time signature scheme is a digital signature scheme with the limitation that a verification key has to be used only once to retain security. Unforgeability against one-time chosen message attacks is defined as in Definition 2 by restricting the game to answer only a single signing oracle request.

Definition 3 (Structure-Preserving Signature Scheme). *A digital signature scheme is called structure-preserving with respect to bilinear group generator* $\mathcal{G}$ *if the following conditions are all satisfied. 1) Common parameter* gk *consists of a group description* Λ *generated by* $\mathcal{G}$ *and constants* a_{ij} *in* $\mathbb{Z}_p$*. 2) Verification key* vk *consists of group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *other than* gk*. 3) Messages and signatures consist of group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*. 4) Verification algorithm* Vrf *consists only of evaluating membership in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *and relations described by paring product equations.*

When messages consist of both source groups, $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, they are called bilateral. They are unilateral, otherwise.*

The notion of structure-preserving cryptography requires *public* components to be group elements. We extend it so that *private* components consist of group elements as well.

Definition 4 (Fully Structure-Preserving Signature Schemes). *A structure-preserving signature scheme is fully structure-preserving if the following additional conditions are also satisfied. 5) Signing key* sk *(other than included* vk*) consists of group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*. 6) Correctness of* sk *with respect to* vk *can be verified by evaluating membership in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *and relations described by pairing product equations.*

Once conditions 5 and 6 are satisfied, one can construct proof of knowledge about the secret keys by using the Groth-Sahai proof system, which allows one to extract a correct secret key corresponding to the verification key. It is however important to note that there could exist more than one correct secret key for a verification key and they may yield signatures in different distributions. One might need stronger extractability that allows to extract the secret key for a particular distribution of signatures. It is indeed the case for the group signature application mentioned in Section 1. Our concrete scheme allows one to efficiently prove the relation between a secret key and a signature. See Section 3.3.

3 Building Blocks

3.1 Common Setup Function

Building blocks in this paper are defined with individual setup functions. As we work over bilinear groups, an output from a setup function should include a description of bilinear groups Λ. Some random generators specific to the building block may be included as well. Other parameters such as message spaces, are also defined there.

The individual setup functions will be merged into a common setup function, denoted as Setup, when the building blocks are used together in constructing upper-level schemes. By $gk \leftarrow \mathsf{Setup}(1^\lambda)$, we mean that Setup takes security parameter λ and generates a common parameter gk. This formulation is useful to share some domains in the building blocks. For instance, we require the message space of a signature scheme to match the key spaces of another signature scheme. Due to the interdependence between building blocks, it is inherent that Setup is constructed from individual setup algorithms in a non-blackbox manner. Suppose that two building blocks, say A and B, are used together. We say that A and B have common setup function Setup if $gk \leftarrow \mathsf{Setup}(1^\lambda)$ can be simulated whichever of $gk_\mathsf{A} \leftarrow \mathsf{A.Setup}(1^\lambda)$ or $gk_\mathsf{B} \leftarrow \mathsf{B.Setup}(1^\lambda)$ is given, and both gk_A and gk_B can be recovered from gk in polynomial time. In the rest of the paper, we abuse this property and give common parameter gk to individual functions of A and B.

3.2 Partially One-time Signatures

When only a part of a verification key of one-time signatures must be updated for every signing, i.e., the remaining part of the verification key can be used an unbounded number of times, the scheme is called partially one-time [17, 2].

Definition 5 (Partially One-time Signature Scheme). *A partially one-time signature scheme is a set of algorithms $\mathsf{POS} = \{\mathsf{Setup}, \mathsf{Key}, \mathsf{Ovk}, \mathsf{Sign}, \mathsf{Vrf}\}$ that:*

- $\mathsf{Setup}(1^\lambda) \to gk$: *A setup function that, given a security parameter λ, generates common parameter gk, which defines message space $\mathcal{M}$.*
- $\mathsf{Key}(gk) \to (vk, sk)$: *A long-term key generation function that takes gk and outputs a long-term key pair (vk, sk).*
- $\mathsf{Ovk}(gk) \to (ovk, osk)$: *A one-time key generation function that takes gk and outputs a one-time key pair (ovk, osk).*
- $\mathsf{Sign}(sk, osk, m) \to \sigma$: *A signing function that takes sk, osk and a message m as inputs and issues a signature σ.*
- $\mathsf{Vrf}(vk, ovk, m, \sigma) \to 1/0$: *A verification function that outputs 1 or 0 according to the validity of the input.*

For any $gk \leftarrow \mathsf{Setup}(1^\lambda)$, $(vk, sk) \leftarrow \mathsf{Key}(gk)$, $m \in \mathcal{M}$, and $(ovk, osk) \leftarrow \mathsf{Ovk}(gk)$, $\sigma \leftarrow \mathsf{Sign}(sk, osk, m)$, it must hold that $1 \leftarrow \mathsf{Vrf}(vk, ovk, m, \sigma)$.

Definition 6 (One-time Chosen-Message Attack for POS). *A partially one-time signature scheme, $\mathsf{POS} = \{\mathsf{Setup}, \mathsf{Key}, \mathsf{Ovk}, \mathsf{Sign}, \mathsf{Vrf}\}$, is unforgeable against non-adaptive partial one-time chosen message attacks (OT-NACMA), if advantage function $\mathrm{Adv}^{\mathsf{ot\text{-}nacma}}_{\mathsf{POS},\mathcal{A}}(\lambda)$ defined by probability*

$$\Pr\left[\begin{array}{l|l} gk \leftarrow \mathsf{Setup}(1^\lambda), & ovk^\dagger \in Q_{mv} \ \wedge \ (ovk^\dagger, m^\dagger) \notin Q_{mv} \ \wedge \\ (vk, sk) \leftarrow \mathsf{Key}(gk), & 1 = \mathsf{Vrf}(vk, ovk^\dagger, m^\dagger, \sigma^\dagger) \\ (ovk^\dagger, \sigma^\dagger, m^\dagger) \leftarrow \mathcal{A}^{\mathcal{O}_{sk}}(vk) & \end{array}\right] \tag{3}$$

is negligible against any polynomial-time adversary $\mathcal{A}$. Here $\mathcal{O}_{sk}$ is an oracle that, given $m \in \mathcal{M}$, executes $(ovk, osk) \leftarrow \mathsf{Ovk}(gk)$, $\sigma \leftarrow \mathsf{Sign}(sk, osk, m)$, records (ovk, m) to Q_{mv}, and returns (σ, ovk). When $\mathcal{O}_{sk}$ allows $\mathcal{A}$ to separately access Ovk and Sign, it is called an adaptive partial one-time chosen message attack (OT-CMA).

Obviously, OT-CMA security implies OT-NACMA security. It is also clear that OT-NACMA (OT-CMA) POS can be used as UF-NACMA (UF-CMA, resp.) OTS. The following construction taken from [2] (with conceptual modifications for better efficiency in generating keys and signatures) is OT-CMA under the DBP assumption in $\mathbb{G}_1$.

[**Partially One-time Signature Scheme: POS**]

$\mathsf{Setup}(1^\lambda)$: Run $gk := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G}) \leftarrow \mathcal{G}(1^\lambda)$. Set message space $\mathcal{M}$ to $\mathbb{G}_2^\ell$ for preliminary-fixed positive integer ℓ.

$\mathsf{Key}(gk)$: Take generators G and $\tilde{G}$ from gk. Choose w_z randomly from $\mathbb{Z}_p^*$, and compute $G_z := G^{w_z}$. For $i = 1, \ldots, \ell$, uniformly choose χ_i from $\mathbb{Z}_p$ and compute $G_i := G^{\chi_i}$. Output $vk := (G_z, G_1, \ldots, G_\ell) \in \mathbb{G}_1^{\ell+1}$ and $sk := (\chi_1, \ldots, \chi_\ell, w_z)$.

$\mathsf{Ovk}(gk)$: Choose $a \leftarrow \mathbb{Z}_p$ and output $ovk = A := G^a$, and $osk := a$.

$\mathsf{Sign}(sk, osk, m)$: Parse m into $(\tilde{M}_1, \cdots, \tilde{M}_\ell) \in \mathbb{G}_2^\ell$. Take a and w_z from osk and sk, respectively. Choose ζ randomly from $\mathbb{Z}_p^*$ and compute the signature as $(\tilde{Z}, \tilde{R})$ where $\tilde{Z} = \tilde{G}^\zeta$, $\tilde{R} = \tilde{G}^{a - \zeta\, w_z} \prod_{i=1}^{\ell} \tilde{M}_i^{-\chi_i}$.

$\mathsf{Vrf}(vk, ovk, m, \sigma)$: Parse σ as $(\tilde{Z}, \tilde{R}) \in \mathbb{G}_2^2$, m as $(\tilde{M}_1, \ldots, \tilde{M}_\ell) \in \mathbb{G}_2^\ell$, and ovk as A. Return 1, if $e(A, \tilde{G}) = e(G_z, \tilde{Z})\, e(G, \tilde{R}) \prod_{i=1}^{\ell} e(G_i, \tilde{M}_i)$ holds. Return 0, otherwise.

3.3 xRMA-secure Fully Structure-Preserving Signature Scheme

We follow the notion of extended random message attacks and take a concrete scheme from [2].

Definition 7 (Unforgeability against Extended Random Message Attacks). *A signature scheme, $\mathsf{xSIG} = \{\mathsf{Setup}, \mathsf{Key}, \mathsf{Sign}, \mathsf{Vrf}\}$, is unforgeable against extended random message attacks (UF-XRMA) with respect to message sampler $\mathsf{SampleM}$ if probability*

$$\mathrm{Adv}^{\mathsf{uf\text{-}xrma}}_{\mathsf{xSIG},\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{l|l} gk \leftarrow \mathsf{Setup}(1^\lambda), & \\ (vk, sk) \leftarrow \mathsf{Key}(gk), & \\ \vec{m} \leftarrow \mathsf{SampleM}(gk; \omega), & m^\dagger \notin \vec{m} \wedge \\ \vec{\sigma} \leftarrow \mathsf{Sign}(sk, \vec{m}), & 1 = \mathsf{Vrf}(vk, m^\dagger, \sigma^\dagger) \\ (\sigma^\dagger, m^\dagger) \leftarrow \mathcal{A}(vk, \vec{\sigma}, \vec{m}, \omega) & \end{array}\right] \tag{4}$$

is negligible against any polynomial-time adversary $\mathcal{A}$. Here ω is a uniformly chosen randomness.

[**xRMA-secure Signature Scheme: xSIG**]

$\mathsf{Setup}(1^\lambda)$: Run $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G}) \leftarrow \mathcal{G}(1^\lambda)$. For some fixed $\ell \geq 1$, choose $u_1, \cdots, u_\ell$, ϱ, δ randomly from $\mathbb{Z}_p^*$ and compute $F_1 := G^\varrho$, $F_2 := G^\delta$, $\tilde{F}_1 := \tilde{G}^\varrho$, $\tilde{F}_2 := \tilde{G}^\delta$, $U_i := G^{u_i}$, and $\tilde{U}_i := \tilde{G}^{u_i}$. Output $gk := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G}, F_1, F_2, \tilde{F}_1, \tilde{F}_2, \{U_i, \tilde{U}_i\}_{i=1}^\ell)$. This constitutes the message space $\mathcal{M} = \{(\tilde{M}_{11}, \tilde{M}_{12}, \tilde{M}_{13}), \ldots, (\tilde{M}_{\ell 1}, \tilde{M}_{\ell 2}, \tilde{M}_{\ell 3}) \,|\, {}^{\forall} i, {}^{\exists} m_i \in \mathbb{Z}_p \text{ s.t. } (\tilde{M}_{i1}, \tilde{M}_{i2}, \tilde{M}_{i3}) = (\tilde{F}_1^{m_i}, \tilde{F}_2^{m_i}, \tilde{U}_i^{m_i})\}$.

$\mathsf{Key}(gk)$: On input gk, choose τ_1, τ_2, $\tau_3, \rho, a, b, \alpha$ from $\mathbb{Z}_p$, and compute

$$\begin{array}{llll} \tilde{V}_1 := \tilde{G}^b, & \tilde{V}_2 := \tilde{G}^a, & \tilde{V}_3 := \tilde{G}^{ba}, & \tilde{V}_4 := \tilde{G}^{\tau_1 + a\tau_2}, \\ \tilde{V}_5 := \tilde{V}_4^b, & \tilde{V}_6 := \tilde{G}^{\tau_3}, & V_7 := G^\rho, & \tilde{V}_8 := \tilde{G}^{\alpha b/\rho}, \\ K_1 := G^\alpha, & K_2 := G^b, & K_3 := G^{\tau_1}, & K_4 := G^{\tau_2}. \end{array} \tag{5}$$

(For completeness of description, pick $\tilde{V}_8$ uniformly from $\mathbb{G}_2$ if $\rho = 0$.) Output $vk := (gk, \tilde{V}_1, \tilde{V}_2, \tilde{V}_3, \tilde{V}_4, \tilde{V}_5, \tilde{V}_6, V_7, \tilde{V}_8)$ and $sk := (vk, K_1, K_2, K_3, K_4)$.

$\mathsf{Sign}(sk, M)$: Parse message M into $\{(\tilde{M}_{11}, \tilde{M}_{12}, \tilde{M}_{13}), \cdots, (\tilde{M}_{\ell 1}, \tilde{M}_{\ell 2}, \tilde{M}_{\ell 3})\} \in \mathcal{M}$. Select $r_1, r_2, z \leftarrow \mathbb{Z}_p$, set $r := r_1 + r_2$, compute

$$\begin{array}{lll} \tilde{S}_0 := (\tilde{V}_6 \prod_{i=1}^{\ell} \tilde{M}_{i3})^{r_1}, & S_1 := K_1 {K_3}^r, & S_2 := {K_4}^r G^{-z}, \\ S_3 := {K_2}^z, & S_4 := {K_2}^{r_2}, & S_5 := G^{r_1}. \end{array} \tag{6}$$

Output $\sigma := (\tilde{S}_0, \ldots, S_5) \in \mathbb{G}_2 \times \mathbb{G}_1^5$.

$\mathsf{Vrf}(vk, M, \sigma)$: Output 1 if the following relations hold:

$$\begin{aligned} & e(S_5, \tilde{V}_6 \prod_{i=1}^{\ell} \tilde{M}_{i3}) = e(G, \tilde{S}_0), \\ & e(S_1, \tilde{V}_1) e(S_2, \tilde{V}_3) e(S_3, \tilde{V}_2) = e(S_4, \tilde{V}_4) e(S_5, \tilde{V}_5) e(V_7, \tilde{V}_8), \\ & e(F_1, \tilde{M}_{i3}) = e(U_i, \tilde{M}_{i1}), \quad e(F_2, \tilde{M}_{i3}) = e(U_i, \tilde{M}_{i2}) \quad \text{for } i = 1, \cdots, \ell. \end{aligned} \tag{7}$$

Output 0, otherwise.

The above scheme comes with trivial modifications from the original in [2]. First it is extended to sign random messages consisting of $\ell \geq 1$ message blocks, and second it takes randomness from $\mathbb{Z}_p$ rather than $\mathbb{Z}_p^*$ in the key generation. Those changes do not essentially affect to the security that we recall below.

Theorem 1 ([2]). *If the DDH_2, $XDLIN_1$, and co-CDH assumptions hold, then the above xSIG is UF-XRMA with respect to the message sampler that returns $aux = m_i$ for every random message block $(\tilde{F}_1^{m_i}, \tilde{F}_2^{m_i}, \tilde{U}_i^{m_i})$.*

Theorem 2. *The above xSIG is fully structure-preserving.*

Proof. By inspection, it is clear that vk (modulo group description in gk), sk, M, and σ consist of source group elements, and $\mathsf{xSIG.Vrf}$ consists of evaluating PPEs.

Next we show that the following PPEs are satisfied if and only if the verification key and the secret key is in the support of $\mathsf{xSIG.Key}$.

$$\begin{aligned} & e(K_2, \tilde{G}) = e(G, \tilde{V}_1), \quad e(G, \tilde{V}_3) = e(K_2, \tilde{V}_2), \quad e(K_1, \tilde{V}_1) = e(V_7, \tilde{V}_8), \\ & e(K_2, \tilde{V}_4) = e(G, \tilde{V}_5), \quad e(K_3, \tilde{G})\, e(K_4, \tilde{V}_2) = e(G, \tilde{V}_4). \end{aligned} \tag{8}$$

Showing correctly generated keys satisfy the above relations is trivial. We argue the other direction as follows. The independent variables that define a key pair are a, b, α, τ_1, τ_2, τ_3 and ρ. They are uniquely determined by $\tilde{V}_2$, $\tilde{V}_1$, K_1, K_3, K_4, $\tilde{V}_6$, and V_7, respectively. We verify that the remaining $\tilde{V}_3$, $\tilde{V}_4$, $\tilde{V}_5$, $\tilde{V}_8$, and K_2 are in the support of the correct distribution if the above relations are satisfied. The first equation is $e(K_2, \tilde{G}) = e(G, \tilde{G})^b$ that defines $K_2 = G^b$. The second equation is $e(G, \tilde{V}_3) = e(G, \tilde{G})^{ba}$ that defines $\tilde{V}_3 = \tilde{G}^{ba}$. The third equation is $e(G, \tilde{G})^{\alpha b} = e(G, \tilde{V}_8)^{\rho}$ that defines $\tilde{V}_8 = \tilde{G}^{\alpha b/\rho}$ for $\rho \neq 0$. If $\rho = 0$, $\tilde{V}_8$ can be an arbitrary value as prescribed in the key generation. The fourth equation is $e(G, \tilde{V}_4)^b = e(G, \tilde{V}_5)$ that defines $\tilde{V}_5 = \tilde{V}_4^b$. The last equation is $e(G, \tilde{G})^{\tau_1 + a\tau_2} = e(G, \tilde{V}_4)$ that defines $\tilde{V}_4 = \tilde{G}^{\tau_1 + a\tau_2}$ as prescribed. ∎

Proving a Correct Secret Key for a Signature. One can construct a proof of knowledge of a correct secret key using Groth-Sahai proofs of the relation in (8). But it does not necessarily mean that the secret key is the one used for creating particular signatures. Observe that in the above xSIG there is more than one secret keys for a verification key and each secret key yields signatures in a different distribution. It is possible to efficiently prove one's possession of a secret key used to create the signature in question by proving the following relation. It requires randomness z and r used for the signature as part of the witness. The underlined variables are the witnesses.

$$\begin{aligned} & e(\underline{K_2}, \underline{\tilde{G}^r}) = e(S_4, \tilde{G}) e(S_5, \tilde{V}_1), \quad e(S_1, \tilde{G}) = e(\underline{K_1}, \tilde{G})\, e(\underline{K_3}, \underline{\tilde{G}^r}), \\ & e(S_3, \tilde{G}) = e(\underline{G^z}, \tilde{V}_1), \qquad\qquad e(S_2, \tilde{G})\, e(\underline{G^z}, \tilde{G}) = e(\underline{K_4}, \underline{\tilde{G}^r}) \end{aligned} \tag{9}$$

Consider a verification key and a signature that satisfy the verification equations in (7) and a secret key that satisfies (8) with respect to the verification key. Suppose that they also satisfy the equations in (9).

Define r_1 and r_2 by $r_1 = \log_G S_5$ and $r_2 = \log_{K_2} S_4$. Parameter b is defined by $b = \log_G K_2 = \log_{\tilde{G}} \tilde{V}_1$ In the exponent, the first relation in (9) is read $br = br_2 + br_1$ that means G^r is correctly related to S_4 and S_5. The second relation in (9) then guarantees $S_1 = K_1 K_3^r$ for this r, K_1, and K_3. The third relation in (9) proves that $S_3 = G^{z \log_{\tilde{G}} \tilde{V}_1} = G^{zb} = {K_2}^z$ for some z determined by G^z. Finally, the last relation in (9) is for $S_2 = {K_4}^r \tilde{G}^{-z}$. Thus the secret key fulfilling all relations in (9) satisfies relations in (6) with respect to the signature and the verification key. Namely, the secret key is the one used to create the signature.

The cost for proving the correct secret key for a signature is as follows. The number of additional witness is 2, i.e., $(G^z, \tilde{G}^r)$. The number of the commitment is $2 \cdot 2 = 4$ ($2 \cdot 2 = 4$ in $\mathbb{G}_1$). Since three of the four equations in (9) have the witness in both source groups and the rest one has a witness only in $\mathbb{G}_1$, the required number of elements for the proof is $8 \cdot 3 + 2 \cdot 1 = 26$ ($4 \cdot 3 = 12$ in $\mathbb{G}_1$, $4 \cdot 3 + 2 \cdot 1 = 14$ in $\mathbb{G}_2$). In total the cost for proving the correct secret key for a signature is $4 + 26 = 30$.

4 Trapdoor Commitment Schemes

4.1 Definitions

We adopt the following standard syntax for trapdoor commitment schemes.

Definition 8 (Trapdoor Commitment Scheme). *A trapdoor commitment scheme* TC *is a tuple of polynomial-time algorithms* $\mathsf{TC} = \{\mathsf{Setup}, \mathsf{Key}, \mathsf{Com}, \mathsf{Vrf}, \mathsf{SimCom}, \mathsf{Equiv}\}$ *that:*

- $\mathsf{Setup}(1^\lambda) \to gk$: *A common-parameter generation algorithm that takes security parameter* λ *and outputs a common parameter,* gk*. It determines the message space* $\mathcal{M}$*, the commitment space* $\mathcal{C}$*, and opening space* $\mathcal{I}$*.*
- $\mathsf{Key}(gk) \to (ck, tk)$: *A key generation algorithm that takes* gk *as input and outputs a commitment key,* ck*, and a trapdoor key,* tk*.*
- $\mathsf{Com}(ck, m) \to (com, open)$: *A commitment algorithm that takes* ck *and message* $m \in \mathcal{M}$ *and outputs a commitment,* $com \in \mathcal{C}$*, and an opening information,* $open \in \mathcal{I}$*.*
- $\mathsf{Vrf}(ck, com, m, open) \to 1/0$: *A verification algorithm that takes* ck*,* com*,* m*, and* $open$ *as input, and outputs* 1 *or* 0 *representing acceptance or rejection, respectively.*
- $\mathsf{SimCom}(gk) \to (com, ek)$: *A sampling algorithm that takes common parameter* gk *and outputs commitment* com *and equivocation key* ek*.*
- $\mathsf{Equiv}(m, ek, tk) \to open$: *An algorithm that takes* ck, ek, tk *and* $m \in \mathcal{M}$ *as input, and returns* $open$*.*

It is correct if, for all $\lambda \in \mathbb{N}$, $gk \leftarrow \mathsf{Setup}(1^\lambda)$, $(ck, tk) \leftarrow \mathsf{Key}(gk)$, $m \leftarrow \mathcal{M}$, $(com, open) \leftarrow \mathsf{Com}(ck, m)$*, it holds that* $1 = \mathsf{Vrf}(ck, com, m, open)$*. Furthermore, it is statistical trapdoor if, for any* $gk \in \mathsf{Setup}(1^\lambda)$, $(ck, tk) \in \mathsf{Key}(gk)$, $m \in \mathcal{M}$, $(com, open) \leftarrow \mathsf{Com}(ck, m)$, $(com', ek) \leftarrow \mathsf{SimCom}(gk)$*;* $open' \leftarrow \mathsf{Equiv}(m, ek, tk)$*, two distributions* $(ck, m, com, open)$ *and* $(ck, m, com', open')$ *are statistically close.*

Definition 9 (Structure-Preserving Trapdoor Commitment Scheme). *A trapdoor commitment scheme is structure-preserving relative to group generator* $\mathcal{G}$ *if its* gk *includes a description of bilinear groups generated by* $\mathcal{G}$ *and its commitment keys, messages, commitments, and opening information consist only of source group elements, and the verification function consists only of evaluating group membership and relations described by pairing product equations.*

We say that a commitment scheme is shrinking if $|com| \leq |m|$ where equality holds only for the case of $|m| = 1$.

Trapdoor commitments should be hiding and binding. Since the hiding property follows from the statistical trapdoor property and is not important for our purpose, we focus on the binding property in the rest of this paper.

The standard binding property requires that it is infeasible for any polynomial-time adversary to find two distinct messages and openings for a single commitment value com. It is also referred to as collision resistance. A weaker notion known as target collision resistance asks the adversary to find a collision on a given message. We here introduce a weaker binding notion that lies between collision resistance and target collision resistance. This new notion, which we call *chosen-message target collision resistance* (CMTCR), allows the adversary to choose the message but it is committed to by the challenger. Thus, the adversary does not know the randomness used to create the target commitment.

Definition 10 (Chosen-Message Target Collision Resistance). *For a trapdoor commitment scheme, TC, let $\mathcal{O}_{ck}$ denote an oracle that, given $m \in \mathcal{M}$, executes $(com, open) \leftarrow \mathsf{Com}(ck, m)$, records (com, m) to Q, and returns $(com, open)$. We say TC is chosen-message target collision resistant if advantage $\mathrm{Adv}^{\mathsf{cmtcr}}_{\mathsf{TC},\mathcal{A}}(\lambda)$ defined by*

$$\Pr\left[\begin{array}{l|l} gk \leftarrow \mathsf{Setup}(1^\lambda), & com^\dagger \in Q \wedge (com^\dagger, m^\dagger) \notin Q \wedge \\ (ck, tk) \leftarrow \mathsf{Key}(gk), & 1 = \mathsf{Vrf}(ck, com^\dagger, m^\dagger, open^\dagger) \\ (com^\dagger, m^\dagger, open^\dagger) \leftarrow \mathcal{A}^{\mathcal{O}_{ck}}(ck) & \end{array}\right] \tag{10}$$

is negligible in security parameter λ for any polynomial-time adversary $\mathcal{A}$.

In [4], a similar notion called random-prefix collision resistance is defined as a binding property for hash functions. It is a special case of CMTCR as it limits the oracle access only once and the opening information is restricted to the random coins.

It is not necessarily clear how this weakened binding property aims for efficient construction of shrinking TC. It should be true, however, that the internal randomness of Com that cannot be controlled by the adversary plays an essential role to bind a message to a commitment. This insight leads us to the two-story construction of shrinking TC in Section 4.3.

4.2 γ-Binding Commitment Scheme

This section presents a new primitive we call a γ-binding commitment scheme. It has a special property that the message space $\mathcal{M}^{com}$ for creating a commitment and the space $\mathcal{M}^{ver}$ for verification differ and there exists an efficiently computable bijection $\gamma : \mathcal{M}^{com} \to \mathcal{M}^{ver}$ that computes messages for verification from those for committing. The formal definition is as follows.

Definition 11 (γ-Binding Commitment Scheme). *A γ-binding commitment scheme is a set of algorithms $\mathsf{TC}\gamma = \{\mathsf{Setup}, \mathsf{Key}, \mathsf{Com}, \mathsf{Vrf}, \mathsf{SimCom}, \mathsf{Equiv}\}$ that:*

- $\mathsf{Setup}(1^\lambda) \to gk$: *A setup function that, given a security parameter λ, generates common parameter gk, which defines message spaces; $\mathcal{M}^{com}$ for commitment generation and $\mathcal{M}^{ver}$ for verification, and an efficiently computable bijection $\gamma : \mathcal{M}^{com} \to \mathcal{M}^{ver}$. It also determines the commitment space $\mathcal{C}$, and the opening space $\mathcal{I}$.*

- $\mathsf{Key}(gk) \to (ck, tk)$: *A key generation algorithm that takes gk and outputs a public commitment key, ck, and a trapdoor key, tk.*

- $\mathsf{Com}(ck, m) \to (com, open)$: *A commitment algorithm that takes ck and message $m \in \mathcal{M}^{com}$ and outputs a commitment, $com \in \mathcal{C}$, and an opening information, $open \in \mathcal{I}$.*

- $\mathsf{Vrf}(ck, com, M, open) \to 1/0$: *A verification algorithm that takes ck, com, $M \in \mathcal{M}^{ver}$, and $open$ as inputs, and outputs 1 or 0 representing acceptance or rejection, respectively.*

- $\mathsf{SimCom}(gk) \to (com, ek)$: *A sampling algorithm that takes common parameter gk and outputs commitment com and equivocation key ek.*

- $\mathsf{Equiv}(M, ek, tk) \to open$: *An algorithm that takes ck, ek, tk, and $M \in \mathcal{M}^{ver}$ as input and returns $open$.*

Correctness, statistical trapdoor, and shrinking property are defined as well as Definition 8.

We say that a γ-binding commitment scheme is structure-preserving with respect to verification if ck, com, $open$, and $\mathcal{M}^{ver}$ consist of source group elements of bilinear groups and the verification function consists only of evaluating group membership and pairing product equations.

Next we formally define the security notions, γ-target collision resistance and γ-collision resistance. As well as ordinary notions of collision resistance, γ-collision resistance implies γ-target collision resistance.

Definition 12 (γ-Target Collision Resistance). *For a γ-binding commitment scheme, $\mathsf{TC}\gamma$, let $\vec{com}$ and $\vec{open}$ denote vectors of commitment and openings produced by Com for uniformly sampled messages $\vec{m}$. We say $\mathsf{TC}\gamma$ is γ-target collision resistant if advantage function $\mathrm{Adv}^{\mathsf{tcr}}_{\mathsf{TC}\gamma,\mathcal{A}}(\lambda)$ defined by*

$$\Pr\left[\begin{array}{l|l} gk \leftarrow \mathsf{Setup}(1^\lambda), (ck, tk) \leftarrow \mathsf{Key}(gk), & com \in \vec{com} \ \wedge \ M \notin \gamma(\vec{m}) \wedge \\ \vec{m} \leftarrow \mathcal{M}^{com}, (\vec{com}, \vec{open}) \leftarrow \mathsf{Com}(ck, \vec{m}), & 1 = \mathsf{Vrf}(ck, com, M, open) \\ (com, M, open) \leftarrow \mathcal{A}(ck, \vec{m}, \vec{com}, \vec{open}) & \end{array}\right]$$

is negligible in security parameter λ for any polynomial-time adversary $\mathcal{A}$.

Definition 13 (γ-Collision Resistance). *A γ-binding commitment scheme, $\mathsf{TC}\gamma$, is γ-collision resistant if advantage $\mathrm{Adv}^{\mathsf{cr}}_{\mathsf{TC}\gamma,\mathcal{A}}(\lambda)$ defined by*

$$\Pr\left[\begin{array}{l|l} gk \leftarrow \mathsf{Setup}(1^\lambda), (ck, tk) \leftarrow \mathsf{Key}(gk), & M_1 \neq M_2 \wedge \\ (com, M_1, open_1, M_2, open_2) \leftarrow \mathcal{A}(ck) & 1 = \mathsf{Vrf}(ck, com, M_1, open_1) \wedge \\ & 1 = \mathsf{Vrf}(ck, com, M_2, open_2) \end{array}\right]$$

is negligible in security parameter λ for any polynomial-time adversary $\mathcal{A}$.

Now we present a concrete scheme for a structure-preserving γ-binding trapdoor commitment scheme for $\gamma : \mathbb{Z}_p \to \mathbb{G}_1$. For our purpose, we only require target collision resistance but the concrete construction satisfies the stronger notion.

[**γ-Binding Trapdoor Commitment Scheme: $\mathsf{TC}\gamma$**]

$\mathsf{Setup}(1^\lambda)$: Run $\mathcal{G}(1^\lambda)$ and obtain $gk := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G})$. It defines message spaces $\mathcal{M}^{com} := \mathbb{Z}_p^\ell$, $\mathcal{M}^{ver} := \mathbb{G}_1^\ell$ for fixed $\ell \geq 1$ and bijection $\gamma : \mathbb{Z}_p^\ell \to \mathbb{G}_1^\ell$ by $\gamma(m_1, \ldots, m_\ell) = (G^{m_1}, \ldots, G^{m_\ell})$. Output gk.

$\mathsf{Key}(gk)$: For $i = 1, \ldots, \ell$, choose $\rho_i \leftarrow \mathbb{Z}_p^*$ and compute $\tilde{X}_i := \tilde{G}^{\rho_i}$. Output $ck := (gk, \tilde{X}_1, \ldots, \tilde{X}_\ell)$ and $tk := (gk, \rho_1, \ldots, \rho_\ell)$.

$\mathsf{Com}(ck, m)$: Parse m into $(m_1, \cdots, m_\ell) \in \mathbb{Z}_p^\ell$. Choose $\zeta \leftarrow \mathbb{Z}_p^*$ and compute

$$\tilde{G}_u := \tilde{G}^\zeta \prod_{i=1}^{\ell} \tilde{X}_i^{m_i}, \quad \text{and} \quad R := G^\zeta.$$

Output $com := \tilde{G}_u$ and $open := R$.

$\mathsf{Vrf}(ck, com, M, open)$: Parse $ck = (gk, \tilde{X}_1, \ldots, \tilde{X}_\ell)$, $open = R$, $M = (M_1, \ldots, M_\ell) \in \mathbb{G}_1^\ell$, and $com = \tilde{G}_u$, respectively. Take generators $(G, \tilde{G})$ from gk. Return 1 if

$$e(G, \tilde{G}_u) = e(R, \tilde{G}) \prod_{i=1}^{\ell} e(M_i, \tilde{X}_i) \tag{11}$$

holds. Return 0, otherwise.

$\mathsf{SimCom}(gk)$: Choose $\omega_u \in \mathbb{Z}_p^*$. Compute $\tilde{G}_u := \tilde{G}^{\omega_u}$ and output $com := \tilde{G}_u$ and $ek := \omega_u$.

$\mathsf{Equiv}(M, ek, tk)$: Parse $tk = (gk, \rho_1, \ldots, \rho_\ell)$, $ek = \omega_u$, and $M = (M_1, \ldots, M_\ell)$. Compute $R := G^{\omega_u} \prod_{i=1}^{\ell} M_i^{-\rho_i}$. Then output $open := R$.

Theorem 3. *TCγ is correct, statistical trapdoor, and structure-preserving with respect to verification. It is γ-collision resistant if the DBP assumption holds.*

Proof. Correctness is verified as

$$e(R,\tilde{G})\prod_{i=1}^{\ell}e(M_i,\tilde{X}_i)=e(G^{\zeta},\tilde{G})\prod_{i=1}^{\ell}e(G^{m_i},\tilde{X}_i)=e(G,\tilde{G}^{\zeta})\,e(G,\prod_{i=1}^{\ell}\tilde{X}_i^{m_i})\;=e(G,\tilde{G}_u).$$

To see if it is statistically trapdoor, observe that SimCom outputs $\tilde{G}_u$ uniformly over $\mathbb{G}_2^*$ whereas that from Com distributes statistically close to uniform over $\mathbb{G}_2$. Then R from Equiv is the one that is uniquely determined by the verification equation since it satisfies

$$e(R,\tilde{G})\prod_{i=1}^{\ell}e(M_i,\tilde{X}_i)=e(G^{\omega_u}\prod_{i=1}^{\ell}M_i^{-x_i},\tilde{G})\prod_{i=1}^{\ell}e(M_i,\tilde{G}^{x_i})=e(G,\tilde{G}_u).$$

Finally, it is obviously structure-preserving with respect to verification due to verification equation (11).

Next we prove the γ-collision resistance. Let $\mathcal{A}$ be an adversary that breaks the CR security of $\mathsf{TC}\gamma$. We show algorithm $\mathcal{B}$ that attacks the DBP with black-box access to $\mathcal{A}$. Given an instance $(e,\mathbb{G}_1,\mathbb{G}_2,G,\tilde{G},\tilde{G}_z)$ of the DBP, algorithm $\mathcal{B}$ sets up key ck as follows. Set $gk:=(p,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,e,G,\tilde{G})$. For $i=1,\ldots,\ell$, choose $\xi_i,\varphi_i\leftarrow(\mathbb{Z}_p^*)^2$ and set $\tilde{X}_i:=(\tilde{G}_z)^{\xi_i}\,\tilde{G}^{\varphi_i}$. Then give $ck:=(gk,\tilde{X}_1,\ldots,\tilde{X}_\ell)$ to $\mathcal{A}$.

Suppose that $\mathcal{A}$ outputs $(\tilde{G}_u,R_1,M_1,R_2,M_2)$ that passes the verification as required. $\mathcal{B}$ then outputs $(Z^\star,R^\star)$ where

$$R^\star:=\frac{R_1}{R_2}\prod_{i=1}^{\ell}\left(\frac{M_{1i}}{M_{2i}}\right)^{\varphi_i},\text{ and }\quad Z^\star:=\prod_{i=1}^{\ell}\left(\frac{M_{1i}}{M_{2i}}\right)^{\xi_i},\tag{12}$$

as the answer to the DBP. This completes the description of $\mathcal{B}$.

We first verify that the simulated ck is correctly distributed. In the key generation, gk is set legitimately to the given output of $\mathcal{G}$. Each simulated $\tilde{X}_i$ distributes uniformly over $\mathbb{G}_2$, whereas the real one distributes uniformly over $\mathbb{G}_2^*$. Thus, the simulated ck is statistically close to the real one.

We then argue that the resulting $(Z^\star,R^\star)$ is a valid answer to the given instance of the DBP. Since the output from $\mathcal{A}$ satisfies the verification equation, we have

$$1=e\left(\frac{R_1}{R_2},\tilde{G}\right)\prod_{i=1}^{\ell}e\left(\frac{M_{1i}}{M_{2i}},(\tilde{G}_z)^{\xi_i}\tilde{G}^{\varphi_i}\right)\tag{13}$$

$$=e\left(\prod_{i=1}^{\ell}\left(\frac{M_{1i}}{M_{2i}}\right)^{\xi_i},\tilde{G}_z^\star\right)\,e\left(\frac{R_1}{R_2}\prod_{i=1}^{\ell}\left(\frac{M_{1i}}{M_{2i}}\right)^{\varphi_i},\tilde{G}\right)=e(Z^\star,\tilde{G}_z^\star)\,e(R^\star,\tilde{G}).\tag{14}$$

Observe that every ξ_i is independent of the view of $\mathcal{A}$ as it is information theoretically hidden into $\tilde{X}_i$. Since a valid output from $\mathcal{A}$ satisfies $M_1\neq M_2$, there exists index $i^\star\in\{1,\ldots,\ell\}$ that $M_{1i^\star}\neq M_{2i^\star}$. Thus $Z^\star$ follows the distribution of $(M_{1i}/M_{2i})^{\xi_i}$ at $i=i^\star$. Since $M_{1i^\star}/M_{2i^\star}\neq 1$ and $\xi_{i^\star}$ is uniform over $\mathbb{Z}_p^*$, we conclude that $Z^\star=1$ occurs only with negligible probability.

Thus, $\mathcal{B}$ breaks the DBP assumption with almost the same probability and running time of $\mathcal{A}$ breaking the γ-collision resistance of $\mathsf{TC}\gamma$. ∎

4.3 Structure-Preserving Shrinking Trapdoor Commitment Scheme

Let POS be a partially one-time signature scheme. Let $\mathcal{M}_{\mathsf{pos}}$ be the message space of POS defined with respect to gk. We denote the key spaces as $\mathcal{K}_{\mathsf{pos}}^{vk}$, $\mathcal{K}_{\mathsf{pos}}^{sk}$, $\mathcal{K}_{\mathsf{pos}}^{ovk}$, and $\mathcal{K}_{\mathsf{pos}}^{osk}$ in a self-explanatory manner. Let $\gamma_{sk}:\mathcal{K}_{\mathsf{pos}}^{sk}\to\mathcal{K}_{\mathsf{pos}}^{vk}$ and $\gamma_{osk}:\mathcal{K}_{\mathsf{pos}}^{osk}\to\mathcal{K}_{\mathsf{pos}}^{ovk}$ be efficiently computable bijections. Let γ be $\gamma=\gamma_{sk}\times\gamma_{osk}^{(1)}\times\cdots\times\gamma_{osk}^{(k)}$. Let $\mathsf{TC}\gamma$ be a γ-binding trapdoor commitment scheme for such γ. It is assumed that POS and $\mathsf{TC}\gamma$ have a common setup function, Setup, that outputs gk based on $\mathsf{POS.Setup}$ and $\mathsf{TC}\gamma\mathsf{.Setup}$, as mentioned in Section 3.1. (When instantiated from POS in Section 3.2 and $\mathsf{TC}\gamma$ from Section 4.2, Setup is as simple as running $gk\leftarrow\mathcal{G}(1^\lambda)$). Using these building blocks, we construct an SPTC scheme, TC, achieving CMTCR security as follows.

[**Trapdoor Commitment Scheme: TC**]

$\mathsf{Setup}(1^\lambda)$: It the same as the common setup function for POS and $\mathsf{TC}\gamma$. The relevant message spaces are set as $\mathcal{M}^{com}_{\mathsf{gbc}} := \mathcal{K}^{sk}_{\mathsf{pos}} \times (\mathcal{K}^{osk}_{\mathsf{pos}})^k$, $\mathcal{M}^{ver}_{\mathsf{gbc}} := \mathcal{K}^{vk}_{\mathsf{pos}} \times (\mathcal{K}^{ovk}_{\mathsf{pos}})^k$, and $\mathcal{M} := (\mathcal{M}_{\mathsf{pos}})^k$ for some integer $k > 0$.

$\mathsf{Key}(gk)$: Run $(ck_{\mathsf{gbc}}, tk_{\mathsf{gbc}}) \leftarrow \mathsf{TC}\gamma.\mathsf{Key}(gk)$. Output $ck := ck_{\mathsf{gbc}}$ and $tk := tk_{\mathsf{gbc}}$. It is assumed that gk is included in ck. The message space for TC is set to $\mathcal{M} := (\mathcal{M}_{\mathsf{pos}})^k$.

$\mathsf{Com}(ck, M)$: Parse $ck := ck_{\mathsf{gbc}}$ and $M := (M^{(1)}, \cdots, M^{(k)}) \in (\mathcal{M}_{\mathsf{pos}})^k$. Take gk from ck. Run

$$
\begin{aligned}
(vk_{\mathsf{pos}}, sk_{\mathsf{pos}}) &\leftarrow \mathsf{POS.Key}(gk),\\
(ovk^{(i)}_{\mathsf{pos}}, osk^{(i)}_{\mathsf{pos}}) &\leftarrow \mathsf{POS.Ovk}(gk),\\
\sigma^{(i)}_{\mathsf{pos}} &\leftarrow \mathsf{POS.Sign}(sk_{\mathsf{pos}}, osk^{(i)}_{\mathsf{pos}}, M^{(i)}) \text{ for } i = 1, \ldots, k, \text{ and}\\
(com_{\mathsf{gbc}}, open_{\mathsf{gbc}}) &\leftarrow \mathsf{TC}\gamma.\mathsf{Com}(ck_{\mathsf{gbc}}, (sk_{\mathsf{pos}}, osk^{(1)}_{\mathsf{pos}}, \cdots, osk^{(k)}_{\mathsf{pos}})).
\end{aligned}
$$

Output $com := com_{\mathsf{gbc}}$ and $open := (open_{\mathsf{gbc}}, vk_{\mathsf{pos}}, ovk^{(1)}_{\mathsf{pos}}, \cdots, ovk^{(k)}_{\mathsf{pos}}, \sigma^{(1)}_{\mathsf{pos}}, \cdots, \sigma^{(k)}_{\mathsf{pos}})$.

$\mathsf{Vrf}(ck, com, M, open)$: Parse $com = com_{\mathsf{gbc}}$, $M = (M^{(1)}, \cdots, M^{(k)}) \in (\mathcal{M}_{\mathsf{pos}})^k$ and $open = (open_{\mathsf{gbc}}, vk_{\mathsf{pos}}, ovk^{(1)}_{\mathsf{pos}}, \cdots, ovk^{(k)}_{\mathsf{pos}}, \sigma^{(1)}_{\mathsf{pos}}, \cdots, \sigma^{(k)}_{\mathsf{pos}})$. Execute

$$
\begin{aligned}
b_0 &\leftarrow \mathsf{TC}\gamma.\mathsf{Vrf}(ck_{\mathsf{gbc}}, com_{\mathsf{gbc}}, (vk_{\mathsf{pos}}, ovk^{(1)}_{\mathsf{pos}}, \cdots, ovk^{(k)}_{\mathsf{pos}}), open_{\mathsf{gbc}}), \text{ and}\\
b_i &\leftarrow \mathsf{POS.Vrf}(vk_{\mathsf{pos}}, ovk^{(i)}_{\mathsf{pos}}, M^{(i)}, \sigma^{(i)}_{\mathsf{pos}}) \text{ for } i = 1, \ldots, k.
\end{aligned}
$$

Output 1 if $b_i = 1$ for all $i = 0, \ldots, k$. Output 0, otherwise.

$\mathsf{SimCom}(gk)$: Take gk_{gbc} from gk and run $(com_{\mathsf{gbc}}, ek_{\mathsf{gbc}}) \leftarrow \mathsf{TC}\gamma.\mathsf{SimCom}(gk_{\mathsf{gbc}})$ and output $com := com_{\mathsf{gbc}}$ and $ek := (com_{\mathsf{gbc}}, ek_{\mathsf{gbc}})$.

$\mathsf{Equiv}(M, ek, tk)$: The same as $\mathsf{TC.Com}$ except that, $\mathsf{TC}\gamma.\mathsf{Com}$ is replaced by $open_{\mathsf{gbc}} \leftarrow \mathsf{TC}\gamma.\mathsf{Equiv}((vk_{\mathsf{pos}}, ovk^{(1)}_{\mathsf{pos}}, \cdots, ovk^{(k)}_{\mathsf{pos}}), ek_{\mathsf{gbc}}, tk_{\mathsf{gbc}})$ and com_{gbc} included in ek.

Theorem 4. *The commitment scheme TC described above is CMTCR if POS is OT-NACMA, and $\mathsf{TC}\gamma$ is γ-target collision resistant.*

Proof. We follow the game transition framework. Let Game 0 be the CMTCR game launched by adversary $\mathcal{A}$. By $com^\dagger = com^\dagger_{\mathsf{gbc}}$, $open^\dagger = (open^\dagger_{\mathsf{gbc}}, vk^\dagger_{\mathsf{pos}}, ovk^{\dagger\ (1)}_{\mathsf{pos}}, \cdots, ovk^{\dagger\ (k)}_{\mathsf{pos}}, \sigma^{\dagger\ (1)}_{\mathsf{pos}}, \cdots, \sigma^{\dagger\ (k)}_{\mathsf{pos}})$ and $M^\dagger = (M^{\dagger(1)}, \cdots, M^{\dagger(k)})$, we denote the collision $\mathcal{A}$ outputs.

In Game 1, abort if $(vk^\dagger_{\mathsf{pos}}, ovk^{\dagger\ (1)}_{\mathsf{pos}}, \cdots, ovk^{\dagger\ (k)}_{\mathsf{pos}})$ differs from any of $(vk_{\mathsf{pos}}, ovk^{(1)}_{\mathsf{pos}}, \cdots, ovk^{(k)}_{\mathsf{pos}})$ observed by the signing oracle. We show that this occurs only if $\mathsf{TC}\gamma$ is broken by constructing adversary $\mathcal{B}$ attacking the γ-target collision resistance of $\mathsf{TC}\gamma$. Adversary $\mathcal{B}$ is given ck_{gbc} and q_s reference commitments com_{gbc} and opening $open_{\mathsf{gbc}}$ for random messages of the form $(sk_{\mathsf{pos}}, osk^{(1)}_{\mathsf{pos}}, \ldots, osk^{(k)}_{\mathsf{pos}})$. Each message is uniquely mapped to $(vk_{\mathsf{pos}}, ovk^{(1)}_{\mathsf{pos}}, \ldots, ovk^{(k)}_{\mathsf{pos}})$ by bijection γ. Adversary $\mathcal{B}$ invokes $\mathcal{A}$ with $ck := ck_{\mathsf{gbc}}$ as input. For every commitment query M, adversary $\mathcal{B}$ takes a fresh sample $(sk_{\mathsf{pos}}, osk^{(1)}_{\mathsf{pos}}, \ldots, osk^{(k)}_{\mathsf{pos}})$ with its commitment com_{gbc} and opening $open_{\mathsf{gbc}}$, and compute $\sigma^{(j)}_{\mathsf{pos}} \leftarrow \mathsf{POS.Sign}(sk_{\mathsf{pos}}, osk^{(j)}_{\mathsf{pos}}, M^{(j)})$ for $j = 1, \ldots, k$. It then returns $com := com_{\mathsf{gbc}}$ and $open := (open_{\mathsf{gbc}}, vk_{\mathsf{pos}}, ovk^{(1)}_{\mathsf{pos}}, \cdots, ovk^{(k)}_{\mathsf{pos}}, \sigma^{(1)}_{\mathsf{pos}}, \cdots, \sigma^{(k)}_{\mathsf{pos}})$. If $\mathcal{A}$ eventually outputs a collision, $\mathcal{B}$ outputs $com^\star_{\mathsf{gbc}} := com^\dagger_{\mathsf{gbc}}$, $open^\star_{\mathsf{gbc}} := open^\dagger_{\mathsf{gbc}}$ and $M^\star := (vk^\dagger_{\mathsf{pos}}, ovk^{\dagger\ (1)}_{\mathsf{pos}}, \cdots, ovk^{\dagger\ (k)}_{\mathsf{pos}})$. This completes the description of $\mathcal{B}$.

The simulated commitments and openings distribute the same as the real ones since every $osk^{(j)}_{\mathsf{pos}}$ is sampled legitimately by the challenger and the commitment generation procedure is the genuine one. Furthermore, the output of $\mathcal{B}$ is a valid collision against $\mathsf{TC}\gamma$ since $\mathcal{A}$ must have chosen $com^\dagger (= com^\dagger_{\mathsf{gbc}})$

from once used commitments and $M^\star$ is fresh due to the condition of abort. Accordingly, we have $|\Pr[\text{Game } 0] - \Pr[\text{Game } 1]| \leq \mathrm{Adv}^{\mathsf{tcr}}_{\mathsf{TC}\gamma,\mathcal{B}}(\lambda)$.

We then argue that $\mathcal{A}$ wins in Game 1 only if POS is broken. Let $\mathcal{C}$ be an adversary attacking the OT-NACMA property of POS. Given $vk^\star_{\mathsf{pos}}$ from outside, $\mathcal{C}$ first flips a coin $i^\dagger \leftarrow \{1,\ldots,q_s\}$. It then takes gk from $vk^\star_{\mathsf{pos}}$ and executes $(ck_{\mathsf{gbc}}, tk_{\mathsf{gbc}}) \leftarrow \mathsf{TC}\gamma.\mathsf{Key}(gk)$. Then it invokes $\mathcal{A}$ with input $ck := ck_{\mathsf{gbc}}$. Given j-th query for $j \neq i^\dagger$, $\mathcal{C}$ runs the legitimate procedure of $\mathsf{TC.Com}$ and returns obtained $(open, com)$. For the $i^\dagger$-th query $M = (M^{(1)},\ldots,M^{(k)})$, $\mathcal{C}$ makes a query $M^{(j)}$ to the signing oracle of POS and obtains $ovk^{(j)}_{\mathsf{pos}}$ and $\sigma^{(j)}_{\mathsf{pos}}$ for $j = 1,\ldots,k$. $\mathcal{C}$ then computes $(com_{\mathsf{gbc}}, ek_{\mathsf{gbc}}) \leftarrow \mathsf{TC}\gamma.\mathsf{SimCom}(gk)$ and $open_{\mathsf{gbc}} \leftarrow \mathsf{TC}\gamma.\mathsf{Equiv}((vk^\star_{\mathsf{pos}}, {ovk_{\mathsf{pos}}}^{(1)}, \cdots, {ovk_{\mathsf{pos}}}^{(k)}), ek_{\mathsf{gbc}}, tk_{\mathsf{gbc}})$ and outputs $com := com_{\mathsf{gbc}}$ and $open := (open_{\mathsf{gbc}}, vk^\star_{\mathsf{pos}}, ovk^{(1)}_{\mathsf{pos}}, \cdots, ovk^{(k)}_{\mathsf{pos}}, \sigma^{(1)}_{\mathsf{pos}}, \cdots, \sigma^{(k)}_{\mathsf{pos}})$. On receiving a collision from $\mathcal{A}$, $\mathcal{C}$ aborts if $vk^\dagger_{\mathsf{pos}} \neq vk^\star_{\mathsf{pos}}$. Otherwise, find $i^\star$ that $M^{\dagger(i^\star)} \neq M^{(i^\star)}$ (such an index must exist since $M^\dagger$ differs from any queried messages) and outputs $ovk^\star_{\mathsf{pos}} := {ovk^\dagger_{\mathsf{pos}}}^{(i^\star)}$ and $M^\star := M^{\dagger(i^\star)}$. This completes the description of $\mathcal{C}$. The simulated signatures are statistically close to the real ones due to the statistical trapdoor property of $\mathsf{TC}\gamma.\mathsf{SimCom}$ and $\mathsf{TC}\gamma.\mathsf{Equiv}$. Aborting event $vk^\dagger_{\mathsf{pos}} \neq vk^\star_{\mathsf{pos}}$ does not occur with probability $1/q_s$. Thus, we have $\frac{1}{q_s}\Pr[\text{Game } 1] - \epsilon_{\mathsf{sim}} \leq \mathrm{Adv}^{\mathsf{ot\text{-}nacma}}_{\mathsf{POS},\mathcal{C}}(\lambda)$, where ϵ_{sim} is the statistical loss by $\mathsf{TC}\gamma.\mathsf{SimCom}$ and $\mathsf{TC}\gamma.\mathsf{Equiv}$.

All in all, we have

$$\mathrm{Adv}^{\mathsf{cmtcr}}_{\mathsf{TC},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathsf{tcr}}_{\mathsf{TC}\gamma,\mathcal{B}}(\lambda) + q_s \cdot \mathrm{Adv}^{\mathsf{ot\text{-}nacma}}_{\mathsf{POS},\mathcal{C}}(\lambda) + \epsilon_{\mathsf{sim}},$$

which proves the statement. ∎

The following is immediate from the construction. In particular, Correctness holds due to the correctness of $\mathsf{TC}\gamma$ and POS and the existence of a bijection from the secret keys of POS to the verification keys.

Theorem 5. *Above TC is a structure-preserving trapdoor commitment scheme if $\mathsf{TC}\gamma$ is structure-preserving with respect to verification, and POS is structure-preserving.*

5 Fully Structure-Preserving Signatures

We argue that constructing an FSPS requires a different approach than those for all known constructions of SPSs. The verification equations of existing structure-preserving constant-size signatures on message vectors $(G^{m_1},\ldots,G^{m_n})$ involve pairings such as $\prod e(G^{x_i}, G^{m_i})$, where G^{x_i} is a public key element and G^{m_i} is a message element. The message is squashed into a signature element, say S, in such a form that $S := A \cdot \prod_{i=1}^{n} G^{m_i x_i}$ where x_i is a signing key component and A is computed from inputs other than the message. Such a structure requires either m_i or x_i to be detected to the signing algorithm. In FSPS, however, neither is given to the signing function.

Our starting point is the FSPS scheme in Section 3.3. The following sections present constructions that upgrade the security to UF-CMA by incorporating one-time signatures or trapdoor commitments.

5.1 Warm-Up

Our first approach is to take x_i from randomness instead of the signing key. That is, x_i works as a random one-time key and G^{x_i} is regarded as a one-time public key, which is then authenticated by an FSPS that is secure against extended random message attacks. This results in a combination of a weaker signature scheme with OTS, which is well known as a method for upgrading the security of the underlying signature scheme. This in fact can be seen as a special case of the construction of SPS by Abe et al. [2]. We nevertheless work out the scheme in detail to discuss our motivation for our main scheme and settle a basis for comparison. Let OTS and xSIG be a one-time and an ordinary signature scheme that have common setup function Setup. We construct $\mathsf{FSP1}$ as follows.

[**Signature Scheme: FSP1**]

$\mathsf{Setup}(1^\lambda)$: It is the same as Setup for OTS and xSIG. It outputs $gk \leftarrow \mathsf{Setup}(1^\lambda)$, and sets $\mathcal{M}_{\mathsf{xsig}} := \mathcal{K}^{vk}_{\mathsf{ots}}$ and $\mathcal{M} := \mathcal{M}_{\mathsf{ots}}$.

$\mathsf{Key}(gk)$: Run $(vk_{\mathsf{xsig}}, sk_{\mathsf{xsig}}) \leftarrow \mathsf{xSIG.Key}(gk)$. (It is assumed that gk is included in vk_{xsig} and sk_{xsig}.) Output $(vk, sk) := (vk_{\mathsf{xsig}}, sk_{\mathsf{xsig}})$.

$\mathsf{Sign}(sk, M)$: Take sk_{xsig} and gk from sk. Compute

$$\begin{aligned}(ovk_{\mathsf{ots}}, osk_{\mathsf{ots}}) &\leftarrow \mathsf{OTS.Key}(gk),\\ \sigma_{\mathsf{xsig}} &\leftarrow \mathsf{xSIG.Sign}(sk_{\mathsf{xsig}}, ovk_{\mathsf{ots}}), \text{ and}\\ \sigma_{\mathsf{ots}} &\leftarrow \mathsf{OTS.Sign}(osk_{\mathsf{ots}}, M).\end{aligned}$$

Output $\sigma := (\sigma_{\mathsf{xsig}}, \sigma_{\mathsf{ots}}, ovk_{\mathsf{ots}})$

$\mathsf{Vrf}(vk, M, \sigma)$: : Take vk_{xsig} and $(\sigma_{\mathsf{xsig}}, \sigma_{\mathsf{ots}}, ovk_{\mathsf{ots}})$ from the input. Output 1 if

$$1 = \mathsf{OTS.Vrf}(vk_{\mathsf{ots}}, M, \sigma_{\mathsf{ots}}) \quad \text{and} \quad 1 = \mathsf{xSIG.Vrf}(vk_{\mathsf{xsig}}, vk_{\mathsf{ots}}, \sigma_{\mathsf{xsig}}).$$

Output 0, otherwise.

Theorem 6. *If OTS is a UF-NACMA secure SPS and xSIG is a UF-XRMA secure FSPS, then FSP1 is a UF-CMA secure FSPS scheme.*

Proof. Since the syntactical consistency and correctness are trivial from the construction, we only show that the scheme is fully structure-preserving. The public component of $\mathsf{FSP1}$ is $(vk, \sigma, M) = (vk_{\mathsf{xsig}}, (\sigma_{\mathsf{xsig}}, \sigma_{\mathsf{ots}}, ovk_{\mathsf{ots}}), M)$, which consists of public components of $\mathsf{xSIG.Key}$ and the OTS. Also, the signing key of $\mathsf{FSP1}$ consists of sk_{xsig}. Thus, both public and private components of $\mathsf{FSP1}$ consist of group elements since xSIG is FSPS and the OTS is SPS. Furthermore, $\mathsf{FSP1.Vrf}$ evaluates $\mathsf{OTS.Vrf}$ and $\mathsf{xSIG.Vrf}$ that evaluate PPEs. Thus, $\mathsf{FSP1}$ is FSPS.

We next prove the UF-CMA security of $\mathsf{FSP1}$ by following the standard game transition technique. Let $\mathcal{A}$ be an adversary against $\mathsf{FSP1}$. By $\Pr[\text{Game } i]$ we denote probability that $\mathcal{A}$ eventually outputs a valid forgery as defined in Definition 2. Let Game 0 be the UF-CMA game that $\mathcal{A}$ is playing. By definition, $\Pr[\text{Game } 0] = \mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{FSP1},\mathcal{A}}(\lambda)$. Let $(\sigma^\dagger, m^\dagger)$ be a forgery $\mathcal{A}$ outputs. Let $\sigma^\dagger := (\sigma^\dagger_{\mathsf{xsig}}, \sigma^\dagger_{\mathsf{ots}}, vk^\dagger_{\mathsf{ots}})$.

In Game 1, abort the game if $(\sigma^\dagger, m^\dagger)$ is a valid forgery and $vk^\dagger_{\mathsf{ots}}$ is never used by the signing oracle. We show that this event occurs only if the UF-XRMA security of xSIG is broken. Let $\mathcal{B}$ be an adversary against xSIG launching an XRMA attack. $\mathcal{B}$ is given $(vk_{\mathsf{xsig}}, (\sigma^{(1)}_{\mathsf{xsig}}, vk^{(1)}_{\mathsf{ots}}, \omega^{(1)}), \ldots, (\sigma^{(q_s)}_{\mathsf{xsig}}, vk^{(q_s)}_{\mathsf{ots}}, \omega^{(q_s)}))$ where $\omega^{(j)}$ is the randomness used to generate $vk^{(j)}_{\mathsf{ots}}$ with $\mathsf{OTS.Key}$. $\mathcal{B}$ first obtains $sk^{(j)}_{\mathsf{ots}}$ from $\omega^{(j)}$ by executing $\mathsf{OTS.Key}$ by itself. Then it invokes $\mathcal{A}$ with input $vk := vk_{\mathsf{xsig}}$. On receiving $m^{(j)}$ for signing, $\mathcal{B}$ computes $\sigma^{(j)}_{\mathsf{ots}} \leftarrow \mathsf{OTS.Sign}(sk_{\mathsf{ots}}, m^{(j)})$ and returns $\sigma^{(j)} := (\sigma^{(j)}_{\mathsf{xsig}}, \sigma^{(j)}_{\mathsf{ots}}, vk^{(j)}_{\mathsf{ots}})$. When $\mathcal{A}$ outputs forgery $\sigma^\dagger := (\sigma^\dagger_{\mathsf{xsig}}, \sigma^\dagger_{\mathsf{ots}}, vk^\dagger_{\mathsf{ots}})$, $\mathcal{B}$ outputs $\sigma^\star_{\mathsf{xsig}} := \sigma^\dagger_{\mathsf{xsig}}$ and $m^\star := vk^\dagger_{\mathsf{ots}}$. This is a valid forgery since $vk^\dagger_{\mathsf{ots}} \neq vk^{(j)}_{\mathsf{ots}}$. Thus, we have $|\Pr[\text{Game } 0] - \Pr[\text{Game } 1]| \leq \mathrm{Adv}^{\mathsf{uf\text{-}xrma}}_{\mathsf{xSIG},\mathcal{B}}(\lambda)$.

Next we show that $\mathcal{A}$ wins Game 1 only if OTS is broken. Let $\mathcal{C}$ be an adversary attacking OTS with NACMA. Given gk from outside, $\mathcal{C}$ first flips a coin $i^\dagger \leftarrow \{1, \ldots, q_s\}$. It then executes $(vk, sk) \leftarrow \mathsf{FSP1.Key}(gk)$. Given $m^{(j)}$ for $j \neq i^\dagger$, $\mathcal{C}$ runs $\sigma^{(j)} \leftarrow \mathsf{FSP1.Sign}(sk, m^{(j)})$ and returns $\sigma^{(j)}$ to $\mathcal{A}$. For $j = i^\dagger$, $\mathcal{C}$ forwards $m^{(i^\dagger)}$ to the signing oracle of OTS and receive $\sigma^{(i^\dagger)}_{\mathsf{ots}}$ and $vk^{(i^\dagger)}_{\mathsf{ots}}$. Then $\mathcal{B}$ executes $\sigma^{(i^\dagger)}_{\mathsf{xsig}} \leftarrow \mathsf{xSIG.Sign}(sk_{\mathsf{xsig}}, vk^{(i^\dagger)}_{\mathsf{ots}})$ and returns $\sigma^{(i^\dagger)} := (\sigma^{(i^\dagger)}_{\mathsf{xsig}}, \sigma^{(i^\dagger)}_{\mathsf{ots}}, vk^{(i^\dagger)}_{\mathsf{ots}})$ to $\mathcal{A}$. When $\mathcal{A}$ outputs forgery $\sigma^\dagger := (\sigma^\dagger_{\mathsf{xsig}}, \sigma^\dagger_{\mathsf{ots}}, vk^\dagger_{\mathsf{ots}})$ and $m^\dagger$, $\mathcal{C}$ aborts if $vk^\dagger_{\mathsf{ots}} \neq vk^{(i^\dagger)}_{\mathsf{ots}}$. Otherwise, $\mathcal{C}$ outputs $\sigma^\star_{\mathsf{xsig}} := \sigma^\dagger_{\mathsf{ots}}$ and $m^\star := m^\dagger$. This is a valid forgery since $m^\dagger \neq m^{(i^\dagger)}$. Thus, we have $\frac{1}{q_s}\Pr[\text{Game } 1] \leq \mathrm{Adv}^{\mathsf{uf\text{-}nacma}}_{\mathsf{OTS},\mathcal{C}}(\lambda)$.

In total, we have

$$\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{FSP1},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathsf{uf\text{-}xrma}}_{\mathsf{xSIG},\mathcal{B}}(\lambda) + q_s \cdot \mathrm{Adv}^{\mathsf{uf\text{-}nacma}}_{\mathsf{OTS},\mathcal{C}}(\lambda),$$

which proves the statement. ∎

Though the above reduction involves a loss factor of q_s, it will vanish if OTS is based on a random-self reducible problem like SDP.

The above construction requires $\mathcal{K}^{vk}_{\mathsf{ots}}$ to match $\mathcal{M}_{\mathsf{xsig}}$. When they are instantiated with the concrete schemes from previous sections (using the POS in Section 3.2 as OTS by swapping $\mathbb{G}_1$ and $\mathbb{G}_2$, and using xSIG in Section 3.3), the space adjustment is done as follows.

[**Procedure: Matching $\mathcal{K}^{vk}_{\mathsf{ots}}$ to $\mathcal{M}_{\mathsf{xsig}}$**]

Setup: It runs xSIG.Setup and sets $(F_1, \tilde{F}_1)$ as default generators $(G, \tilde{G})$ for OTS. It also provide extra generators $(F_2, U_1, \ldots, U_{\ell+2})$ to OTS for the following procedures to work.

OTS.Key: It runs POS.Key and POS.Ovk in sequence and set $vk_{\mathsf{ots}} := (vk_{\mathsf{pos}}, ovk_{\mathsf{pos}})$. The key spaces are adjusted as follows.

- POS.Key On top of the legitimate procedure with $G := F_1$ to obtain $(G^{w_z}, G^{\chi_1}, \ldots, G^{\chi_\ell})$, it computes the extended part as $G_{i2} := F_2^{\chi_i}$ $G_{i3} := U_i^{\chi_i}$ for $i = 1, \ldots, \ell$, and $G_{z2} := F_2^{w_z}$, $G_{z3} := U_{\ell+1}^{w_z}$, and include all of them to vk_{pos}.
- POS.Ovk On top of legitimate procedure with $G := F_1$ that computes $A := G^a$, it computes extra parts $A_2 := F_2^a$ and $A_3 := U_{\ell+2}^a$ and includes them to ovk_{pos}.

Then those extended vk_{pos} and ovk_{pos} constitute a message $((G_z, G_{z2}, G_{z3}), (G_1, G_{12}, G_{13}), \ldots, (G_\ell, G_{\ell 2}, G_{\ell 3}), (A, A_2, A_3))$ given to xSIG to sign. We present a summary of the resulting instantiation of FSP1 below.

$$
\begin{aligned}
\text{Common Parameter} \quad & (G, \tilde{G}, F_1, F_2, \tilde{F}_1, \tilde{F}_2, \{U_i, \tilde{U}_i\}_{i=1}^{\ell+2}) \\
\text{Public-key} \quad & (\tilde{V}_1, \tilde{V}_2, \tilde{V}_3, \tilde{V}_4, \tilde{V}_5, \tilde{V}_6, V_7, \tilde{V}_8) \\
\text{Secret-key} \quad & (K_1, K_2, K_3, K_4) \\
\text{Message} \quad & (M_1, \ldots, M_\ell) \\
\text{Signature} \quad & (\tilde{S}_0, S_1, \ldots, S_5, \tilde{A}, \tilde{A}_2, \tilde{A}_3, \tilde{G}_z, \tilde{G}_{z2}, \tilde{G}_{z3}, \{\tilde{G}_i, \tilde{G}_{i2}, \tilde{G}_{i3}\}_{i=1}^{\ell}, Z, R) \\
\text{Verification PPEs} \quad & e(G, \tilde{A}) = e(Z, \tilde{G}_z)\, e(R, \tilde{G}) \textstyle\prod_{i=1}^{\ell} e(M_i, \tilde{G}_i), \\
& e(S_5, \tilde{V}_6\, \tilde{A}_3\, \tilde{G}_{z3} \textstyle\prod_{i=1}^{\ell} \tilde{G}_{i3}) = e(G, \tilde{S}_0), \\
& e(S_1, \tilde{V}_1)\, e(S_2, \tilde{V}_3)\, e(S_3, \tilde{V}_2) = e(S_4, \tilde{V}_4)\, e(S_5, \tilde{V}_5)\, e(V_7, \tilde{V}_8), \\
& e(F_1, \tilde{A}_3) = e(U_{\ell+2}, \tilde{A}), \qquad e(F_2, \tilde{A}_3) = e(U_{\ell+2}, \tilde{A}_2) \\
& e(F_1, \tilde{G}_{z3}) = e(U_{\ell+1}, \tilde{G}_z), \quad e(F_2, \tilde{G}_{z3}) = e(U_{\ell+1}, \tilde{G}_{z2}) \\
& e(F_1, \tilde{G}_{i3}) = e(U_i, \tilde{G}_i), \qquad e(F_2, \tilde{G}_{i3}) = e(U_i, \tilde{G}_{i2}) \text{ (for } i = 1, \ldots, \ell).
\end{aligned}
$$

Motivation for Improvement. Since an SPS is an OTS, construction FSP1 can be seen as a generic conversion from any SPS to an FSPS. In exchange for the generality, the construction has several shortcomings when instantiated with current building blocks.

- ($O(|m|)$-size signatures) The resulting signature σ includes the one-time verification key ovk_{ots}, which is linear in the size of messages in all current instantiations of OTS.
- (Factor 3 expansion in xSIG) As shown above, the message space of xSIG must cover ovk_{ots}, which is linear in the size of the message. Even worse, the currently known instantiation of xSIG suffers from an expansion factor of $\mu = 3$ for messages. That is, to sign a message consisting of a group element, say G^x, it requires to represent the message with two more extra elements F_2^x and U_i^x for given bases F_2 and U_i. Thus, the size of ovk_{ots} will actually be μ times larger than the one-time verification key that OTS originally requires.

The above shortcomings amplify each other. Finding an instantiation of xSIG with a smaller expansion factor is one direction of improvement. We leave it as an interesting open problem and focus on a generic approach in the next section.

5.2 Main Construction

Our idea is to avoid signing any components whose size grows to that of messages directly with xSIG. We achieve this by committing to the message using a shrinking commitment scheme and signing the commitment with xSIG. Again, combining a trapdoor commitment scheme (or a chameleon hash) and a signature scheme to achieve such an improvement is ultimately a known approach. What is important here is the security required from each building block. We show that chosen-message target collision resistance is sufficient for TC to reach UF-CMA in combination with an XRMA-secure signature scheme.

Let xSIG be a UF-XRMA secure FSPS scheme and TC be a CMTCR secure trapdoor commitment scheme with common setup function Setup. We construct our FSPS scheme FSP2 from xSIG and TC as follows.

[**Signature Scheme: FSP2**]

$\mathsf{Setup}(1^\lambda)$: Run common setup $gk \leftarrow \mathsf{Setup}(1^\lambda)$ and output gk. Set the message spaces $\mathcal{M}_{\mathsf{xsig}} := \mathcal{C}_{\mathsf{tc}}$ and $\mathcal{M} := \mathcal{M}_{\mathsf{tc}}$.

$\mathsf{Key}(gk)$: Run $(vk_{\mathsf{xsig}}, sk_{\mathsf{xsig}}) \leftarrow \mathsf{xSIG.Key}(gk)$, and $(ck_{\mathsf{tc}}, tk_{\mathsf{tc}}) \leftarrow \mathsf{TC.Key}(gk)$. Set $vk := (vk_{\mathsf{xsig}}, ck_{\mathsf{tc}})$, $sk := sk_{\mathsf{xsig}}$. Output (vk, sk)

$\mathsf{Sign}(sk, M)$: Parse sk into sk_{xsig}. Run

$$(com_{\mathsf{tc}}, open_{\mathsf{tc}}) \leftarrow \mathsf{TC.Com}(ck_{\mathsf{tc}}, M) \text{ and}$$
$$\sigma_{\mathsf{xsig}} \leftarrow \mathsf{xSIG.Sign}(sk_{\mathsf{xsig}}, com_{\mathsf{tc}}).$$

Output $\sigma := (\sigma_{\mathsf{xsig}}, open_{\mathsf{tc}}, com_{\mathsf{tc}})$

$\mathsf{Vrf}(vk, M, \sigma)$: Parse $vk = (vk_{\mathsf{xsig}}, ck_{\mathsf{tc}})$ and $\sigma = (\sigma_{\mathsf{xsig}}, open_{\mathsf{tc}}, com_{\mathsf{tc}})$. Output 1 if $1 = \mathsf{TC.Vrf}(ck_{\mathsf{tc}}, com_{\mathsf{tc}}, M, open_{\mathsf{tc}})$ and $1 = \mathsf{xSIG.Vrf}(vk_{\mathsf{xsig}}, com_{\mathsf{tc}}, \sigma_{\mathsf{xsig}})$. Output 0, otherwise.

Theorem 7. *If TC is a CMTCR secure SPTC, and xSIG is a UF-XRMA secure FSPS relative to TC.SimCom as a message sampler, then FSP2 is a UF-CMA FSPS.*

Proof. Correctness holds trivially from those of the underlying TC and xSIG. Regarding the full structure-preserving property, observe that sk consists of sk_{xsig}, which that are source group elements since xSIG is fully structure-preserving. The same is true for public components, i.e., public keys, messages, and signatures. The verification only evaluates verification functions of these underlying building blocks, which evaluate PPEs. Thus, FSP2 is FSPS.

We next prove the security property. Let $\mathcal{A}$ be an adversary against FSP2. Let Game 0 be the UF-CMA game that $\mathcal{A}$ is playing. By definition, $\Pr[\text{Game 0}] = \mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{FSP2},\mathcal{A}}(\lambda)$. Let $(\sigma^\dagger, m^\dagger)$ be a forgery $\mathcal{A}$ outputs. Let $\sigma^\dagger := (\sigma^\dagger_{\mathsf{xsig}}, open^\dagger_{\mathsf{tc}}, com^\dagger_{\mathsf{tc}})$.

In Game 1, abort the game if $(\sigma^\dagger, m^\dagger)$ is a valid forgery and $com^\dagger_{\mathsf{tc}}$ is never viewed by the signing oracle. We show that this event occurs only if the UF-XRMA security of xSIG is broken. Let $\mathcal{B}$ be an adversary against xSIG launching an XRMA attack. The message sampler for XRMA is TC.SimCom. That is, the challenger samples random messages by $(com_{\mathsf{tc}}, ek_{\mathsf{tc}}) \leftarrow \mathsf{TC.SimCom}(gk; \omega)$ with random coin ω and gives com_{tc} and ω with signature σ_{xsig} on com_{tc} as a message. Let $sample^{(j)}$ be the j-th sample, i.e., $sample^{(j)} := (com^{(j)}_{\mathsf{tc}}, \omega^{(j)}, \sigma^{(j)}_{\mathsf{xsig}})$. Given $(vk_{\mathsf{xsig}}, sample^{(1)}, \ldots, sample^{(q_s)})$ as input, $\mathcal{B}$ runs as follows. It first takes gk from vk_{xsig} and recovers every $ek^{(j)}_{\mathsf{tc}}$ from $\omega^{(j)}$ by $(com_{\mathsf{tc}}, ek_{\mathsf{tc}}) \leftarrow \mathsf{TC.SimCom}(gk; \omega)$. It then runs $(ck_{\mathsf{tc}}, tk_{\mathsf{tc}}) \leftarrow \mathsf{TC.Key}(gk)$ and invokes $\mathcal{A}$ with input $vk := (vk_{\mathsf{xsig}}, ck_{\mathsf{tc}})$. Given the j-th signing query $m^{(j)}$ from $\mathcal{A}$, it executes $open^{(j)}_{\mathsf{tc}} \leftarrow \mathsf{TC.Equiv}(m^{(j)}, tk_{\mathsf{tc}}, ek^{(j)}_{\mathsf{tc}},)$ and returns $\sigma := (\sigma^{(j)}_{\mathsf{xsig}}, open^{(j)}_{\mathsf{tc}}, com^{(j)}_{\mathsf{tc}})$ to $\mathcal{A}$. If $\mathcal{A}$ eventually outputs a forgery, $\sigma^\dagger = (\sigma^\dagger_{\mathsf{xsig}}, open^\dagger_{\mathsf{tc}}, com^\dagger_{\mathsf{tc}})$ and $m^\dagger$, it outputs $\sigma^\star_{\mathsf{xsig}} := \sigma^\dagger_{\mathsf{xsig}}$ and $m^\star := com^\dagger_{\mathsf{tc}}$ as a forgery with respect to xSIG.

Correctness of the above reduction holds from statistically close distribution of simulated $com^{(j)}_{\mathsf{tc}}$, and $open^{(j)}_{\mathsf{tc}}$. The output $(\sigma^\star_{\mathsf{xsig}}, m^\star)$ is also a valid forgery since $com^\dagger_{\mathsf{tc}}$ differs from any $com^{(j)}_{\mathsf{tc}}$. Letting ϵ_{sim} denote the statistical distance, we have $|\Pr[\text{Game 0}] - \Pr[\text{Game 1}]| \leq \mathrm{Adv}^{\mathsf{uf\text{-}xrma}}_{\mathsf{xSIG},\mathcal{B}}(\lambda) + \epsilon_{\mathsf{sim}}$.

Now we claim that $\mathcal{A}$ winning in Game 1 occurs only if the CMTCR security of TC is broken. The reduction from successful $\mathcal{A}$ in Game 1 to adversary $\mathcal{C}$ that breaks TC is straightforward. Given ck_{tc}, $\mathcal{C}$ runs $(vk_{\mathsf{xsig}}, sk_{\mathsf{xsig}}) \leftarrow \mathsf{xSIG.Key}(gk)$ and invokes $\mathcal{A}$ with $vk := (vk_{\mathsf{xsig}}, ck_{\mathsf{tc}})$. Then, given message $m^{(j)}$, forward it to the oracle of TC and obtain $(com_{\mathsf{tc}}^{(j)}, open_{\mathsf{tc}}^{(j)})$. Then sign $com_{\mathsf{tc}}^{(j)}$ using sk_{xsig} to obtain $\sigma_{\mathsf{xsig}}^{(j)}$ and return $(\sigma_{\mathsf{xsig}}^{(j)}, open_{\mathsf{tc}}^{(j)}, com_{\mathsf{tc}}^{(j)})$ to $\mathcal{A}$. Given a forged signature $(\sigma_{\mathsf{xsig}}^{\dagger}, open_{\mathsf{tc}}^{\dagger}, com_{\mathsf{tc}}^{\dagger})$ and $m^{\dagger}$, output $open_{\mathsf{tc}}^{\star} := open_{\mathsf{tc}}^{\dagger}$ and $m^{\star} := m^{\dagger}$. It is a valid forgery since $m^{\dagger} \neq m^{(j)}$ for all j. We thus have $\Pr[\text{Game 1}] = \mathrm{Adv}^{\mathsf{cmtcr}}_{\mathsf{TC},\mathcal{C}}(\lambda)$.

By summing up the differences, we have

$$\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{FSP2},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{\mathsf{uf\text{-}xrma}}_{\mathsf{xSIG},\mathcal{B}}(\lambda) + \mathrm{Adv}^{\mathsf{cmtcr}}_{\mathsf{TC},\mathcal{C}}(\lambda) + \epsilon_{\mathsf{sim}}, \tag{15}$$

which proves the statement. ∎

To instantiate this construction with the building blocks from previous sections, we again need to duplicate $com_{\mathsf{gbc}} = \tilde{G}_u = \tilde{G}^{\zeta} \prod_{i=1}^{\ell} \tilde{X}_i^{m_i}$ to a triple with respect to bases $\tilde{G} = \tilde{F}_2$, $\tilde{F}_3$ and $\tilde{U}_1$ as follows. To be able to do so without holding the discrete logarithms of the $\tilde{X}_i$'s, we need to duplicate $\tilde{X}$ to the same set of bases as well. Details are shown below.

[**Procedure: Matching $\mathcal{C}_{\mathsf{gbc}}$ to $\mathcal{M}_{\mathsf{xsig}}$**]

Setup: It runs $\mathsf{xSIG.Setup}$ and sets $(F_1, \tilde{F}_1)$ as default generators $(G, \tilde{G})$ for $\mathsf{TC}\gamma$ with extra generators (F_2, U_1) as well.

$\mathsf{TC}\gamma\mathsf{.Key}$: On top of the legitimate procedure with $G := \tilde{F}_1$ to obtain $\tilde{X}_i := G^{\rho_i}$, additionally compute $\tilde{X}_{i2} := \tilde{F}_2^{\rho_i}$ and $\tilde{X}_{i3} := \tilde{U}_1^{\rho_i}$ for $i = 1, \ldots, \ell$ and include them to ck_{gbc}.

$\mathsf{TC}\gamma\mathsf{.Com}$: On top of the legitimate procedure that computes $\tilde{G}_u = \tilde{G}^{\zeta} \prod_{i=1}^{\ell} \tilde{X}_i^{m_i}$ for $\tilde{G} := \tilde{F}_1$, compute $\tilde{G}_{u2} := \tilde{F}_2^{\zeta} \prod_{i=1}^{\ell} \tilde{X}_{i1}^{m_i}$ and $\tilde{G}_{u3} := \tilde{U}_1^{\zeta} \prod_{i=1}^{\ell} \tilde{X}_{i3}^{m_i}$ and include them to com_{gbc}.

$\mathsf{TC}\gamma\mathsf{.SimCom}$: Compute the above extra components as $\tilde{G}_{u2} := \tilde{F}_2^{\omega_u}$, and $\tilde{G}_{u3} := U_1^{\omega_u}$.

The result is an extended commitment $com_{\mathsf{gbc}} = (\tilde{G}_u, \tilde{G}_{u2}, \tilde{G}_{u3})$ that matches to the message space of xSIG with $\ell = 1$. Note that the duplicated keys have no effect on the security of POS nor $\mathsf{TC}\gamma$ since they can be easily simulated when the discrete-logs of the extra bases to the original base $\tilde{G}$ are known.

We summarize the instantiation of $\mathsf{FSP2}$ in the following. Let $k = \lceil \frac{\ell}{\ell_{\mathsf{pos}}} \rceil$ and $\ell_{\mathsf{gbc}} = 1 + k + \ell_{\mathsf{pos}}$.

$$\begin{aligned}
\text{Common Parameter} \quad & (G, \tilde{G}, F_1, F_2, \tilde{F}_1, \tilde{F}_2, U_1, \tilde{U}_1) \\
\text{Public-key} \quad & (\tilde{V}_1, \tilde{V}_2, \tilde{V}_3, \tilde{V}_4, \tilde{V}_5, \tilde{V}_6, V_7, \tilde{V}_8, \{\tilde{X}_i, \tilde{X}_{i2}, \tilde{X}_{i3}\}_{i=1}^{\ell_{\mathsf{gbc}}}) \\
\text{Secret-key} \quad & (K_1, K_2, K_3, K_4) \\
\text{Message} \quad & (\tilde{M}_1, \ldots, \tilde{M}_{\ell}) \\
\text{Signature} \quad & (\tilde{S}_0, \ldots, S_5, \tilde{G}_u, \tilde{G}_{u2}, \tilde{G}_{u3}, R, G_z, G_1, \ldots, G_{\ell_{\mathsf{pos}}}, \{A_i, \tilde{Z}_i, \tilde{R}_i\}_{i=1}^{k}) \\
\text{Verification PPEs} \quad & \text{Let } (N_1, \ldots, N_{\ell_{\mathsf{gbc}}}) := (G_z, G_1, \ldots, G_{\ell_{\mathsf{pos}}}, A_1, \ldots, A_k). \\
& \text{For } j = 1, \ldots, k\text{:} \\
& \quad e(A_j, \tilde{G}) = e(G_z, \tilde{Z}_j)\, e(G, \tilde{R}_j) \prod_{i=1}^{\ell_{\mathsf{pos}}} e(G_i, \tilde{M}_{(j-1)\ell_{\mathsf{pos}}+i}), \\
& e(G, \tilde{G}_u) = e(R, \tilde{G}) \prod_{i=1}^{\ell_{\mathsf{gbc}}} e(N_i, \tilde{X}_i) \\
& e(S_5, \tilde{V}_6\, \tilde{G}_{u3}) = e(G, \tilde{S}_0), \\
& e(S_1, \tilde{V}_1)\, e(S_2, \tilde{V}_3)\, e(S_3, \tilde{V}_2) = e(S_4, \tilde{V}_4)\, e(S_5, \tilde{V}_5)\, e(V_7, \tilde{V}_8), \\
& e(F_1, \tilde{G}_{u3}) = e(U_1, \tilde{G}_u), \quad e(F_2, \tilde{G}_{u3}) = e(U_1, \tilde{G}_{u2}).
\end{aligned}$$

5.3 Efficiency

In this section, we assess the efficiency of FSP1 and FSP2 instantiated as described in Section 5.1 and 5.2. Note that FSP1 uses a one-time signature scheme, OTS, and we evaluate the efficiency where OTS is instantiated by POS in Section 3.2 since the POS is the best known OTS under a standard static assumption.

Signature Size and Number of PPEs. Here we assess the sizes of a key and a signature for unilateral messages consisting of ℓ group elements. By $|vk_{\mathsf{x}}|$ for $\mathsf{x} \in \{\mathsf{ots}, \mathsf{xsig}\}$, we denote the number of group elements in vk_{x} except for those in $|gk|$. By the term $\#\,\mathsf{PPE}_{\mathsf{x}}$ we denote the number of pairing product equations in the corresponding building block x. Table 1 summarizes the comparison with signature length for some concrete message lengths.

Scheme	$\lvert sk\rvert$	$\lvert vk\rvert$	$\lvert\sigma\rvert$						$\#\,\mathsf{PPE}$
			ℓ	$\ell=1$	4	9	25	100	
FSP1	4	$18+2\,\ell$	$14+3\,\ell$	17	26	41	89	314	$7+2\ell$
FSP2	4	$19+6\,\lceil\sqrt{\ell}\rceil$	$11+4\,\lceil\sqrt{\ell}\rceil$	15	19	23	31	51	$5+\lceil\sqrt{\ell}\rceil$

Table 1: Size of secret keys, verification keys, signatures, and number of PPEs in verification for unilateral messages of size ℓ.

- FSP1. According to the descriptions in Section 3.2 and Section 3.3, we have the following parameters for the building blocks.
  - OTS: $|vk_{\mathsf{ots}}| = |vk_{\mathsf{pos}}| + |ovk_{\mathsf{pos}}| = \ell + 2$, $|\sigma_{\mathsf{ots}}| = 2$, and $\#\,\mathsf{PPE}_{\mathsf{ots}} = 1$.
  - xSIG: $|sk_{\mathsf{xsig}}| = 4$, $|vk_{\mathsf{xsig}}| = 8$, and $\#\,\mathsf{PPE}_{\mathsf{xsig}} = 2 + 2\,|vk_{\mathsf{ots}}|$.

  The common setup function for these building blocks generates bases $(G, \tilde{G}, F_1, F_2, \tilde{F}_1, \tilde{F}_2, \{U_i, \tilde{U}_i\}_{i=1}^{\ell_{\mathsf{xsig}}})$ for $\ell_{\mathsf{xsig}} = |vk_{\mathsf{ots}}|$ to allow xSIG to sign vk_{ots}. (Note that vk_{ots} consists only of group elements from $\mathbb{G}_1$, which xSIG can sign.) Taking the message expansion factor $\mu = 3$ into account, we obtain the following for FSP1:

$$
\begin{aligned}
|gk| &= 6 + 2\,|vk_{\mathsf{ots}}| \\
|sk| &= |sk_{\mathsf{xsig}}| = 4 \\
|vk| &= |gk| + |vk_{\mathsf{xsig}}| = 18 + 2\ell \\
|\sigma| &= |\sigma_{\mathsf{xsig}}| + |\sigma_{\mathsf{ots}}| + \mu\,|vk_{\mathsf{ots}}| = 14 + 3\ell \\
\#\,\mathsf{PPE} &= \#\,\mathsf{PPE}_{\mathsf{xsig}} + \#\,\mathsf{PPE}_{\mathsf{ots}} = 7 + 2\,\ell
\end{aligned}
$$

- FSP2. The underlying components are xSIG, TCγ and POS. Since POS is repeatedly used in FSP2, its message size ℓ_{pos} can be set independently from the input message size ℓ. The parameters for these underlying components are:
  - POS: $|vk_{\mathsf{pos}}| = \ell_{\mathsf{pos}} + 1$, $|ovk_{\mathsf{pos}}| = 1$, $|\sigma_{\mathsf{pos}}| = 2$, and $\#\,\mathsf{PPE}_{\mathsf{pos}} = 1$.
  - TCγ: $|ck_{\mathsf{gbc}}| = |vk_{\mathsf{pos}}| + \lceil \ell/\ell_{\mathsf{pos}} \rceil \cdot |ovk_{\mathsf{pos}}| = 1 + \lceil \ell/\ell_{\mathsf{pos}} \rceil + \ell_{\mathsf{pos}}$, $|com_{\mathsf{gbc}}| = 1$, and $|open_{\mathsf{gbc}}| = 1$.
  - xSIG: $|sk_{\mathsf{xsig}}| = 4$, $|vk_{\mathsf{xsig}}| = 8$, and $\#\,\mathsf{PPE}_{\mathsf{xsig}} = 2 + 2\,|com_{\mathsf{gbc}}|$.

  As well as the previous case, the common setup function outputs gk including bases $(G, \tilde{G}, F_1, F_2, \tilde{F}_1, \tilde{F}_2, \{U_i, \tilde{U}_i\}_{i=1}^{\ell_{\mathsf{xsig}}})$ for $\ell_{\mathsf{xsig}} = |com_{\mathsf{gbc}}|$ to allow xSIG to sign com_{gbc}. Based on these parameters, the

following evaluation is obtained for FSP2:

$$\begin{aligned}
|sk| &= |sk_{\mathsf{xsig}}| = 4 \\
|vk| &= |gk| + |vk_{\mathsf{xsig}}| + |ck_{\mathsf{gbc}}| = 19 + 3\,\lceil \ell/\ell_{\mathsf{pos}} \rceil + 3\,\ell_{\mathsf{pos}} = 19 + 6\,\lceil \sqrt{\ell} \rceil \\
|\sigma| &= |\sigma_{\mathsf{xsig}}| + |open_{\mathsf{gbc}}| + |\sigma_{\mathsf{pos}}| + \mu |com_{\mathsf{gbc}}| + |vk_{\mathsf{pos}}| + \lceil \ell/\ell_{\mathsf{pos}} \rceil \cdot |ovk_{\mathsf{pos}}| \\
&= 11 + 3\,\lceil \ell/\ell_{\mathsf{pos}} \rceil + \ell_{\mathsf{pos}} = 11 + 4\,\lceil \sqrt{\ell} \rceil \\
\#\,\mathsf{PPE} &= \#\,\mathsf{PPE}_{\mathsf{xsig}} + \#\,\mathsf{PPE}_{\mathsf{gbc}} + \lceil \ell/\ell_{\mathsf{pos}} \rceil \cdot \#\,\mathsf{PPE}_{\mathsf{pos}} \\
&= 5 + \lceil \ell/\ell_{\mathsf{pos}} \rceil = 5 + \lceil \sqrt{\ell} \rceil
\end{aligned}$$

The last equality in each evaluation is obtained at the optimal setting; $\ell_{\mathsf{pos}} = \lceil \ell/\ell_{\mathsf{pos}} \rceil = \lceil \sqrt{\ell} \rceil$.

Proof Size for Knowing a Secret Key. Next we assess the cost for proving one's knowledge of a secret key for FSP1 and FSP2 with the Groth-Sahai proof as a non-interactive witness indistinguishable proof (NIWIPoK) or a zero-knowledge proof (NIZKPoK). Results are summarized in Table 2. In either scheme, a secret key comes only from xSIG, which is of the form (K_1, K_2, K_3, K_4).

- NIWIPoK: Relations to prove are in (8) that we recall as

$$\begin{aligned}
&e(\underline{K_2}, \tilde{G}) = e(G, \tilde{V}_1), \quad e(G, \tilde{V}_3) = e(\underline{K_2}, \tilde{V}_2), \quad e(\underline{K_1}, \tilde{V}_1) = e(V_7, \tilde{V}_8), \\
&e(\underline{K_2}, \tilde{V}_4) = e(G, \tilde{V}_5), \quad e(\underline{K_3}, \tilde{G})\, e(\underline{K_4}, \tilde{V}_2) = e(G, \tilde{V}_4),
\end{aligned} \tag{16}$$

  These are linear relations in $\mathbb{G}_1$ when proved with the Groth-Sahai proofs. Underlined variables are the witnesses the prover commits to. According to [33], committing to a group element in $\mathbb{G}_1$ (or $\mathbb{G}_2$) requires 2 elements in $\mathbb{G}_1$ (or $\mathbb{G}_2$, respectively). Proving a linear relation with a PPE yields a proof consisting of 2 group elements in $\mathbb{G}_2$. Thus, with 4 witnesses, and 5 linear relations, the resulting proof (i.e. commitments and proofs for all relations) consists of $4 \times 2 + 5 \times 2 = 18$ group elements (8 in $\mathbb{G}_1$ and 10 in $\mathbb{G}_2$).

- NIZKPoK: The above witness-indistinguishable proof is turned into zero-knowledge in the following manner. First, the prover commits to public-key elements V_7 and G and proves relations

$$\underline{W} = V_7 \quad \text{and} \quad \underline{V} = G. \tag{17}$$

  Committing to W and V costs $2 \times 2 = 4$ group elements in $\mathbb{G}_1$, and proving elations in (17) as multiscalar multiplication equations requires $2 \times 2 = 4$ scalar values in $\mathbb{Z}_p$. The prover also proves relations:

$$\begin{aligned}
&e(\underline{K_2}, \tilde{G}) = e(\underline{V}, \tilde{V}_1), \quad e(\underline{V}, \tilde{V}_3) = e(\underline{K_2}, \tilde{V}_2), \quad e(\underline{K_1}, \tilde{V}_1) = e(\underline{W}, \tilde{V}_8), \\
&e(\underline{K_2}, \tilde{V}_4) = e(\underline{V}, \tilde{V}_5), \quad e(\underline{K_3}, \tilde{G}) e(\underline{K_4}, \tilde{V}_2) = e(\underline{V}, \tilde{V}_4).
\end{aligned} \tag{18}$$

  Since all witnesses in (18) belong to $\mathbb{G}_1$, the cost for proving the relations is unchanged from that for (16). Thus the total cost is $18 + 4 = 22$ group elements (12 in $\mathbb{G}_1$ and 10 in $\mathbb{G}_2$) and 4 scalar values in $\mathbb{Z}_p$.

Proof Size for Knowing a Valid Signature. Here we assess the cost for proving possession of a valid signature using the Groth-Sahai proofs as NIWIPoK. The result is summarized in Table 2.

Scheme	WI (sk)	ZK (sk)	WI (σ)	ZK (σ)
FSP1	18	22	$54+10\ell$	$56+10\ell$
FSP2			$44+16\lceil\sqrt{\ell}\rceil$	$46+16\lceil\sqrt{\ell}\rceil$

Table 2: Number of group elements in the Groth-Sahai proofs for possession of a secret key and a signature for unilateral messages of size ℓ with the optimal parameter setting. For ZK, proofs actually include a small number of elements in $\mathbb{Z}_p$ omitted here.

- Case of FSP1. According to the descriptions in Section 5.1, a valid signature satisfies the following relations.

$$e(G,\tilde{\underline{A}}) = e(\underline{Z},\tilde{\underline{G}_z})\, e(\underline{R},\tilde{G}) \prod_{i=1}^{\ell} e(M_i,\tilde{\underline{G}_i}), \quad e(\underline{S_5},\tilde{V}_6\ \tilde{\underline{A}_3}\ \tilde{\underline{G}_{z3}} \prod_{i=1}^{\ell} \tilde{\underline{G}_{i3}}) = e(G,\tilde{\underline{S_0}}),$$

$$e(\underline{S_1},\tilde{V}_1)\, e(\underline{S_2},\tilde{V}_3)\, e(\underline{S_3},\tilde{V}_2) = e(\underline{S_4},\tilde{V}_4)\, e(\underline{S_5},\tilde{V}_5)\, e(V_7,\tilde{V}_8),$$

$$e(F_1,\tilde{\underline{A}_3}) = e(U_{\ell+2},\tilde{\underline{A}}), \quad e(F_2,\tilde{\underline{A}_3}) = e(U_{\ell+2},\tilde{\underline{A}_2}), \quad e(F_1,\tilde{\underline{G}_{z3}}) = e(U_{\ell+1},\tilde{\underline{G}_z}),$$

$$e(F_2,\tilde{\underline{G}_{z3}}) = e(U_{\ell+1},\tilde{\underline{G}_{z2}}), \quad e(F_1,\tilde{\underline{G}_{i3}}) = e(U_i,\tilde{\underline{G}_i}), \quad e(F_2,\tilde{\underline{G}_{i3}}) = e(U_i,\tilde{\underline{G}_{i2}})$$

  for $i = 1,\ldots,\ell$ for the last two relations. There are 7 underlined witnesses in $\mathbb{G}_1$ and $1+3(\ell+2)$ in $\mathbb{G}_2$. Committing to these witnesses requires 14 elements in $\mathbb{G}_1$ and $14+6\ell$ elements in $\mathbb{G}_2$. The first two relations involve witnesses in both groups whose proofs require 2×4 elements in $\mathbb{G}_1$ and $\mathbb{G}_2$. The third relation has witnesses only in $\mathbb{G}_1$. Its proof consists of 2 elements in $\mathbb{G}_2$. The remaining $4+2\ell$ relations have witnesses only in $\mathbb{G}_2$, and each of their proof costs 2 elements in $\mathbb{G}_1$. In total the proofs and commitments consist of $14+4\times 2+2\times(4+2\ell) = 30+4\ell$ elements in $\mathbb{G}_1$ and $14+6\ell+4\times 2+2 = 24+6\ell$ elements in $\mathbb{G}_2$, which sum up to $54+10\ell$ group elements.

- Case of FSP2. As described in Section 5.2, a valid signature satisfies the following relations:

$$e(\underline{A_j},\tilde{G}) = e(\underline{G_z},\tilde{\underline{Z}_j})\, e(G,\tilde{\underline{R}_j}) \prod_{i=1}^{\ell_{\mathsf{pos}}} e(\underline{G_i},\tilde{M}_{(j-1)\ell_{\mathsf{pos}}+i}) \text{ (for } j=1,\ldots,k),$$

$$e(G,\tilde{\underline{G}_u}) = e(\underline{R},\tilde{G}) \prod_{i=1}^{\ell_{\mathsf{gbc}}} e(\underline{N_i},\tilde{X}_i), \quad e(\underline{S_5},\tilde{V}_6\ \tilde{\underline{G}_{u3}}) = e(G,\tilde{\underline{S_0}}),$$

$$e(\underline{S_1},\tilde{V}_1)\, e(\underline{S_2},\tilde{V}_3)\, e(\underline{S_3},\tilde{V}_2) = e(\underline{S_4},\tilde{V}_4)\, e(\underline{S_5},\tilde{V}_5)\, e(V_7,\tilde{V}_8),$$

$$e(F_1,\tilde{\underline{G}_{u3}}) = e(U_1,\tilde{\underline{G}_u}), \quad e(F_2,\tilde{\underline{G}_{u3}}) = e(U_1,\tilde{\underline{G}_{u2}}).$$

  where $(N_1,\ldots,N_{\ell_{\mathsf{gbc}}})$ is actually $(G_z,G_1,\ldots,G_{\ell_{\mathsf{pos}}},A_1,\ldots,A_k)$ that are also witnesses. Thus we do not need to count the cost for committing to N_i. We consider $\ell_{\mathsf{gbc}} = k = \lceil\sqrt{\ell}\rceil$. A signature consists of $4+2\lceil\sqrt{\ell}\rceil$ elements in $\mathbb{G}_1$ and $7+2\lceil\sqrt{\ell}\rceil$ elements in $\mathbb{G}_2$. Thus committing to the signature costs $2(4+2\lceil\sqrt{\ell}\rceil)$ and $2(7+2\lceil\sqrt{\ell}\rceil)$ elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. The first three relations (indeed $\lceil\sqrt{\ell}\rceil+2$ relations) that came from POS and TCγ involve witnesses in both groups. Hence proofs for them cost $4(\lceil\sqrt{\ell}\rceil+2)$ elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. The remaining three relations that came from xSIG involves witnesses for either of $\mathbb{G}_1$ or $\mathbb{G}_2$. Proofs for those relations costs 2 group elements in $\mathbb{G}_2$ and 2×2 group elements in $\mathbb{G}_1$. In total the proofs and commitments consists of $2(4+2\lceil\sqrt{\ell}\rceil)+4(\lceil\sqrt{\ell}\rceil+2)+4 = 20+8\lceil\sqrt{\ell}\rceil$ and $2(7+2\lceil\sqrt{\ell}\rceil)+4(\lceil\sqrt{\ell}\rceil+2)+2 = 24+8\lceil\sqrt{\ell}\rceil$ in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. They sum up to $44+16\lceil\sqrt{\ell}\rceil$ group elements in total.

For either scheme, proving in zero-knowledge is possible only by additionally committing to V_7 and proving the correctness. It adds 2 elements in $\mathbb{G}_1$ for the commitment of V_7 and 2 $\mathbb{Z}_p$ elements as a proof.

5.4 Lower Bound on Signature Size and Verification key size

The signatures of our concrete FSPSs consist of $\Omega(\sqrt{\ell})$ group elements when signing ℓ-element messages. This may seem disappointing compared to previous constructions of SPS, which have generally achieved constant-size signatures, but we argue that, at least for our modular constructions of FSPS, the $\sqrt{\ell}$ factor is unavoidable. This is a consequence of the following new trade-off between signature and verification key size for arbitrary (possibly one-time) SPS schemes.

Theorem 8. *Consider a (possibly one-time) SPS scheme on messages in $\mathbb{G}_2^\ell$ in the asymmetric (Type III) bilinear group setting. Let κ be the number of verification key elements and σ the number of group elements in signatures. If the scheme is existentially unforgeable in a model in which the adversary has access to a valid signature on a known message and the scheme has an algebraic signing algorithm, we have $\kappa + \sigma \geq \sqrt{\ell}$.*

Proof. Denote by $(M_1, \ldots, M_\ell) \in \mathbb{G}_2^\ell$ the message vector, by $(U_1, \ldots, U_{\kappa_1}, V_1, \ldots, V_{\kappa_2}) \in \mathbb{G}_1^{\kappa_1} \times \mathbb{G}_2^{\kappa_2}$ $(\kappa_1 + \kappa_2 = \kappa)$ the verification key elements, and by $(R_1, \ldots, R_{\sigma_1}, S_1, \ldots, S_{\sigma_2}) \in \mathbb{G}_1^{\sigma_1} \times \mathbb{G}_2^{\sigma_2}$ $(\sigma_1 + \sigma_2 = \sigma)$ the signature elements. The corresponding discrete logarithms are written in lowercase letters.

Each verification equation of the scheme can be expressed as a bilinear relation between the discrete logarithms of the group elements in $\mathbb{G}_1$ (namely the U_i's and R_i's) on the one hand, and those of the elements in $\mathbb{G}_2$ (namely the M_i's, V_i's and S_i's) on the other. The i-th pairing product equation can thus be written in matrix form as:

$$X^T E_i Y = 0, \tag{19}$$

where X and Y are the column vectors given by

$$X = (r_1, \ldots, r_{\sigma_1}, u_1, \ldots, u_{\kappa_1}, 1)^T, \text{ and}$$
$$Y = (m_1, \ldots, m_\ell, s_1, \ldots, s_{\sigma_2}, v_1, \ldots, v_{\kappa_2}, 1)^T,$$

and E_i is a public $(\kappa_1 + \sigma_1 + 1) \times (\ell + \kappa_2 + \sigma_2 + 1)$ matrix over $\mathbb{Z}_p$.

Now fix a valid message-signature pair $(M_1, \ldots, M_\ell, R_1, \ldots, R_{\sigma_1}, S_1, \ldots, S_{\sigma_2})$, and suppose that there exists a non-zero tuple $(m_1^*, \ldots, m_\ell^*) \in \mathbb{Z}_p^\ell$ such that

$$E_i(m_1^*, \ldots, m_\ell^*, 0, \ldots, 0)^T = 0$$

for all i. Then, it is clear from the shape (19) of the corresponding verification equations that $(R_1, \ldots, R_{\sigma_1}, S_1, \ldots, S_{\sigma_2})$ is still a valid signature on the distinct message vector $(M_1 \tilde{G}^{m_1^*}, \ldots, M_\ell \tilde{G}^{m_\ell^*})$, which contradicts existential unforgeability.

Therefore, by denoting by n as the number of verification equations, the linear map $\mathbb{Z}_p^\ell \to \mathbb{Z}_p^{n(\kappa_1+\sigma_1+1)}$ mapping $(m_1, \ldots, m_\ell)$ to the concatenation of all vectors $E_i(m_1, \ldots, m_\ell, 0, \ldots, 0)^T$ must be injective. In particular, we have:

$$\ell \leq n \cdot (\kappa_1 + \sigma_1 + 1) \leq n \cdot (\kappa + \sigma)\ ,$$

where the second inequality comes from the fact that we must have $\sigma_2 \geq 1$; otherwise, the algebraic signing algorithm would output signatures that cannot depend on the message.

Finally, an argument similar to [7, Theorem 5] shows that we must have $n \leq \sigma$ (after removing possibly redundant verification equations). Indeed, if it were not the case, the quadratic system satisfied by the discrete logarithms of the signature elements would be overdetermined, and a generic message would not admit any valid signature at all. We thus obtain $\ell \leq \sigma \cdot (\kappa + \sigma) \leq (\kappa + \sigma)^2$, which concludes the proof. ∎

As a result, we immediately see that an FSPS scheme obtained from construction FSP1 must have signatures of more than $\sqrt{\ell}$ elements. This is because all signatures include as a subset including both the verification key and signature of a structure-preserving OTS scheme signing ℓ-element messages. Similarly, the following result holds with the same proof as above:

Theorem 9. *Consider a structure-preserving commitment scheme on messages in $\mathbb{G}_2^\ell$ in the asymmetric (Type III) bilinear group setting. Assume that the commitment key consists of elements in $\mathbb{G}_2$, and let χ be the number of elements in commitments and o the number of group elements in the opening information. If the scheme is collision resistant and has an algebraic commitment algorithm, we have $\chi + o \geq \sqrt{\ell}$.*

This shows that an FSPS scheme obtained from construction FSP2 must also have signatures of more than $\sqrt{\ell}$ elements, at least when the underlying trapdoor commitment scheme has its key elements on the same side as the resulting signature, which seems necessary with our approach based on $\mathsf{TC}\gamma$ (in particular, this holds for the instantiation above and all the variants in Section 6).

6 Variations

In our construction, it is POS or OTS that actually signs the input messages. We call the component a "front end." In our constructions, POS, OTS, and SPS are inter-changeable as a front end. (With care for details.) In FSP1, it is possible to replace OTS with SPS or POS. Similarly, POS used in FSP2 can be replaced by SPS (with empty ovk_{pos}) or OTS (with repetition $k = 1$). We take a concrete instantiation of SPS from [2] whose security relies only on static assumptions, and evaluate the efficiency of the variations in terms of the sizes of signatures and public-keys as described in Sections 6.1 to 6.4. The result is summarized in Table 3. We also measure the size of the Groth-Sahai proofs for ones knowledge of valid signaures. We omit the details and show the result in Table 4.

We conclude that FSP2 based on $\mathsf{TC}\gamma$ and POS currently has the most efficient instantiation in these aspects lited here. While SPS works to eliminate one-time keys of POS, it is not as efficient as expected since currently available instantiation of SPS under standard assumptions requires verification keys consisting of both source groups that forces succeeding building blocks to handle bilateral messages.

Scheme	Front End	$\lvert vk\rvert$	$\lvert\sigma\rvert$						# PPE
			ℓ	$\ell = 1$	4	9	25	100	
FSP1	OTS (§5.1)	$18 + 2\,\ell$	$14 + 3\,\ell$	17	26	41	89	314	$7 + 2\ell$
	SPS (§6.2)	$58 + 2\,\lceil\sqrt{\ell}\rceil$	$81 + 14\,\lceil\sqrt{\ell}\rceil$	95	109	123	151	221	$65 + 7\,\lceil\sqrt{\ell}\rceil$
	POS (§6.1)	$16 + 4\,\lceil\sqrt{\ell}\rceil$	$9 + 8\,\lceil\sqrt{\ell}\rceil$	17	25	33	49	89	$4 + 5\,\lceil\sqrt{\ell}\rceil$
FSP2	OTS (§6.3)	$22 + 3\,\ell$	$14 + \ell$	15	18	23	39	114	6
	SPS (§6.4)	$82 + 3\,\lceil\sqrt{\ell}\rceil$	$41 + 12\,\lceil\sqrt{\ell}\rceil$	53	65	77	101	161	$6 + 5\,\lceil\sqrt{\ell}\rceil$
	POS (§5.2)	$19 + 6\,\lceil\sqrt{\ell}\rceil$	$11 + 4\,\lceil\sqrt{\ell}\rceil$	15	19	23	31	51	$5 + \lceil\sqrt{\ell}\rceil$

Table 3: Size of secret keys, verification keys, signatures, and number of PPEs in verification for unilateral messages of size ℓ. The message sizes of POS and SPS that are used as building blocks are set to the optimal $\lceil\sqrt{\ell}\rceil$ assuming ℓ is a square of an integer. As samples, the signature sizes for $\ell \in \{1, 4, 9, 25, 100\}$ are compared.

Scheme	Front End	Proof Size	
		ℓ	$\ell = 1$
FSP1	OTS (§5.1)	$54 + 10\,\ell$	64
	SPS (§6.2)	$278 + 44\,\lceil\sqrt{\ell}\rceil$	322
	POS (§6.1)	$32 + 32\,\lceil\sqrt{\ell}\rceil$	64
FSP2	OTS (§6.3)	$58 + 2\,\ell$	60
	SPS (§6.4)	$122 + 34\,\lceil\sqrt{\ell}\rceil$	156
	POS (§5.2)	$44 + 16\,\lceil\sqrt{\ell}\rceil$	60

Table 4: Size of a Groth-Sahai witness indistinguishable proof for knowing a valid signature for a unilateral message of size ℓ. The message size for the front-end components is set to the optimal $\lceil\sqrt{\ell}\rceil$ assuming ℓ is a square of an integer.

6.1 xSIG + POS

Let $\mathcal{M}_{\mathsf{pos}}$ be the message space of POS. The following construction is for messages in $(\mathcal{M}_{\mathsf{pos}})^k$ for $k \geq 1$.

[**xSIG + POS**]

$\mathsf{SIG.Setup}(1^\lambda)$: Run $gk \leftarrow \mathsf{xSIG.Setup}()$ and output gk.

$\mathsf{SIG.Key}(gk)$: Run $(vk_{\mathsf{xsig}}, sk_{\mathsf{xsig}}) \leftarrow \mathsf{xSIG.Key}(gk)$ and output $vk := vk_{\mathsf{xsig}}$ and $sk := sk_{\mathsf{xsig}}$.

$\mathsf{SIG.Sign}(sk_{\mathsf{xsig}}, m)$: Parse $m := (m^{(1)}, \cdots, m^{(k)}) \in (\mathcal{M}_{\mathsf{pos}})^k$. Run $(vk_{\mathsf{pos}}, sk_{\mathsf{pos}}) \leftarrow \mathsf{POS.Key}(gk)$. Repeat the following for $j = 1, \cdots, k$.

1. $(ovk_{\mathsf{pos}}^{(j)}, osk_{\mathsf{pos}}^{(j)}) \leftarrow \mathsf{POS.Ovk}(gk)$
2. $\sigma_{\mathsf{pos}}^{(j)} := \mathsf{POS.Sign}(sk_{\mathsf{pos}}, osk_{\mathsf{pos}}^{(j)}, m^{(j)})$.

Then compute $\sigma_{\mathsf{xsig}} \leftarrow \mathsf{xSIG.Sign}(sk_{\mathsf{xsig}}, (vk_{\mathsf{pos}}, ovk_{\mathsf{pos}}^{(1)}, \cdots, ovk_{\mathsf{pos}}^{(k)}))$ and output $\sigma := (\sigma_{\mathsf{xsig}}, vk_{\mathsf{pos}}, ovk_{\mathsf{pos}}^{(1)}, \cdots, ovk_{\mathsf{pos}}^{(k)}, \sigma_{\mathsf{pos}}^{(1)}, \cdots, \sigma_{\mathsf{pos}}^{(k)})$

$\mathsf{SIG.Vrf}(vk, m, \sigma)$: Parse $m := (m^{(1)}, \cdots, m^{(k)}) \in (\mathcal{M}_{\mathsf{pos}})^k$. Output 1 if $1 = \mathsf{xSIG.Vrf}(vk_{\mathsf{xsig}}, (vk_{\mathsf{pos}}, ovk_{\mathsf{pos}}^{(1)}, \cdots, ovk_{\mathsf{pos}}^{(k)}), \sigma_{\mathsf{xsig}})$, and $1 = \mathsf{POS.Vrf}(vk_{\mathsf{pos}}, ovk_{\mathsf{pos}}^{(i)}, m^{(i)}, \sigma_{\mathsf{pos}}^{(i)})$ for $i = 1, \ldots, k$. Output 0, otherwise.

The total signature size is measured as follows. The vk_{pos} and ovk_{pos} should be messages of xSIG, and should be extended three times, i.e., $3\,(|vk_{\mathsf{pos}}| + \lceil \ell/\ell_{pos} \rceil\, |ovk_{\mathsf{pos}}|)$. Regarding the POS presented in Section 3.2, let ℓ_{pos} be the size of a message in $\mathcal{M}_{\mathsf{pos}}$. $|vk_{\mathsf{pos}}| = \ell_{\mathsf{pos}} + 1$. $|ovk_{\mathsf{pos}}| = 1$. $|\sigma_{\mathsf{pos}}^{(i)}| = 2$. $\#\,\mathsf{PPE}_{\mathsf{pos}} = 1$. The size of common parameter is $|gk| = 6 + 2\,(|vk_{\mathsf{pos}}| + \lceil \ell/\ell_{pos} \rceil \cdot |ovk_{\mathsf{pos}}|)$. The secret key of SIG is only that of xSIG and group elements. The size of the secret key is $|sk| = |sk_{\mathsf{xsig}}| = 4$. $\#\,\mathsf{PPE}_{\mathsf{xsig}} = 2 + 2\,(|vk_{\mathsf{pos}}| + \lceil \ell/\ell_{\mathsf{pos}} \rceil \cdot |ovk_{\mathsf{pos}}|)$. Thus we have:

- $|vk| = |vk_{\mathsf{xsig}}| = (6 + 2(|vk_{\mathsf{pos}}| + \lceil \ell/\ell_{\mathsf{pos}} \rceil \cdot |ovk_{\mathsf{pos}}|)) + 8 = 14 + 2(|vk_{\mathsf{pos}}| + \lceil \ell/\ell_{\mathsf{pos}} \rceil \cdot |ovk_{\mathsf{pos}}|) = 14 + 2\,(\ell_{\mathsf{pos}} + 1 + \lceil \ell/\ell_{\mathsf{pos}} \rceil) = 16 + 2\,\lceil \ell/\ell_{\mathsf{pos}} \rceil + 2\,\ell_{\mathsf{pos}} = 16 + 4\,\lceil \sqrt{\ell} \rceil$
- $|\sigma| = |\sigma_{\mathsf{xsig}}| + |\sigma_{\mathsf{pos}}| + 3\,(|vk_{\mathsf{pos}}| + \lceil \ell/\ell_{\mathsf{pos}} \rceil \cdot |ovk_{\mathsf{pos}}|) = 6 + \lceil \ell/\ell_{\mathsf{pos}} \rceil \cdot |\sigma_{\mathsf{pos}}| + 3\,|vk_{\mathsf{pos}}| + 3\,\lceil \ell/\ell_{\mathsf{pos}} \rceil \cdot |ovk_{\mathsf{pos}}| = 6 + 2\,\lceil \ell/\ell_{\mathsf{pos}} \rceil + 3\,(\ell_{\mathsf{pos}} + 1 + \lceil \ell/\ell_{\mathsf{pos}} \rceil) = 9 + 5\,\lceil \ell/\ell_{\mathsf{pos}} \rceil + 3\,\ell_{\mathsf{pos}} = 9 + 8\,\lceil \sqrt{\ell} \rceil$
- $\#\,\mathsf{PPE} = \#\,\mathsf{PPE}_{\mathsf{xsig}} + \#\,\mathsf{PPE}_{\mathsf{pos}} = \{2 + 2\,(|vk_{\mathsf{pos}}| + \lceil \ell/\ell_{\mathsf{pos}} \rceil \cdot |ovk_{\mathsf{pos}}|)\} + \lceil \ell/\ell_{\mathsf{pos}} \rceil = 2 + 2(\ell_{\mathsf{pos}} + 1) + 3\,\lceil \ell/\ell_{\mathsf{pos}} \rceil = 4 + 2\,\ell_{\mathsf{pos}} + 3\,\lceil \ell/\ell_{\mathsf{pos}} \rceil = 4 + 5\,\lceil \sqrt{\ell} \rceil$

The last equality in each evaluation is obtained by setting $\ell_{pos} = \lceil \ell/\ell_{\mathsf{pos}} \rceil = \lceil \sqrt{\ell} \rceil$, which is optimal for square ℓ.

6.2 xSIG + SPS

General construction follows from Section 6.1. Since the verification keys of SPS presented in [2] exist in both groups, we need to incorporate OTS to sign the verification keys of SPS as a bilateral message. OTS signs on a part of verification key of SPS and the verification key of OTS and the remaining part of verification key of SPS are given to xSIG as a unilateral message.

Let $M \in \mathbb{G}_1^\ell$ be a message to sign and ℓ_{sps} be the size of message for SPS. We have the following parameters for SPS from [2]: $|vk_{\mathsf{sps}}^{\mathbb{G}_1}| = 7$, and $|vk_{\mathsf{sps}}^{\mathbb{G}_2}| = 13 + \ell_{\mathsf{sps}}$. We then set $vk_{\mathsf{ots}} = |vk_{\mathsf{sps}}^{\mathbb{G}_1}| + 2 = 7 + 2 = 9$. Then the messages of xSIG is $|vk_{\mathsf{sps}}^{\mathbb{G}_2}| + |vk_{\mathsf{ots}}| = (13 + \ell_{\mathsf{sps}}) + 9 = 22 + \ell_{\mathsf{sps}}$. The signature size of each building block is $|\sigma_{\mathsf{sps}}| = 11$, $|\sigma_{\mathsf{ots}}| = 2$, and $|\sigma_{\mathsf{xsig}}| = 6$. $\#\,\mathsf{PPE}_{\mathsf{sps}} = 1 + 4 = 5$. This results in the following:

- $|vk| = |vk_{\mathsf{xsig}}| = 14 + 2(|vk_{\mathsf{sps}}^{\mathbb{G}_2}| + |vk_{\mathsf{ots}}|) = 14 + 2(22 + \ell_{\mathsf{sps}}) = 58 + 2\,\ell_{\mathsf{sps}} = 58 + 2\,\lceil \sqrt{\ell} \rceil$
- $|\sigma| = |\sigma_{\mathsf{xsig}}| + |\sigma_{\mathsf{ots}}| + |\sigma_{\mathsf{sps}}| + 3\,(|vk_{\mathsf{sps}}^{\mathbb{G}_2}| + |vk_{\mathsf{ots}}|) + |vk_{\mathsf{sps}}^{\mathbb{G}_1}| = |\sigma_{\mathsf{xsig}}| + |\sigma_{\mathsf{ots}}| + \lceil \ell/\ell_{\mathsf{pos}} \rceil \cdot |\sigma_{\mathsf{sps}}| + 3\,(|vk_{\mathsf{sps}}^{\mathbb{G}_2}| + |vk_{\mathsf{ots}}|) + |vk_{\mathsf{sps}}^{\mathbb{G}_1}| = 6 + 2 + 11\,\lceil \ell/\ell_{\mathsf{sps}} \rceil + 3(22 + \ell_{\mathsf{sps}}) + 7 = 81 + 11\,\lceil \ell/\ell_{\mathsf{pos}} \rceil + 3\,\ell_{\mathsf{sps}} = 81 + 14\,\lceil \sqrt{\ell} \rceil$
- $\#\,\mathsf{PPE} = \#\,\mathsf{PPE}_{\mathsf{xsig}} + \#\,\mathsf{PPE}_{\mathsf{ots}} + \lceil \ell/\ell_{\mathsf{sps}} \rceil \cdot \#\,\mathsf{PPE}_{\mathsf{sps}} = \{2 + 2\,(|vk_{\mathsf{sps}}^{\mathbb{G}_2}| + |vk_{\mathsf{ots}}|)\} + 1 + 5\,\lceil \ell/\ell_{\mathsf{sps}} \rceil = 2 + 2\,((22 + \ell_{\mathsf{sps}}) + 9) + 1 + 5\,\lceil \ell/\ell_{\mathsf{sps}} \rceil = 65 + 5\,\lceil \ell/\ell_{\mathsf{pos}} \rceil + 2\,\ell_{\mathsf{sps}} = 65 + 7\,\lceil \sqrt{\ell} \rceil$

6.3 xSIG + TCγ + OTS

The POS in Section 3.2 is used as OTS by setting the number of iterations to 1 (i.e., one-time) and both of vk_{pos} and ovk_{pos} as vk_{ots}. We thus have the following parameters. $|vk_{\mathsf{ots}}| = \ell + 2$. $|\sigma_{\mathsf{ots}}| = 2$. $\#\,\mathsf{PPE}_{\mathsf{ots}} = 1$. According to the description in Section 4.2, $|ck_{\mathsf{gbc}}| = |vk_{\mathsf{ots}}| = 2 + \ell$, $|com_{\mathsf{gbc}}| = 1$, and $|open_{\mathsf{gbc}}| = 1$. $\#\,\mathsf{PPE}_{\mathsf{gbc}} = 1$ and $\#\,\mathsf{PPE}_{\mathsf{xsig}} = 2 + 2\,|com_{\mathsf{gbc}}| = 4$. $|gk| = 6 + 2\,|com_{\mathsf{gbc}}|$. Thus we have:

- $|vk| = |gk| + |vk_{\mathsf{xsig}}| + 3\,|ck_{\mathsf{gbc}}| = (6 + 2|com_{\mathsf{gbc}}|) + 8 + 3\,|ck_{\mathsf{gbc}}| = 14 + 2|com_{\mathsf{gbc}}| + 3\,|ck_{\mathsf{gbc}}| = 14 + 2 \cdot 1 + 3\,(2 + \ell) = 22 + 3\,\ell$
- $|\sigma| = |\sigma_{\mathsf{xsig}}| + |open_{\mathsf{gbc}}| + |\sigma_{\mathsf{ots}}| + |vk_{\mathsf{ots}}| + 3|com_{\mathsf{gbc}}| = 6 + 1 + 2 + (2 + \ell) + 3 \cdot 1 = 14 + \ell$
- $\#\,\mathsf{PPE} = \#\,\mathsf{PPE}_{\mathsf{xsig}} + \#\,\mathsf{PPE}_{\mathsf{gbc}} + \#\,\mathsf{PPE}_{\mathsf{ots}} = (2 + 2\,|com_{\mathsf{gbc}}|) + 1 + 1 = 4 + 2 \cdot 1 = 6$

6.4 xSIG + TCγ + SPS

The same caution from Section 6.2 applies to incorporate SPS. Since the verification keys of SPS exist in both groups, OTS is involved to sign the verification keys of SPS in one-side group. The verification keys of SPS in the other group and the verification keys of OTS will be given to TCγ as input.

Let $M \in \mathbb{G}_1^{\ell}$, and ℓ_{sps} be the size of a message for SPS. $|vk_{\mathsf{sps}}^{\mathbb{G}_1}| = 7$. $|vk_{\mathsf{sps}}^{\mathbb{G}_2}| = 13 + \ell_{\mathsf{sps}}$. $vk_{\mathsf{ots}} = |vk_{\mathsf{sps}}^{\mathbb{G}_1}| + 2 = 7 + 2 = 9$. $\#\,\mathsf{PPE}_{\mathsf{ots}} = 1$. Following for Section 4.2, $|open_{\mathsf{gbc}}| = 1$. $|com_{\mathsf{gbc}}| = 1$ and $|ck_{\mathsf{gbc}}| = |vk_{\mathsf{sps}}^{\mathbb{G}_2}| + |vk_{\mathsf{ots}}| = (13 + \ell_{\mathsf{sps}}) + 9 = 22 + \ell_{\mathsf{sps}}$. The size of $|ck_{\mathsf{gbc}}|$ should be extended to $3\,|ck_{\mathsf{gbc}}|$, as discussed in Section 3.1. $|\sigma_{\mathsf{sps}}| = 11$. $|\sigma_{\mathsf{ots}}| = 2$. $|\sigma_{\mathsf{xsig}}| = 6$. $\#\,\mathsf{PPE}_{\mathsf{sps}} = 1 + 4 = 5$. $\#\,\mathsf{PPE}_{\mathsf{xsig}} = 2 + 2\,|com_{\mathsf{gbc}}| = 2 + 2 \cdot 1 = 4$. $|gk| = 6 + 2\,|com_{\mathsf{gbc}}|$. This results in:

- $|vk| = |gk| + |vk_{\mathsf{xsig}}| + |ck_{\mathsf{gbc}}| = (6 + 2|com_{\mathsf{gbc}}|) + |vk_{\mathsf{xsig}}| + |ck_{\mathsf{gbc}}| = (6 + 2 \cdot 1) + 8 + 3\,(22 + \ell_{\mathsf{sps}}) = 82 + 3\,\ell_{\mathsf{sps}} = 82 + 3\,\lceil\sqrt{\ell}\rceil$
- $|\sigma| = |\sigma_{\mathsf{xsig}}| + |open_{\mathsf{gbc}}| + |\sigma_{\mathsf{ots}}| + |\sigma_{\mathsf{sps}}| + 3\,|com_{\mathsf{gbc}}| + |vk_{\mathsf{ots}}| + |vk_{\mathsf{sps}}| = |\sigma_{\mathsf{xsig}}| + |open_{\mathsf{gbc}}| + |\sigma_{\mathsf{ots}}| + \lceil\ell/\ell_{\mathsf{sps}}\rceil \cdot |\sigma_{\mathsf{sps}}| + 3|com_{\mathsf{gbc}}| + |vk_{\mathsf{ots}}| + |vk_{\mathsf{sps}}^{\mathbb{G}_1}| = 6 + 1 + 2 + 11\,\lceil\ell/\ell_{\mathsf{sps}}\rceil + 3 \cdot 1 + (22 + \ell_{\mathsf{sps}}) + 7 = 41 + 11\,\lceil\ell/\ell_{\mathsf{sps}}\rceil + \ell_{\mathsf{sps}} = 41 + 12\,\lceil\sqrt{\ell}\rceil$
- $\#\,\mathsf{PPE} = \#\,\mathsf{PPE}_{\mathsf{xsig}} + \#\,\mathsf{PPE}_{\mathsf{gbc}} + \#\,\mathsf{PPE}_{\mathsf{ots}} + \lceil\ell/\ell_{\mathsf{sps}}\rceil \cdot \#\,\mathsf{PPE}_{\mathsf{sps}} = 4 + 1 + 1 + 5\,\lceil\ell/\ell_{\mathsf{sps}}\rceil = 6 + 5\,\lceil\ell/\ell_{\mathsf{sps}}\rceil = 6 + 5\,\lceil\sqrt{\ell}\rceil$

7 Conclusion

In this paper we introduced the notion of fully structure-preserving signatures and present its instantiations based on static assumptions over Type-III bilinear groups. We discuss a trade-off between the size of signatures and verification keys for general structure-preserving signature schemes under generic constructions and show that $\sqrt{\ell}$ factor appear in our concrete construction is the optimal balancing point. We also presented a structure-preserving shrinking commitment scheme satisfying a relaxed binding property.

Open issues:

- Reducing the message expansion factor in xSIG is certainly desirable. Although it would only have a limited impact on the efficiency of our main construction FSP2, it would make the conceptually simpler scheme FSP1 more practical.
- While this work focused on standard static assumptions, it be worth investigating efficiency improvement under stronger assumptions.

References

[1] M. Abe, J. Camenisch, R. Dowsley, and M. Dubovitskaya. On the impossibility of structure-preserving deterministic primitives. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, pages 713–738, 2014.

[2] M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In *Advances in Cryptology - ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 4–24. Springer, 2012.

[3] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. *Journal of Cryptology*, 2014. DOI: 10.1007/s00145-014-9196-7.

[4] M. Abe, R. Gennaro, and K. Kurosawa. Tag-KEM/DEM: A new framework for hybrid encryption. *Journal of Cryptology*, 21(1):97–130, 2008.

[5] M. Abe, J. Groth, K. Haralambiev, and M. Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. In *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *LNCS*, pages 649–666. Springer, 2011.

[6] M. Abe, J. Groth, and M. Ohkubo. Separating short structure-preserving signatures from non-interactive assumptions. In *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 628–646. Springer, 2011.

[7] M. Abe, J. Groth, M. Ohkubo, and M. Tibouchi. Structure-preserving signatures from type II pairings. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 390–407. Springer, 2014.

[8] M. Abe, J. Groth, M. Ohkubo, and M. Tibouchi. Unified, minimal and selectively randomizable structure-preserving signatures. In *Theory of Cryptography - 11th Theory of Cryptography Conference*, volume 8349 of *LNCS*, pages 688–712. Springer, 2014.

[9] M. Abe, K. Haralambiev, and M. Ohkubo. Group to group commitments do not shrink. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 301–317. Springer, 2012.

[10] M. Abe, M. Kohlweiss, M. Ohkubo, and M. Tibouchi. Fully structure-preserving signatures and shrinking commitments. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, Lecture Notes in Computer Science. Springer, 2015. To appear.

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

[12] G. Barthe, E. Fagerholm, D. Fiore, A. Scedrov, B. Schmidt, and M. Tibouchi. Strongly-optimal structure preserving signatures from type II pairings: synthesis and lower bounds. In J. Katz, editor, *PKC 2015*, Lecture Notes in Computer Science. Springer, 2015. To appear.

[13] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In S. Halevi, editor, *Advances in Cryptology - CRYPTO*, volume 5677 of *LNCS*, pages 108–125. Springer, 2009.

[14] M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. P-signatures and noninteractive anonymous credentials. In R. Canetti, editor, *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 356–374. Springer, 2008.

[15] M. Bellare and A. Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In M. K. Franklin, editor, *CRYPTO*, volume 3152 of *LNCS*, pages 273–289. Springer, 2004.

[16] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings*, pages 136–153, 2005.

[17] M. Bellare and S. Shoup. Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. In *Public-Key Cryptography*, volume 4450 of *LNCS*, pages 201–216, 2007.

[18] A. Bender, J. Katz, and R. Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. *J. Cryptology*, 22(1):114–138, 2009.

[19] J. Camenisch, K. Haralambiev, M. Kohlweiss, J. Lapon, and V. Naessens. Structure preserving CCA secure encryption and applications. In D. H. Lee and X. Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 89–106. Springer, 2011.

[20] J. Camenisch, S. Krenn, and V. Shoup. A framework for practical universally composable zero-knowledge protocols. In *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, pages 449–467, 2011.

[21] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, pages 93–118, 2001.

[22] D. Catalano, M. D. Raimondo, D. Fiore, and R. Gennaro. Off-line/on-line signatures: Theoretical aspects and experimental results. In *Public Key Cryptography - PKC 2008, 11th International Workshop on Practice and Theory in Public-Key Cryptography, Barcelona, Spain, March 9-12, 2008. Proceedings*, volume 4939 of *LNCS*, pages 101–120. Springer, 2008.

[23] J. Cathalo, B. Libert, and M. Yung. Group encryption: Non-interactive realization in the standard model. In M. Matsui, editor, *ASIACRYPT*, volume 5912 of *LNCS*, pages 179–196, 2009.

[24] M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn. Malleable signatures: New definitions and delegatable anonymous credentials. In *2013 IEEE 27th Computer Security Foundations Symposium*, 2014.

[25] S. Chatterjee and A. Menezes. Type 2 structure-preserving signature schemes revisited. IACR ePrint Archive, Report 2014/635, 2014. `http://eprint.iacr.org`.

[26] M. Dubovitskaya. Cryptographic Protocols for Privacy-Preserving Access Control in Databases, 2014.

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

[28] M. Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In V. Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 152–168. Springer, 2005.

[29] G. Fuchsbauer. Commuting signatures and verifiable encryption. In *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, pages 224–245, 2011.

[30] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[31] 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.

[32] J. Groth. Fully anonymous group signatures without random oracles. In *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, pages 164–180, 2007.

[33] J. Groth and A. Sahai. Efficient noninteractive proof systems for bilinear groups. *SIAM J. Comput.*, 41(5):1193–1232, 2012.

[34] S. Hada and T. Tanaka. On the existence of 3-round zero-knowledge protocols. In H. Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, volume 1462 of *LNCS*, pages 354–369. Springer-Verlag, 1998. Full version available from IACR e-print archive 1999/009.

[35] A. Kate, G. M. Zaverucha, and I. Goldberg. Constant-size commitments to polynomials and their applications. In M. Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *LNCS*, pages 177–194. Springer, 2010.

[36] B. Libert, T. Peters, M. Joye, and M. Yung. Linearly homomorphic structure-preserving signatures and their applications. In R. Canetti and J. Garay, editors, *Advances in Cryptology - CRYPTO*, LNCS. Springer, 2013.

[37] S. Meiklejohn. An extension of the Groth-Sahai proof system. In *Brown University Masters thesis*, 2009.

[38] S. Micali, K. Ohta, and L. Reyzin. Accountable-subgroup multisignatures: extended abstract. In *CCS 2001, Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, Pennsylvania, USA, November 6-8, 2001.*, pages 245–254, 2001.

[39] P. Mohassel. One-time signatures and chameleon hash functions. In A. Biryukov, G. Gong, and D. R. Stinson, editors, *Selected Areas in Cryptography — SAC 2010*, volume 6544 of *LNCS*, pages 302–319. Springer, 2011.

[40] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In J. Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *LNCS*, pages 129–140. Springer, 1992.

[41] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, pages 552–565, 2001.