Compact Structure-preserving Signatures with Almost Tight Security

Masayuki Abe[1], Dennis Hofheinz*[2], Ryo Nishimaki[1], Miyako Ohkubo[3], and Jiaxin Pan**[2]

[1] Secure Platform Laboratories, NTT Corporation, Japan
{abe.masayuki, nishimaki.ryo}@lab.ntt.co.jp
[2] Karlsruhe Institute of Technology, Germany
{dennis.hofheinz, jiaxin.pan}@kit.edu
[3] Security Fundamentals Laboratory, CSR, NICT, Japan
m.ohkubo@nict.go.jp

Abstract. In structure-preserving cryptography, every building block shares the same bilinear groups. These groups must be generated for a specific, a priori fixed security level, and thus it is vital that the security reduction of all involved building blocks is as tight as possible. In this work, we present the first generic construction of structure-preserving signature schemes whose reduction cost is independent of the number of signing queries. Its chosen-message security is almost tightly reduced to the chosen-plaintext security of a structure-preserving public-key encryption scheme and the security of Groth-Sahai proof system. Technically, we adapt the adaptive partitioning technique by Hofheinz (Eurocrypt 2017) to the setting of structure-preserving signature schemes. To achieve a structure-preserving scheme, our new variant of the adaptive partitioning technique relies only on generic group operations in the scheme itself. Interestingly, however, we will use non-generic operations during our security analysis. Instantiated over asymmetric bilinear groups, the security of our concrete scheme is reduced to the external Diffie-Hellman assumption with linear reduction cost in the security parameter, independently of the number of signing queries. The signatures in our schemes consist of a larger number of group elements than those in other non-tight schemes, but can be verified faster, assuming their security reduction loss is compensated by increasing the security parameter to the next standard level.

Keywords. Structure-preserving signatures, Tight reduction, Adaptive partitioning

1 Introduction

Background. A structure-preserving signature (SPS) scheme [3] is designed over bilinear groups, and features public keys, messages, and signatures that only consist of source group elements. Furthermore, signature verification only uses group membership testing and relations that can be expressed as pairing product equations. Coupled with the Groth-Sahai non-interactive proof system [31] (GS proofs for short), SPS schemes are a powerful tool in constructing a wide range of cryptographic applications. Various SPS schemes based on compact standard

* Supported by DFG grants HO 4534/4-1 and HO 4534/2-2.
** Supported by the DFG grant HO 4534/4-1.

assumptions exist in the literature [30,19,3,20,17,4,2,41,37,35]. When looking at schemes from standard assumptions the state-of-the-art scheme in [35] yields signatures as compact as consisting of six source group elements.

In this paper, we address the tightness of security proofs for SPS schemes with compact parameters, i.e., constant-size signatures and standard (non q-type) assumptions. Formally, a security reduction constructs an adversary $\mathcal{A}$ on a computational assumption out of an adversary $\mathcal{A}'$ on the security of a cryptographic scheme. If we let ϵ and t denote the success probability and runtime of $\mathcal{A}$, and ϵ' and t' the success probability and runtime of $\mathcal{A}'$, then we define the security loss of the reduction, or simply the reduction cost, as $(\epsilon' t)/(\epsilon t')$ [21]. The reduction is tight if the security loss is a small constant or almost tight if it grows only (as a preferably small function) in the security parameter λ. In particular, we are concerned with the independence of the security loss from the number q_s of $\mathcal{A}'$'s signing queries in a chosen-message attack. We note that in practice, q_s can be as large as 2^{30} while λ is typically 128.

The only tightly secure SPS under compact assumptions is that by Hofheinz and Jager [34]. Their tree-based construction, however, yields unacceptably large signatures consisting of hundreds of group elements. For other SPS schemes under compact assumptions the security is proven using a hybrid argument that repeat reductions in q_s. Thus, their security loss is $\mathcal{O}(q_s)$ [2,41] or even $\mathcal{O}(q_s^2)$ [37], as shown in Table 1.

Reference	$\lvert\mathsf{M}\rvert$	$\lvert\sigma\rvert$	$\lvert pk\rvert$	Sec. Loss	Assumptions
HJ [34]	1	$10d+6$	13	8	DLIN
ACDKNO [2]	$(n_1, 0)$	$(7, 4)$	$(5, n_1+12)$	$\mathcal{O}(q_s)$	SXDH, XDLIN_1
ACDKNO [2]	(n_1, n_2)	$(8, 6)$	(n_2+6, n_1+13)	$\mathcal{O}(q_s)$	SXDH, XDLIN_1
LPY [41]	$(n_1, 0)$	$(10, 1)$	$(16, 2n_1+5)$	$\mathcal{O}(q_s)$	SXDH, XDLIN_2
KPW [37]	$(n_1, 0)$	$(6, 1)$	$(0, n_1+6)$	$\mathcal{O}(q_s^2)$	SXDH
KPW [37]	(n_1, n_2)	$(7, 3)$	(n_2+1, n_1+7)	$\mathcal{O}(q_s^2)$	SXDH
JR [35]	$(n_1, 0)$	$(5, 1)$	$(0, n_1+6)$	$\mathcal{O}(q_s \log q_s)$	SXDH
Ours (Sect. 4.2)	$(n_1, 0)$	$(13, 12)$	$(18, n_1+11)$	$\mathcal{O}(\lambda)$	SXDH
Ours (Sect. 4.3)	(n_1, n_2)	$(14, 14)$	(n_2+19, n_1+12)	$\mathcal{O}(\lambda)$	SXDH

Table 1: Object sizes and loss of security among structure-preserving signature schemes with assumptions in the standard model. Smallest possible parameters are set to parameterized assumptions. Notation (x, y) means x and y elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. The $\lvert\mathsf{M}\rvert$, $\lvert\sigma\rvert$, $\lvert pk\rvert$ columns mean the number of messages, the number of group elements in a signature, and the number of group elements in a public key, respectively. The "Sec. Loss" column means reduction costs. The "Assumptions" column means the underlying assumptions for proving security. For HJ, parameter d limits number of signing to 2^d. Parameters q_s and λ represent number of signing queries and security parameter, respectively.

The non-tightness of security reductions does not necessarily mean the existence of a forger with reduced complexity, but the security guarantees given

Reference	$\|\mathsf{M}\|$	#(s.mult) in signing	#(PPEs)	#(Pairings)	
				Plain	Batched
KPW [37]	$(n_1, 0)$	$(6, 1)$	3	$n_1 + 11$	$n_1 + 10$
JR [35]		$(6, 1)$	2	$n_1 + 8$	$n_1 + 6$
Ours (Sect. 4.2)		$(15, 15)$	15	$n_1 + 57$	$n_1 + 16$
KPW [37]	(n_1, n_2)	$(8, 3.5)$	4	$n_1 + n_2 + 15$	$n_1 + n_2 + 14$
Ours (Sect. 4.3)		$(17.5, 16)$	16	$n_1 + n_2 + 61$	$n_1 + n_2 + 18$

Table 2: Comparison of factors relevant to computational efficiency against SPS schemes having smallest signature sizes. Third column indicates number of scalar multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$ for signing. Multi-scalar multiplication is counted as 1.5. For JR, a constant pairing is included. Column "Batched" shows the number of pairings in a verification when pairing product equations are merged into one by using a batch verification technique [13].

by non-tight reductions are quantitatively weaker than those given by tight reductions. Recovering from the security loss by increasing the security parameter is not a trivial solution when bilinear groups are involved. The security in source and target groups should be balanced, and computational efficiency is influenced by the choice of curves, pairings, and parameters such as embedding degrees, and the presence of dedicated techniques. In practice, an optimal setting for a targeted security parameter is determined by actual benchmarks, e.g., [28,6,25], and only standard security parameters such as 128, 192, and 256, have been investigated. One would thus have to hop to the next standard security level to offset the security loss in reality. Besides, we stress that increasing the security parameter for a building block in structure-preserving cryptography is more costly than usual as it results in losing efficiency in all other building blocks using the same bilinear groups. Thus, the demand for tight security is stronger in structure-preserving cryptography.

Even in ordinary (i.e. non-structure-preserving) signature schemes, most of the constructions satisfying tight security are either in the random oracle model, e.g. [11,36,23,1], rely on q-type or strong RSA assumptions, e.g., [15,42], or lead to large signatures and/or keys, e.g., [22,39]. Hofheinz presented the first tightly secure construction with compact signatures and keys under a standard compact assumption over bilinear groups [32]. However, his construction can only be used to sign integer messages (and not group elements or, e.g., its own public key), so it is not structure-preserving.

Our Contributions. We propose the *first (almost) tightly secure* SPS schemes with *constant* number of group elements in signatures. Our schemes are proven secure based on standard assumptions (e.g., the symmetric external Diffie-Hellman (SXDH) assumption). Concretely, we first present a generic construction of an almost tightly secure SPS scheme from a structure-preserving public-key encryption secure against chosen-plaintext attacks and the GS proof system.

With ElGamal encryption and the GS proofs over asymmetric pairing groups, we obtain concrete SPS schemes with compact signature size whose unforgeability against adaptive chosen-message attacks (UF-CMA) is reduced from the SXDH assumption with security loss of $\mathcal{O}(\lambda)$, which is independent of q_s.

The primary benefit of our tightly secure SPS schemes is their availability in structure-preserving cryptography under the current standard security level. For a system modularly built with structure-preserving building blocks, a compact and tightly secure SPS scheme has been a missing piece, since other useful building blocks, such as one-time signatures and commitments, are known to be tightly secure. Plugging in our scheme, one can increase the proven security in applications of structure-preserving cryptography such as blind signatures [3], group signatures [41], and unlinkable redactable signatures [18] used in anonymous credential systems.

The second benefit of our result is the removal of q_s from the security bound, which aims to simplify the systems design. With previous schemes, there are trade-offs among security, efficiency, and usability; if one desires stronger security guarantees without sacrificing efficiency, a rigid limitation has to be put on the number of signatures per public key, or, if more flexibility on the number of possible signatures is important in considered applications, one has to take the risk with weaker security guarantees or less efficiency. With our schemes, one no longer needs to fix q_s in advance and can focus on desirable security and permissible efficiency for the targeted system.

Nevertheless, the performance as a stand-alone signature scheme is of a concern. We summarise several parameters that dominate the space and computation costs in Tables 1 and 2. The bare numbers in the tables imply that our schemes are outperformed by those in the literature if they are used at the *same* security level. Taking the security loss into consideration, however, the tightness of our schemes offsets the difference in terms of computational complexity. We elaborate this point in the following. Though concrete complexity varies widely depending on platforms and implementations, it is safe to say that computing a pairing in the 192-bit security level is slowed by a factor of $\delta := 6$ to 7 on ordinary personal computers [8,25] and $\delta := 9$ to 12 on processors for embedded systems [5,29,44] compared to those in the 128-bit security level. According to the number of pairings in Table 2, our scheme for bilateral messages at the 128-bit security level verifies a signature with batch verification $4.6 < \delta(n_1+n_2+14)/(n_1+n_2+18) < 9.3$ times faster than the KPW scheme at the 192-bit security setting for offsetting its security loss of 60 bits. Applying the same argument to the case of unilateral messages, ours in the 128-bit security level will be $2.2 < \delta(n_1+6)/(n_1+16) < 4.5$ times faster compared to the JR scheme in the 192-bit security level. Even with plain verification, i.e., without batch technique, the advantage remains depending on the platform and the size of messages.

We note that the above simple argument ignores dedicated techniques for computing pairing products, e.g., [43], and costs for subtle computations. It may not be fair to ignore the concrete security loss in our schemes, which can be as large as 13 bits at the 128-bit security level, as mentioned in Section 4.

Nevertheless, taking into account the fact that the performance gap between different security levels will be larger than those shown in the above benchmarks published previously [38] (i.e., slowdown factor δ in the above argument will be much larger), even the simple estimation is aimed to show the practical significance of tightly secure schemes.

Technical Overview. Eliminating any representation-dependent computation in the construction is a crucial technical challenge. Towards this goal, we adapt the "adaptive partitioning" technique of Hofheinz [33] (which in turn builds upon [22]) to the setting of structure-preserving signatures. Thus, in our security proof, we gradually transform the conditions necessary for a successful forgery until a valid forgery is impossible. This will require $\mathcal{O}(\lambda)$ game hops, thus leading to a security loss independent of the number of adversarial signing queries.

Concretely, in the scheme itself, we require that every valid signature must carry an (encrypted) "authentication tag" $Z = X$, where $X \in \mathbb{G}$ is a fixed group element. We will gradually transform this requirement $Z = X$ into the following combination of requirements on the authentication tag Z^* from a valid forgery:

(a) We must have $Z^* = X \cdot \mathsf{M}^*$, where $X \in \mathbb{G}$ is a fixed random group element, and $\mathsf{M}^* \in \mathbb{G}$ is the signed message in the forgery.
(b) Also, we must have $Z^* = X \cdot \mathsf{M}_i$ for some previously signed message M_i.

Since we may assume $\mathsf{M}^* \notin \{\mathsf{M}_i\}$ in the (non-strong) existential unforgeability experiment, any attempted forgery will thus be invalid.

The key technique to establishing these modified requirements is a "partitioning argument" similar to the one from [33]. That is, in the proof, we will enforce more and more dependencies of the authentication tag Z on the *bit representation* of M. Note that this bit representation is not used in the real scheme; this would in fact be problematic in the context of structure-preserving constructions. For instance, to establish a dependence of Z on the k-th bit b_{M} of the bit representation of M, we proceed as follows:

1. First, we "partition" the set of all messages into two subsets, depending on b_{M}. This means that signatures issued by the experiment now carry (an encryption of) b_{M} in a special component. The reason for this partitioning is that we can now, depending on the encrypted b_{M}, use different verification rules.
2. We guess the encrypted bit b^* from the forgery, and change the encrypted Z in issued signatures for all $b_{\mathsf{M}} \neq b^*$. (This change can be justified by setting up things such that Z can only be retrieved from a signature if the encrypted bit b is equal to b^*. If $b \neq b^*$, then Z is hidden, and can hence be modified in issued signatures.) This introduces a dependence of Z in issued signatures on b_{M}.

However, the encrypted bit b^* from the forgery is not necessarily identical to b_{M^*} (since this property cannot be easily enforced in a structure-preserving way). As a consequence, we cannot force the adversary to respect the additional dependencies in his forgery. Yet, we will show that we *can* force the adversary to *reuse* one $Z = X \cdot \mathsf{M}_i$ from a signing query. This leads to requirement (b) in verification

forgeries, and requirement (a) will finally be enforced by a regular GS proof in signatures (that GS proof is simulated in all intermediate steps).

This line of reasoning borrows from Chen and Wee's [22] general idea of establishing tight security through a repeated partitioning of the message space (resp. identity space in an identity-based encryption scheme) into two sets, each time adjusting signatures for messages from one of the two sets in the process. However, their approach, as well as other follow-up approaches (e.g., [14,40,7,32,27]) embeds the partitioning already in the scheme (in the sense that the scheme must already contain all potentially possible "partitioning rules," for instance according to each message bit). Since these rules in the mentioned schemes are based on the message bits (or an algebraic predicate on the discrete logarithm of the message [32]), this would not lead to a structure-preserving scheme.

Instead, we adapt the "adaptive partitioning" (AP) technique of Hofheinz [33], in which the partitioning is performed dynamically, through an *encrypted* partitioning bit embedded in signatures. This allows us to separate partitioning from the way messages are bound to signatures in the scheme. We thus bind a message through an authentication tag, as mentioned above, that is more algebraic and admits structure-preserving GS proofs. The encrypted partitioning bit is fixed to a constant in the real scheme and turned into a variable only in the security proof where non-generic computations are allowed.

In adapting AP to our setting, we face two difficulties, however: the partitioning used in AP is bit-based (which is incompatible with our requirement of a structure-preserving scheme), and its complexity leads to comparatively complex schemes. More specifically, AP leads to several expensive "OR"-proofs in ciphertexts, resp. signatures. As a consequence, the (encryption) schemes in [33] are not competitive in complexity to non-tightly secure schemes, even when taking into account a potentially larger security level for non-tightly secure schemes. On the other hand, our signature schemes are carefully designed so that GS proofs in signatures are done only for less costly linear relations (except for one crucial "OR"-proof). We further use optimization techniques of Escala and Groth [26] to reduce the size of GS proofs in our instantiation.

Moreover, AP crucially relies on the bit representation of messages (resp. encryption tags that are hash values in [33]). In particular, the encryption scheme from [33] is not structure-preserving. For our purposes, we thus have to modify this technique to work with group elements instead of hash values. This leads to a very simple and clean structure-preserving signature scheme whose security proof still crucially uses the bit representation of group elements. We find this property surprising and conceptually interesting.

Open Problems. While being compact and tightly secure, our concrete SPS schemes contain a moderate number of group elements in a signature. We leave as an open problem to design more compact SPSes with even smaller number of group elements. Another interesting open problem is to decrease the security loss from $\mathcal{O}(\lambda)$ to $\mathcal{O}(1)$.

ORGANIZATION. The rest of the paper is organized as follows. After introducing notations, security definitions, and building blocks in Section 2, we present our generic construction and its security proof in Section 3. We discuss an instantiation over asymmetric bilinear groups in Section 4.

2 Preliminaries

2.1 Notations

For an integer p, define $\mathbb{Z}_p$ as the residual ring $\mathbb{Z}/p\mathbb{Z}$. If $\mathcal{B}$ is a set, then $x \xleftarrow{\$} \mathcal{B}$ denotes the process of sampling an element x from set $\mathcal{B}$ uniformly at random. All our algorithms are probabilistic polynomial time (p.p.t. for short) unless stated otherwise. If $\mathcal{A}$ is an algorithm, then $a \xleftarrow{\$} \mathcal{A}(b)$ denotes the random variable, which is defined as the output of $\mathcal{A}$ on input b. To make the randomness explicit, we use the notation $a \leftarrow \mathcal{A}(b; r)$ meaning that the algorithm is executed on input b and randomness r. Note that $\mathcal{A}$'s execution is now deterministic.

We say that a function ϵ is negligible in security parameter λ if, for all constant $c > 0$ and all sufficiently large λ, $\nu(\lambda) < \lambda^{-c}$ holds.

2.2 Pairing Groups and Diffie-Hellman Assumptions

Let PGGen be an algorithm that on input security parameter λ returns a description $\mathsf{par} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ of pairing groups, where p is a $\mathsf{poly}(\lambda)$-bit prime, $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ are cyclic groups of order p, G_1 and G_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable non-degenerate bilinear map. Pairing group par is said to be a Type-III asymmetric pairing group if $\mathbb{G}_1 \neq \mathbb{G}_2$, and there does not exist an efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. When distinction between source groups is not important, we use $\mathbb{G}$ and G to represent $\mathbb{G}_1$ and/or $\mathbb{G}_2$, and their default generator, respectively. When a group element is given to an algorithm as an input, its membership to the intended group must be tested, but we make it implicit throughout the paper for conciseness of the description.

Our instantiation in Section 4 is based on the following standard assumption over asymmetric pairing groups.

Definition 2.1 (Decisional Diffie-Hellman assumption). The decisional Diffie-Hellman assumption (DDH_s) *holds relative to* PGGen *in group* $\mathbb{G}_s$ *($s \in \{1, 2, T\}$) if, for all p.p.t. adversaries $\mathcal{A}$, advantage function*

$$\mathrm{Adv}^{\mathsf{ddh}_s}_{\mathsf{PGGen}}(\mathcal{A}) := |\Pr[\mathcal{A}(\mathsf{par}, G_s^a, G_s^b, G_s^{ab}) = 1] - \Pr[\mathcal{A}(\mathsf{par}, G_s^a, G_s^b, G_s^c) = 1]|$$

is negligible in security parameter λ, where the probability is taken over $\mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda), a, b, c \xleftarrow{\$} \mathbb{Z}_p$. The SXDH assumption *holds relative to* PGGen *if for all p.p.t. adversaries $\mathcal{A}$, advantage function* $\mathrm{Adv}^{\mathsf{sxdh}}_{\mathsf{PGGen}}(\mathcal{A}) := \max(\mathrm{Adv}^{\mathsf{ddh}_1}_{\mathsf{PGGen}}(\mathcal{A}), \mathrm{Adv}^{\mathsf{ddh}_2}_{\mathsf{PGGen}}(\mathcal{A}))$ *is negligible.*

2.3 Structure-preserving Signatures

Definition 2.2 (Structure-preserving signature scheme). *An SPS scheme* SPS *with respect to* PGGen *is a tuple of algorithms* $\mathsf{SPS} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$*:*

- *The key generation algorithm* $\mathsf{Gen}(\mathsf{par})$ *takes* $\mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda)$ *as input and returns a public/secret key pair,* (pk, sk)*, where* $pk \in \mathbb{G}^{n_{pk}}$ *for some* $n_{pk} \in \mathsf{poly}(\lambda)$*. Message space* $\mathcal{M} := \mathbb{G}^n$ *for some constant* $n \in \mathsf{poly}(\lambda)$ *is implicitly determined by* pk*.*
- *The signing algorithm* $\mathsf{Sign}(sk, \mathsf{M})$ *returns a signature* $\sigma \in \mathbb{G}^{n_\sigma}$ *for some* $n_\sigma \in \mathsf{poly}(\lambda)$*.*
- *The deterministic verification algorithm* $\mathsf{Ver}(pk, \mathsf{M}, \sigma)$ *solely evaluates pairing product equations and returns 1 (accept) or 0 (reject).*

(Perfect correctness.) *For all* $(pk, sk) \xleftarrow{\$} \mathsf{Gen}(\mathsf{par})$*, all messages* $\mathsf{M} \in \mathcal{M}$*, and all* $\sigma \xleftarrow{\$} \mathsf{Sign}(sk, \mathsf{M})$*,* $\mathsf{Ver}(pk, \mathsf{M}, \sigma) = 1$ *holds.*

Though our final goal is to achieve security against adaptive chosen-message attacks, we use the following slightly relaxed notion in the generic construction.

Definition 2.3 (UF-XCMA Security). *A signature scheme* SPS *is* unforgeable against auxiliary chosen-message attacks (UF-XCMA-secure) *for relation* $\mathcal{R}$ *if, for all p.p.t. adversaries* $\mathcal{A}$*, advantage function*

$$\mathrm{Adv}^{\mathsf{uf\text{-}xcma}}_{\mathsf{SPS}}(\mathcal{A}) := \Pr\left[\textsc{Ver}(\mathsf{M}^*, \sigma^*) = 1 \;\middle|\; \begin{array}{l} \mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda); \\ (\mathsf{M}^*, \sigma^*) \xleftarrow{\$} \mathcal{A}^{\textsc{Init},\textsc{Sign}(\cdot,\cdot)}(\mathsf{par}) \end{array}\right]$$

is negligible in security parameter λ *where*

- $\textsc{Init}$ *runs* $(pk, sk) \xleftarrow{\$} \mathsf{Gen}(\mathsf{par})$*, initializes* $\mathcal{Q}_\mathsf{M}$ *with* $\emptyset$*, and returns* pk *to* $\mathcal{A}$*,*
- $\textsc{Sign}(\mathsf{M}, \mathsf{m})$ *checks if* $\mathcal{R}(\mathsf{M}, \mathsf{m}) = 1$*, runs* $\sigma \xleftarrow{\$} \mathsf{Sign}(sk, \mathsf{M})$*, adds the* M *to* $\mathcal{Q}_\mathsf{M}$*, and returns* σ *to* $\mathcal{A}$*, and*
- $\textsc{Ver}(\mathsf{M}^*, \sigma^*)$ *returns 1 if* $\mathsf{M}^* \notin \mathcal{Q}_\mathsf{M}$ *and* $1 = \mathsf{Ver}(pk, \mathsf{M}^*, \sigma^*)$*, or returns 0, otherwise.*

As we are concerned with structure-preserving schemes, we fix $\mathcal{R}(\mathsf{M}, \mathsf{m})$ to a relation that returns 1 iff $\mathsf{M} = G^\mathsf{m}$ where G is a generator in a group. This relation is sufficient for our purpose, that is, combining with a partial one-time signature scheme described below. By letting $\mathcal{R}$ be a constant function $\mathcal{R} = 1$, we obtain a standard notion of *unforgeability against chosen-message attacks* (UF-CMA-secure) and denote its advantage function by $\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SPS}}(\mathcal{A})$. UF-XCMA is slightly stronger than unforgeability against extended random message attacks (UF-XRMA) introduced by Abe et al.[2]. While UF-XRMA is relative to a preliminary fixed algorithm that chooses messages to sign, it is the adversary that selects messages in UF-XCMA. Thus, UF-XCMA implies UF-XRMA.

From UF-XCMA to UF-CMA. In this paper, we focus on constructing UF-XCMA secure structure-preserving signature and then transform it to a UF-CMA secure SPS by using a partial one-time signature (POS) scheme [12,2] in the standard way [2,37]. POS is also known as two-tier signature schemes and is a variation of one-time signatures where parts of keys are updated after every signing. Here we recall useful definitions of POS and the transform.

Definition 2.4 (Partial One-Time Signature Scheme [12]). *A partial one-time signature scheme* POS *with respect to* PGGen *is a set of polynomial-time algorithms* $(\mathsf{G}, \mathsf{Update}, \mathsf{S}, \mathsf{V})$ *that, for* $\mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda)$*:*

- $\mathsf{G}(\mathsf{par})$ *generates a long-term public key* pk *and secret key* sk*, and implicitly defines the associated message space* $\mathcal{M}_o$ *and the one-time public key space* $\mathcal{K}_{opk}$*.*
- $\mathsf{Update}(\mathsf{par})$ *takes* par *as input, and outputs a one-time key pair* (opk, osk)*.*
- $\mathsf{S}(sk, osk, \mathsf{M})$ *outputs a signature* σ *on message* M *based on* sk *and* osk*.*
- $\mathsf{V}(pk, opk, \mathsf{M}, \sigma)$ *outputs* 1 *for acceptance or* 0 *for rejection.*

(Perfect correctness.) *For all* $(pk, sk) \xleftarrow{\$} \mathsf{G}(\mathsf{par})$*, all* $(opk, osk) \xleftarrow{\$} \mathsf{Update}(\mathsf{par})$*, all messages* $\mathsf{M} \in \mathcal{M}$*, and all* $\sigma \xleftarrow{\$} \mathsf{S}(sk, osk, \mathsf{M})$*,* $\mathsf{V}(pk, opk, \mathsf{M}, \sigma) = 1$ *holds.*

POS *is structure-preserving if* pk*,* opk*,* M*, and* σ *consist only elements of* $\mathbb{G}$*, and* V *evaluates group membership testing and pairing product equations.*

We require POS to be *unforgeable against one-time non-adaptive chosen-message attacks* ($\mathsf{OT\text{-}nCMA}$), which is defined as follows.

Definition 2.5 ($\mathsf{OT\text{-}nCMA}$ Security). *A* POS *scheme is* unforgeable against one-time non-adaptive chosen-message attacks ($\mathsf{OT\text{-}nCMA}$) *if for any algorithm* $\mathcal{A}$*, the following advantage function* $\mathrm{Adv}^{\mathsf{ncma}}_{\mathsf{POS}}(\mathcal{A})$ *is negligible in* λ*,*

$$\mathrm{Adv}^{\mathsf{ncma}}_{\mathsf{POS}}(\mathcal{A}) := \Pr\left[\mathrm{VER}(opk^*, \mathsf{M}^*, \sigma^*) = 1 \;\middle|\; \begin{array}{l}\mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda);\\ (opk^*, \sigma^*, \mathsf{M}^*) \xleftarrow{\$} \mathcal{A}^{\mathrm{INIT}, \mathrm{SIGN}(\cdot)}(\mathsf{par})\end{array}\right]$$

where

- INIT *runs* $(pk, sk) \xleftarrow{\$} \mathsf{G}(\mathsf{par})$*, initializes* $\mathcal{Q}_\mathsf{M}$ *with* $\emptyset$*, and returns* pk *to* $\mathcal{A}$*.*
- $\mathrm{SIGN}(\mathsf{M})$ *runs* $(opk, osk) \xleftarrow{\$} \mathsf{Update}(\mathsf{par})$ *and* $\sigma \xleftarrow{\$} \mathsf{S}(sk, osk, \mathsf{M})$*, and then returns* (opk, σ) *to* $\mathcal{A}$*, and records* $(opk, \mathsf{M}, \sigma)$ *to the list* $\mathcal{Q}_\mathsf{M}$*.*
- $\mathrm{VER}(opk^*, \sigma^*, \mathsf{M}^*)$ *returns 1 if there exists* $(opk^*, \mathsf{M}, \sigma) \in \mathcal{Q}_\mathsf{M}$ *and* $\mathsf{M}^* \neq \mathsf{M}$ *and* $1 = \mathsf{V}(pk, opk^*, \mathsf{M}^*, \sigma^*)$*, or returns 0, otherwise.*

Let $\mathsf{POS} := (\mathsf{G}, \mathsf{Update}, \mathsf{S}, \mathsf{V})$ be a structure-preserving partially one-time signature scheme with message space $\mathcal{M}$ and one-time public key space $\mathcal{K}_{opk}$, and $\mathsf{xSPS} := (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Ver}')$ be a structure-preserving signature scheme with message space $\mathcal{K}_{opk}$. The transformed $\mathsf{UF\text{-}CMA}$ secure SPS scheme, $\mathsf{SPS} := (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$, is defined as follows.

$\underline{\mathsf{Gen}(\mathsf{par})}$:	$\underline{\mathsf{Sign}(sk, \mathsf{M})}$:	$\underline{\mathsf{Ver}(pk, \mathsf{M}, \sigma)}$:
$(pk_1, sk_1) \xleftarrow{\$} \mathsf{G}(\mathsf{par})$	$(opk, osk) \xleftarrow{\$} \mathsf{Update}(\mathsf{par})$	Parse $\sigma = (opk, \sigma_1, \sigma_2)$
$(pk_2, sk_2) \xleftarrow{\$} \mathsf{Gen}'(\mathsf{par})$	$\sigma_1 \xleftarrow{\$} \mathsf{S}(sk_1, osk, \mathsf{M})$	If $\mathsf{V}(pk_1, opk, \mathsf{M}, \sigma_1) = 1$
$pk := (pk_1, pk_2)$	$\sigma_2 \xleftarrow{\$} \mathsf{Sign}'(sk_2, opk)$	$\wedge \mathsf{Ver}'(pk_2, opk, \sigma_2) = 1$
$sk := (sk_1, sk_2)$	Return $(opk, \sigma_1, \sigma_2)$	then return 1
Return (pk, sk)		Else return 0

The correctness and structure-preserving property of SPS are implied by those of POS and xSPS in a straightforward way. The following theorem ([2, Theorem 3]) states $\mathsf{UF\text{-}CMA}$ security of SPS.

Theorem 2.1. *If* POS *is* OT-nCMA *secure and* xSPS *is* UF-XRMA *secure, then* SPS *defined as above is* UF-CMA *secure. In particular, for all adversaries* $\mathcal{A}$ *against* UF-CMA *security of* SPS*, there exist adversaries* $\mathcal{B}$ *against* OT-nCMA *security of* POS *and* $\mathcal{C}$ *against* UF-XRMA *security of* xSPS *with running times* $\mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{C})$ *and* $\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SPS}}(\mathcal{A}) \leq \mathrm{Adv}^{\mathsf{ncma}}_{\mathsf{POS}}(\mathcal{B}) + \mathrm{Adv}^{\mathsf{uf\text{-}xcma}}_{\mathsf{xSPS}}(\mathcal{C})$.

2.4 Public-Key Encryption Schemes

Definition 2.6 (Public-key encryption). *A Public-Key Encryption scheme (PKE) consists of algorithms* $\mathsf{PKE} := (\mathsf{Gen}_\mathsf{P}, \mathsf{Enc}, \mathsf{Dec})$*:*

- *The key generation algorithm* $\mathsf{Gen}_\mathsf{P}(\mathsf{par})$ *takes* $\mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda)$ *as input and generates a pair of public and secret keys* (pk, sk)*. Message space* $\mathcal{M}$ *is implicitly defined by* pk*.*
- *The encryption algorithm* $\mathsf{Enc}(pk, \mathsf{M})$ *returns a ciphertext* ct*.*
- *The deterministic decryption algorithm* $\mathsf{Dec}(sk, \mathsf{ct})$ *returns a message* M*.*

(Perfect correctness.) *For all* $\mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda)$*,* $(pk, sk) \xleftarrow{\$} \mathsf{Gen}_\mathsf{P}(\mathsf{par})$*, messages* $\mathsf{M} \in \mathcal{M}$*, and* $\mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(pk, \mathsf{M})$*,* $\mathsf{Dec}(sk, \mathsf{ct}) = \mathsf{M}$ *holds.*

Definition 2.7 (IND-mCPA Security [9]). *A PKE scheme* PKE *is* indistinguishable against multi-instance chosen-plaintext attack (IND-mCPA-secure) *if for any* $q_e \geq 0$ *and for all p.p.t. adversaries* $\mathcal{A}$ *with access to oracle* $\textsc{Enc}$ *at most* q_e *times the following advantage function* $\mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{A})$ *is negligible,*

$$\mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{A}) := \left| \Pr\left[b' = b \;\middle|\; \begin{array}{l} \mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda); (pk, sk) \xleftarrow{\$} \mathsf{Gen}_\mathsf{P}(\mathsf{par}); \\ b \xleftarrow{\$} \{0,1\}; b' \xleftarrow{\$} \mathcal{A}^{\textsc{Enc}(\cdot,\cdot)}(pk) \end{array} \right] - \frac{1}{2} \right|,$$

where $\textsc{Enc}(\mathsf{M}_0, \mathsf{M}_1)$ *runs* $\mathsf{ct}^* \xleftarrow{\$} \mathsf{Enc}(pk, \mathsf{M}_b)$*, and returns* ct^* *to* $\mathcal{A}$*.*

Some public-key encryption schemes, e.g., ElGamal encryption [24] and Linear encryption [16], are structure-preserving and satisfy IND-mCPA security with tight reductions to compact assumptions such as DDH and the Decision Linear assumption [16], respectively (cf. [34]).

2.5 The Groth-Sahai Proof System

We recall the Groth-Sahai proof system and its properties as a commit-and-prove scheme. We follow definitions by Escala and Groth in [26] in a simplified form that is sufficient for our purpose. For a given pairing group $\mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda)$, the GS-proof system is a non-interactive zero-knowledge proof (NIZK) system for satisfiability of a set of equations over par. Let $\mathcal{L}_{\mathsf{par}}$ be a family of NP languages defined over par. For a language $\mathcal{L} \in \mathcal{L}_{\mathsf{par}}$, let $R_\mathcal{L} := \{(x, \omega) : x \in \mathcal{L} \text{ and } \omega \in W(x)\}$ be a witness relation, where $W(x)$ is the set of witnesses for $x \in \mathcal{L}$. As our construction fixes the language in advance, it is sufficient for our purpose to define the proof system to be specific to $\mathcal{L}$ as follows.

Definition 2.8 (The Groth-Sahai Proof System). *The Groth-Sahai commit-and-prove system for* $\mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda)$ *and* $\mathcal{L} \in \mathcal{L}_{\mathsf{par}}$ *consists of p.p.t. algorithms* $\mathsf{GS} := (\mathsf{BG}, \mathsf{Com}, \mathsf{P}, \mathsf{V})$ *that:*

- $\mathsf{BG}(\mathsf{par})$ *is a binding common reference string generation algorithm that outputs* $\mathbf{crs}$.
- $\mathsf{Com}(\mathbf{crs}, \omega; r)$ *is a commitment algorithm that outputs a commitment* c *for given witness* ω *with randomness* $r \leftarrow \mathcal{R}_c$ *and* $\mathbf{crs}$.
- $\mathsf{P}(\mathbf{crs}, (x, \mathsf{c}), (\omega, r))$ *is a prover algorithm that returns a proof* ρ *on* $(x, \omega) \in R_{\mathcal{L}} \wedge \mathsf{c} = \mathsf{Com}(\mathbf{crs}, \omega; r)$.
- $\mathsf{V}(\mathbf{crs}, x, \mathsf{c}, \rho)$ *is a deterministic verification algorithm that returns 0 (reject) or 1 (accept).*

(Perfect correctness.) *For all* $\mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda)$, $\mathbf{crs} \xleftarrow{\$} \mathsf{BG}(\mathsf{par})$, $(x, \omega) \in R_{\mathcal{L}}$, *and* $r \in \mathcal{R}_c$, $\mathsf{V}(\mathbf{crs}, x, \mathsf{c}, \mathsf{P}(\mathbf{crs}, (x, \mathsf{c}), (\omega, r))) = 1$ *holds, where* $\mathsf{c} \leftarrow \mathsf{Com}(\mathbf{crs}, \omega; r)$.

When witness ω consists of several objects and only part of them are committed to c, commitments for the remaining part of the witness is prepared by P and included in the proof.

The following properties of the GS-proof system are used in this paper. For fully formal treatment, we refer to [26].

Definition 2.9 (Security properties of the Groth-Sahai proof system). *The following properties hold for all* $\mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda)$,

- ***Perfect Soundness.*** *For all* $\mathbf{crs} \in \mathsf{BG}(\mathsf{par})$, *all* $x \notin \mathcal{L}$, *all* c, *and all* ρ, *we have* $\mathsf{V}(\mathbf{crs}, x, \mathsf{c}, \rho) = 0$.
- ***CRS Indistinguishability.*** *There exists a algorithm* HG, *called the hiding common reference string generator that, for all adversaries* $\mathcal{A}$, *the following advantage function is negligible,*

$$\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{A}) := \left| \Pr\left[b' = b \;\middle|\; \begin{array}{l} \mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda); \\ \mathbf{crs}_0 \xleftarrow{\$} \mathsf{BG}(\mathsf{par}); (\mathbf{crs}_1, \mathsf{trap}) \xleftarrow{\$} \mathsf{HG}(\mathsf{par}); \\ b \xleftarrow{\$} \{0,1\}; b' \xleftarrow{\$} \mathcal{A}(\mathbf{crs}_b) \end{array} \right] - \frac{1}{2} \right|.$$

- ***Dual-mode Commitment.*** *For all* $\mathbf{crs} \in \mathsf{BG}(\mathsf{par})$, Com *is* perfectly binding. *Namely, for all* $w_0 \neq w_1$, *we have* $\{\mathsf{c}_0 \leftarrow \mathsf{Com}(\mathbf{crs}, w_0; r_0)\} \bigcap \{\mathsf{c}_1 \leftarrow \mathsf{Com}(\mathbf{crs}, w_1; r_1)\} = \emptyset$ *(where the sets are taken over* $r_0, r_1 \in \mathcal{R}_c$*).*

  For all $(\mathbf{crs}, \mathsf{trap}) \in \mathsf{HG}(\mathsf{par})$, Com *is* perfectly hiding. *Namely, for all* $\omega_0 \neq \omega_1$, *the following two distributions are identical:* $\{\mathsf{c}_0 \leftarrow \mathsf{Com}(\mathbf{crs}, \omega_0; r_0)\}$ *and* $\{\mathsf{c}_1 \leftarrow \mathsf{Com}(\mathbf{crs}, \omega_1; r_1)\}$, *where* $r_0, r_1 \in \mathcal{R}_c$.
- ***Perfect Zero-knowledge.*** *There exists a simulator* $\mathsf{Sim} := (\mathsf{SimCom}, \mathsf{SimP})$ *such that, for all* $(\mathbf{crs}, \mathsf{trap}) \in \mathsf{HG}(\mathsf{par})$, *and* $(x, \omega) \in R_{\mathcal{L}}$, *the following two distributions are identical:*

$$\{(\mathsf{c}, \rho) \mid r \xleftarrow{\$} \mathcal{R}_c; \mathsf{c} \leftarrow \mathsf{Com}(\mathbf{crs}, \omega; r); \rho \xleftarrow{\$} \mathsf{P}(\mathbf{crs}, (x, \mathsf{c}), (\omega, r))\}, \text{and}$$
$$\{(\mathsf{c}', \rho') \mid (\mathsf{c}', \gamma) \xleftarrow{\$} \mathsf{SimCom}(\mathbf{crs}, \mathsf{trap}); \rho' \xleftarrow{\$} \mathsf{SimP}(\mathbf{crs}, \mathsf{trap}, \gamma)\}.$$

  Since the above distributions are identical, it also holds for reused commitment and multiple adaptively chosen statements x *that involve the same witness and commitment. This implies perfect witness indistinguishability that two valid witnesses for a true instance yield proofs and commitments in the same joint distribution.*

The GS-proof system is structure-preserving for proving satisfiability of linear multi-scalar multiplication equations (MSEs) and a non-linear quadratic equation (QE). Regarding security, it is known that its CRS indistinguishability is tightly reduced to the SXDH assumption (cf. Theorem 4.3).

3 Generic Construction

In this section, we focus on a generic construction of a UF-XCMA-secure SPS scheme, xSPS. By coupling it with an off-the-shelf structure-preserving POS scheme, we obtain a UF-CMA-secure SPS scheme via Theorem 2.1.

3.1 Scheme Description

Let $\mathsf{par} \xleftarrow{\$} \mathsf{PGGen}(1^\lambda)$ be a set of system parameters. We represent a source group and its generator by $\mathbb{G}$ and G, respectively. Let $\mathsf{PKE} := (\mathsf{Gen}_\mathsf{P}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme, and $\mathsf{GS} := (\mathsf{BG}, \mathsf{Com}, \mathsf{P}, \mathsf{V})$ be the Groth-Sahai proof system for languages $\mathcal{L}_0$ and $\mathcal{L}_1$ defined below. Our SPS scheme $\mathsf{xSPS} := (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ is defined in Figure 1.

The correctness of xSPS is implied by that of the Groth-Sahai proof system, and the structure-preserving property is implied by that of the PKE scheme and the Groth-Sahai proof system.

Remark 3.1 (Role of proof ρ_0). The main role is to bind a message into a signature. In the real scheme, it is just a proof of the signing key x_0 in ct_0 (and c_0) since x_1 is fixed to 0. Yet the proof is bound to message M through randomness r_1 used for committing to x_1. In the security proof, it can be seen as an encrypted one-time message authentication code (MAC) of M and forces the adversary to reuse given signatures since, intuitively, the adversary cannot generate a new MAC for hidden keys x_0 and x_1.

Remark 3.2 (Role of proof ρ_1). ρ_1 is used for partitioning. It proves that two ciphertexts ct_0 and ct_1 are consistent (namely, the same plaintext is encrypted) *or* the plaintext in the ciphertext ct_2 is committed to in c_2. In the real scheme, ρ_1 proves the consistency of double encryption ct_0 and ct_1. In the security proof, ρ_1 enables us to achieve two (seemingly incompatible) functionalities under a binding mode CRS. One is forcing the adversary to use consistent ciphertexts in its forgery. A simulator guesses z_2^* in the forgery and makes $x_2 \neq z_2^*$ hold. The other is letting the simulator use inconsistent ciphertexts in a special situation achieved using a partitioning technique (see Section 3.2 for more details). In that situation, the simulator can make $x_2 = z_2$ hold and use a real witness of ρ_0.

Remark 3.3 (On the range of z_2). The range of z_2 is $\mathbb{Z}_p$ since z_2 is the plaintext of ct_2. Readers might think we should bind z_2 on $\{0,1\}$ by using a Groth-Sahai proof since the simulator in the security proof guesses z_2^* in the forgery as explained in the previous remark. This is not the case. In fact, even if an adversary uses z_2^* such that $z_2^* \notin \{0,1\}$, it has no advantage because the simualtor uses x_2 such

$\underline{\mathsf{Gen}(\mathsf{par})}$:
$\mathbf{crs}_0, \mathbf{crs}_1 \xleftarrow{\$} \mathsf{BG}(\mathsf{par})$; For $i = 0, 1, 2 : (pk_i, sk_i) \xleftarrow{\$} \mathsf{Gen}_{\mathsf{P}}(\mathsf{par})$
$x_0 \xleftarrow{\$} \mathbb{Z}_p$; $x_1 := x_2 := 0 \in \mathbb{Z}_p$; $r_0, r_1, r_2, t_0, t_1, t_2, t_3 \xleftarrow{\$} \mathcal{R}_c$
$\mathsf{c}_0 \leftarrow \mathsf{Com}(\mathbf{crs}_0, x_0; r_0)$; $\mathsf{c}_1 \leftarrow \mathsf{Com}(\mathbf{crs}_0, x_1; r_1)$; $\mathsf{c}_2 \leftarrow \mathsf{Com}(\mathbf{crs}_1, x_2; r_2)$
$\mathsf{k}_0 \leftarrow \mathsf{Com}(\mathbf{crs}_1, sk_0; t_0)$; $\mathsf{k}_1 \leftarrow \mathsf{Com}(\mathbf{crs}_1, sk_1; t_1)$; $\mathsf{k}_2 \leftarrow \mathsf{Com}(\mathbf{crs}_1, sk_2; t_2)$
$\mathsf{k}_3 \leftarrow \mathsf{Com}(\mathbf{crs}_0, sk_0; t_3)$
$pk := (\mathbf{crs}_0, \mathbf{crs}_1, (pk_i, \mathsf{c}_i)_{0\le i\le 2}, (\mathsf{k}_i)_{0\le i\le 3})$; $sk := ((sk_i, x_i, r_i)_{0\le i\le 2}, (t_i)_{0\le i\le 3}))$
Return (pk, sk)

$\underline{\mathsf{Sign}(sk, \mathsf{M} \in \mathbb{G})}$:
$z_0 := z_1 := x_0$; $z_2 := 0$; For $i = 0, 1, 2 : \mathsf{ct}_i \xleftarrow{\$} \mathsf{Enc}(pk_i, G^{z_i})$
$\mathsf{ins}_0 := (pk_0, \mathsf{ct}_0, \mathsf{M})$; $\mathsf{cv}_0 := (\mathsf{c}_0, \mathsf{c}_1, \mathsf{k}_3)$; $w_0 := (x_0, x_1, sk_0)$; $R_0 := (r_0, r_1, t_3)$
$\mathsf{ins}_1 := (pk_i, \mathsf{ct}_i)_{0\le i\le 2}$; $\mathsf{cv}_1 := (\mathsf{c}_2, (\mathsf{k}_i)_{0\le i\le 2})$; $w_1 := (x_2, (sk_i)_{0\le i\le 2})$
$R_1 := (r_2, (t_i)_{0\le i\le 2})$
$\rho_0 \xleftarrow{\$} \mathsf{P}(\mathbf{crs}_0, (\mathsf{ins}_0, \mathsf{cv}_0), (w_0, R_0))$ //Prove that $\mathsf{ins}_0 \in \mathcal{L}_0$ and w_0 is committed to in cv_0
$\rho_1 \xleftarrow{\$} \mathsf{P}(\mathbf{crs}_1, (\mathsf{ins}_1, \mathsf{cv}_1), (w_1, R_1))$ //Prove that $\mathsf{ins}_1 \in \mathcal{L}_1$ and w_1 is committed to in cv_1
Return $\sigma := (\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ct}_2, \rho_0, \rho_1)$

$\underline{\mathsf{Ver}(pk, \mathsf{M}, \sigma)}$:
Parse $\sigma := ((\mathsf{ct}_i)_{0\le i\le 2}, \rho_0, \rho_1)$
$\mathsf{ins}_0 := (pk_0, \mathsf{ct}_0, \mathsf{M})$; $\mathsf{cv}_0 := (\mathsf{c}_0, \mathsf{c}_1, \mathsf{k}_3)$; $\mathsf{ins}_1 := (pk_i, \mathsf{ct}_i)_{0\le i\le 2}$; $\mathsf{cv}_1 := (\mathsf{c}_2, \mathsf{k}_0, \mathsf{k}_1, \mathsf{k}_2)$
Return $(\mathsf{V}(\mathbf{crs}_0, \mathsf{ins}_0, \mathsf{cv}_0, \rho_0) \wedge \mathsf{V}(\mathbf{crs}_1, \mathsf{ins}_1, \mathsf{cv}_1, \rho_1))$

$\underline{\text{Languages:}}$
$\mathcal{L}_0 := \{ (pk_0, \mathsf{ct}_0, \mathsf{M}) \mid \exists x_0, x_1 \in \mathbb{Z}_p, sk_0 \in \mathcal{SK}$ s.t.
$G^{z_0} = G^{x_0}\mathsf{M}^{x_1} \wedge G^{z_0} = \mathsf{Dec}(sk_0, \mathsf{ct}_0) \wedge (pk_0, sk_0) \in \mathsf{Gen}_{\mathsf{P}}(\mathsf{par})\}$
$\mathcal{L}_1 := \{ (pk_i, \mathsf{ct}_i)_{0\le i\le 2} \mid \exists x_2 \in \mathbb{Z}_p, sk_0, sk_1, sk_2 \in \mathcal{SK}$ s.t.
$((z_0 - z_1)(x_2 - z_2) = 0) \wedge_{i=0}^{2} (G^{z_i} = \mathsf{Dec}(sk_i, \mathsf{ct}_i) \wedge (pk_i, sk_i) \in \mathsf{Gen}_{\mathsf{P}}(\mathsf{par}))\}$

Fig. 1: Our signature scheme xSPS.

that $x_2 \in \{0, 1\}$ in the security proof. Value z_2 affects ρ_1. However, to make a valid forgery by using $x_2 = z_2^*$ as a witness in ρ_1, adversaries have no choice but to use $z_2^* \in \{0, 1\}$ as long as $x_2 \in \{0, 1\}$. Accordingly, we do not need to bind z_2 on $\{0, 1\}$. This intuition is implemented formally in the proof of Lemma 3.14.

Remark 3.4 (On verifying correctness of pk). Verifying correctness of commitment k_i with respect to sk_i is not necessary for achieving UF-CMA security where keys are generated honestly by definition. But it may have to be verified (once for all at the time of publishing pk) if the scheme is used in an application where signers can be corrupted at the time of key generation.

Remark 3.5 (On XCMA and CMA security of xSPS.). We prove that xSPS is UF-XCMA for efficiency though, in fact, we can prove xSPS is UF-CMA. When we prove UF-CMA, a simulator does not have exponents of queried messages, but the simulator must generate proofs ρ_0 for $x_1 \neq 0$ under the *binding mode*

$\mathbf{crs}_0$ in the security proof (see Section 3.3 for details). This is achievable if ρ_0 is generated as a proof of "pairing product equations (PPEs)" (in both the real and simulated schemes). If the simulator has exponents, then ρ_0 is generated as a proof of "(linear) multiscalar multiplication equations", which is more efficient than that of PPEs. We not only upgrade UF-XCMA to UF-CMA but also achieve an SPS scheme for vector messages by combining our xSPS with (partial) one-time signature at very low cost [2]. Thus, we select the UF-XCMA-secure scheme. See also Section 4 for efficiency.

3.2 Overview of Security Proof

Our main goal is to implement an additional check of $\mathcal{A}$'s forgery $\sigma^* := (\mathsf{ct}_0^*, \mathsf{ct}_1^*, \mathsf{ct}_2^*, \rho_0^*, \rho_1^*)$. We not only verify Groth-Sahai proofs, but also check $Z_0^* \in \{G^{x_0} \cdot \mathsf{M}_i^{x_1}\}_{i=1}^{q_s}$ for $Z_0^* \leftarrow \mathsf{Dec}(sk_0, \mathsf{ct}_0^*)$. That is, we will force $\mathcal{A}$ to *reuse* an M_i in queried messages for Z_0^* (we will set $x_1 := 1$ to achieve this during the game transitions). With $\mathbf{crs}_0$ for ρ_0^* being in the perfect soundness mode, $\mathcal{A}$ is forced to fulfill $G^{z_0^*} = G^{x_0} \cdot \mathsf{M}^*$. This leads to a contradiction and $\mathcal{A}$ never wins.

To change the success forgery condition, we replace the value $z_0 := x_0$ in signatures of the signing oracle and the additional forgery check with a value $z_0 := \mathbf{RF}_k(\mu|_k)$ where $\mathbf{RF}_k : \{0,1\}^k \to \mathbb{Z}_p$ is truly random, and $\mu|_k$ is the k-bit prefix of a random binary encoding $\mu \in \{0,1\}^L$ of a signed message $\mathsf{M} \in \mathbb{G}$, where L is the smallest even integer that is equal to or larger than the bit size of p. Note that encoding μ appears only in the security proof (not in the real scheme). We start with $\mathbf{RF}_0(\epsilon) := x_0$ for the empty string ϵ. We will introduce more dependencies of z_0 on x_2 and z_2^* in ct_2^*.

To increase the entropy of z_0 (this will make z_0 unpredictable for M^* and force $\mathcal{A}$ to reuse z_0 from the signing oracle) and eventually set $z_0 := \mathbf{RF}_L(\mu)$, we replace $z_0 := \mathbf{RF}_k(\mu|_k)$ with $z_0 := \mathbf{RF}_{k+1}(\mu|_{k+1})$ step by step. At each step, we partition the signature space into two halves according to the $(k+1)$-th bit of μ. The partitioning bit is dynamically changed by z_2^* hidden in ct_2^*. At the beginning of the game, the simulator guesses the bit z_2^* used in a forgery and aborts if the guess is incorrect (z_2^* is accessible with the decryption key sk_2). Signature queries are created with a case distinction depending on the $(k+1)$-th bit $\mu[k+1]$ of μ. If $\mu[k+1]$ is equal to the guessed z_2^* from the forgery, nothing is changed in the signing process. However, if $\mu[k+1]$ is different from z_2^*, we use another independent random function $\mathbf{RF}'_k$ and set $z_1 := \mathbf{RF}'_k(\mu|_k)$ in the generated signature (i.e., more randomness is supplied).

Note that at this point, we want to change the encrypted values z_0, z_1 in the generated signature, while being able to decrypt the value z_0^* from the forgery (to implement the additional check mentioned above). Intuitively, we can do so since the proved statement $(z_0 - z_1)(x_2 - z_2) = 0$ guarantees a consistent double encryption with $z_0 = z_1$ precisely when $x_2 \neq z_2$. Hence, if we initially set up x_2 as $1 - z_2^*$ (using our guess for z_2^*), it is possible for the simulator to generate inconsistent double encryptions (with $z_0 \neq z_1$) whenever $\mu[k+1] = z_2 \neq z_2^*$. On the other hand, a decryption key for either z_0^* or z_1^* can be used to implement the final check on the adversary's forgery (since $z_0^* = z_1^*$). These observations

enable a Naor-Yung-like double encryption argument to modify the z_0, z_1 values in all generated signatures with $\mu[k+1] \neq z_2^*$.

After the above transition is iterated, all signatures are generated with (or checked for) $z_0 := z_1 := \mathbf{RF}_L(\mu)$ for a truly random function $\mathbf{RF}_L$. At this point, we can replace z_0 and z_1 with $z_0 := z_1 := \mathbf{RF}_L(\mu) + \mathsf{m}$ since $\mathbf{RF}_L(\mu)$ is an independently and uniformly random element.

We can replace $z_0 := z_1 := \mathbf{RF}_L(\mu) + \mathsf{m}$ with $z_0 := z_1 := x + \mathsf{m}$ in a similar way to the case from $\mathbf{RF}_0(\epsilon) = x$ to $\mathbf{RF}_L(\mu)$ (see the proof for the detail). Thus, we can force $\mathcal{A}$ to reuse an M_i in queried messages for Z_0^*, as we explained at the beginning of this section.

3.3 Security Proof

Theorem 3.1. *If* PKE *is* IND-mCPA*-secure and* GS *is a Groth-Sahai proof system, then* xSPS *(defined in Section 3.1) is* UF-XCMA*-secure. Particularly, for all adversaries $\mathcal{A}$, there exist adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ with running time $\mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B}_2)$ and*

$$\mathrm{Adv}^{\mathsf{uf\text{-}xcma}}_{\mathsf{xSPS}}(\mathcal{A}) \leq (8L+6)\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}_1) + 12L \cdot \mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{B}_2) + \frac{4Lq_s}{p},$$

where L is the smallest even integer that is equal or larger than the bit size of p.

Proof. Let $\mathcal{A}$ be an adversary against UF-XCMA security of xSPS. We prove Theorem 3.1 via Games G_0-G_3 defined in Figure 2. We use AdvG_i to denote the advantage of $\mathcal{A}$ in Game G_i.

G_0 is the real attack game. We have lemmata below.

Lemma 3.1. $\mathrm{AdvG}_0 = \mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{xSPS}}(\mathcal{A})$.

Lemma 3.2 (G_0 to G_1). *There exist adversaries $\mathcal{B}_1$ against CRS indistinguishability of* GS *and $\mathcal{B}_2$ against* IND-mCPA *security of* PKE *with running times $\mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{B}_2)$ and $\mathrm{AdvG}_0 \leq \mathrm{AdvG}_1 + (4L+3) \cdot \mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}_1) + 6L \cdot \mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{B}_2) + \frac{2Lq_s}{p}$.*

We prove Lemma 3.2 in Section 3.4.

Lemma 3.3 (G_1 to G_2). $\mathrm{AdvG}_1 = \mathrm{AdvG}_2$.

Proof. The changes in G_2 are:

- Switching x_1 from 0 to 1: since c_1 is already simulated and is independent of x_1 in G_1, pk is distributed identically in both G_1 and G_2.
- Switching Z_0 and Z_1 from $G^{\mathbf{F}(\mathsf{M}_j)}$ to $G^{\mathbf{F}(\mathsf{M}_j)} \cdot \mathsf{M}_j$: since $\mathbf{F}$ is a truly random function, $\{G^{\mathbf{F}(\mathsf{M}_j)}\}_{j=1}^{q_s}$ and $\{G^{\mathbf{F}(\mathsf{M}_j)} \cdot \mathsf{M}_j\}_{j=1}^{q_s}$ are distributed identically.

Thus, games G_1 and G_2 are identical. ■

Lemma 3.4 (G_2 to G_3). *There exist adversaries $\mathcal{B}_1$ against CRS indistinguishability of* GS *and $\mathcal{B}_2$ against* IND-mCPA *security of* PKE *with running times $\mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{B}_2)$ and $\mathrm{AdvG}_2 \leq \mathrm{AdvG}_3 + (4L+3) \cdot \mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}_1) + 6L \cdot \mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{B}_2) + \frac{2Lq_s}{p}$.*

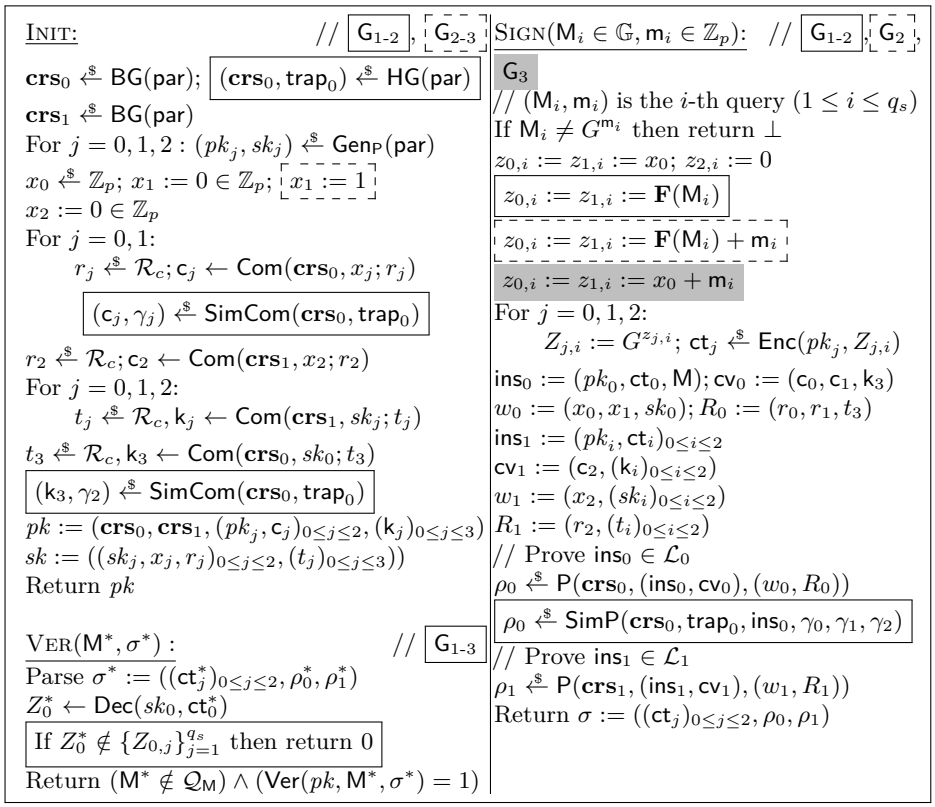

Fig. 2: Games G_0-G_3 for the proof of Theorem 3.1. Boxed code is only executed in the games marked in the same box style at the top right of every procedure. Non-boxed code is always run. $\mathbf{F} : \mathbb{G} \to \mathbb{Z}_p$ is a truly random function. $\mathcal{L}_0$ and $\mathcal{L}_1$ are languages defined in Section 3.1.

After switching $z_{0,i}$ and $z_{1,i}$ from $\mathbf{F}(\mathsf{M}_i)$ to $\mathbf{F}(\mathsf{M}_i) + \mathsf{m}_i$ in G_2, G_3 switches them from $\mathbf{F}(\mathsf{M}_i) + \mathsf{m}_i$ to $x_0 + \mathsf{m}_i$, which is exactly the step from G_0 to G_1, but in a reverse direction. The proof of Lemma 3.4 is similar to that of Lemma 3.2. The details are found in the full version of this paper.

Lemma 3.5 (G_3). $\mathrm{Adv}\mathsf{G}_3 = 0$.

Proof. In G_3, $\mathbf{crs}_0 \xleftarrow{\$} \mathsf{BG}(\mathsf{par})$ is in the binding mode. By the perfect soundness, $Z^*_0 = G^{x_0} \cdot \mathsf{M}^*$ if $\mathsf{V}(\mathbf{crs}_0, (pk_0, \mathsf{ct}^*_0, \mathsf{M}^*), (\mathsf{c}_0, \mathsf{c}_1, \mathsf{k}_3), \rho^*_0) = 1$. Since $\mathsf{M}^* \notin \mathcal{Q}_\mathsf{M}$, $Z^*_0 \notin \{Z_{0,j} = G^{\mathbf{F}(\mathsf{M}_j)} \cdot \mathsf{M}_j\}_{j=1}^{q_s}$ always holds and $\textsc{Ver}(\mathsf{M}^*, \sigma^*)$ outputs 0. ■

Summarizing Lemmata 3.1-3.5, we have Theorem 3.1. ■

3.4 From G_0 to G_1: Proof of Lemma 3.2

In this section, we prove Lemma 3.2. The proof proceeds via Games H_0-H_3 and $\mathsf{H}_{4,0}$-$\mathsf{H}_{4,L}$ defined in Figure 4 and Figure 3 gives an overview of the game transitions. The advantage of $\mathcal{A}$ in Game H_i is denoted by AdvH_i.

Game	$\mathbf{crs}_0$	$\mathbf{crs}_1$	$z_{0,i} = z_{1,i}$	ρ_0	Additional forgery check	Reduction
H_0	B	B	x_0	real	-	$\equiv \mathsf{G}_0$
H_1	B	B	x_0	real	$\boxed{Z_0^* \in \{G^{x_0}\}_{i=1}^{qs}}$	Soundness
H_2	$\boxed{\mathrm{H}}$	B	x_0	real	$Z_0^* \in \{G^{x_0}\}_{i=1}^{qs}$	CRS IND
H_3	H	B	x_0	$\boxed{\mathrm{sim}}$	$Z_0^* \in \{G^{x_0}\}_{i=1}^{qs}$	ZK
$\mathsf{H}_{4,0}$	H	$\boxed{\mathrm{H}}$	$\mathbf{RF}_0(\epsilon) := x_0$	sim	$Z_0^* \in \{G^{x_0}\}_{i=1}^{qs}$	CRS IND
$\mathsf{H}_{4,k}$	H	H	$\boxed{\mathbf{RF}_k(\mu_i\|_k)}$	sim	$\boxed{Z^*_{k \bmod 2} \in \{G^{\mathbf{RF}_k(\mu_i\|_k)}\}_{i=1}^{qs}}$	Loop
$\mathsf{H}_{4,k+1}$	H	H	$\boxed{\mathbf{RF}_{k+1}(\mu_i\|_{k+1})}$	sim	$\boxed{Z^*_{(k+1) \bmod 2} \in \{G^{\mathbf{RF}_{k+1}(\mu_i\|_{k+1})}\}_{i=1}^{qs}}$	
$\mathsf{H}_{4,L}$	H	H	$\mathbf{RF}_L(\mu_i\|_L)$	sim	$Z_0^* \in \{G^{\mathbf{RF}_L(\mu_i\|_L)}\}_{i=1}^{qs}$	Loop END
G_1	H	$\boxed{\mathrm{B}}$	$\mathbf{F}(\mathsf{M}_i) := \mathbf{RF}_L(\mu_i\|_L)$	sim	$Z_0^* \in \{G^{\mathbf{RF}_L(\mu_i\|_L)}\}_{i=1}^{qs}$	CRS IND

Fig. 3: Overview of transitions in Lemma 3.2. In the "$\mathbf{crs}_0$" and "$\mathbf{crs}_1$" columns, "B" (resp. "H") means that commitments are perfectly binding and proofs are perfectly sound (resp. commitments are perfectly hiding and proofs are perfectly zero-knowledge). In the "ρ_0" column, "real" (resp. "sim") means that proofs are generated by using the real witness w_0 (resp. the trapdoor trap). In the "reduction" column, we write what kind of security is used. "Soundness" (resp. "ZK") means the perfect soundness (resp. zero-knowledge) of the Groth-Sahai proof system.

We define $\mathsf{H}_0 := \mathsf{G}_0$ and have lemmata as follows.

Lemma 3.6 (H_0). $\mathrm{AdvH}_0 = \mathrm{AdvG}_0$.

Lemma 3.7 (H_0 to H_1). $\mathrm{AdvH}_1 = \mathrm{AdvH}_0$.

Proof. In H_1, $\mathbf{crs}_0 \xleftarrow{\$} \mathsf{BG}(\mathsf{par})$ is in the binding mode and the GS proof for $\mathcal{L}_0$ is perfectly sound. Then $Z_0^* = G^{x_0}$ holds if ρ_0 is accepted. Thus, H_1 and H_0 are identical. ■

Lemma 3.8 (H_1 to H_2). *There exists an adversary $\mathcal{B}$ against CRS indistinguishability with running time $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and $\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}) \geq |\mathrm{AdvH}_2 - \mathrm{AdvH}_1|$.*

Proof. Games H_2 and H_1 only differ in the distribution of $\mathbf{crs}_0$ returned by Init, namely, $\mathbf{crs}_0$ is in the hiding or binding mode. From that, we obtain a straightforward reduction to CRS indistinguishability of GS. ■

Lemma 3.9 (H_2 to H_3). $\mathrm{AdvH}_3 = \mathrm{AdvH}_2$.

Proof. Instead of using the prover algorithm P, H_3 generates ρ_0 and relevant commitments with the zero-knowledge simulator, Sim. By the perfect zero-knowledge property, $\mathsf{H}_3 = \mathsf{H}_2$. ■

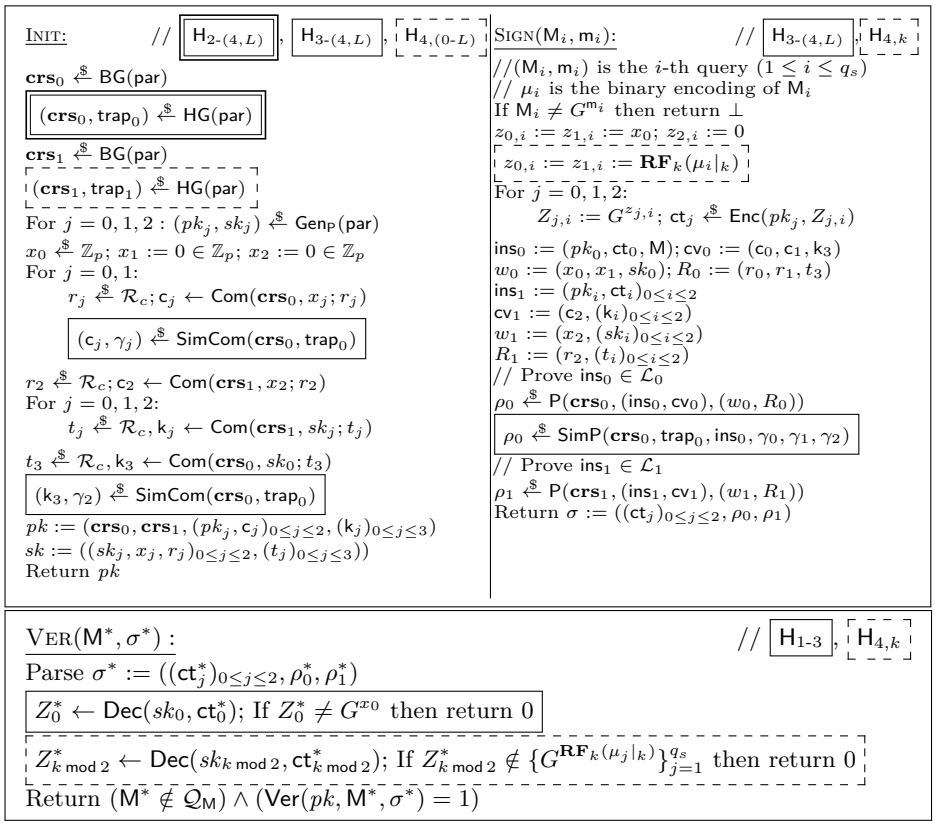

Fig. 4: Games H_0-H_3 and $\mathsf{H}_{4,0}$-$\mathsf{H}_{4,L}$ for the proof of Lemma 3.2. $\mathbf{RF}_k : \{0,1\}^k \to \mathbb{Z}_p$ is a truly random function. $\mu_i|_k$ is the first k bits of μ_i.

In $\mathsf{H}_{4,0}$, we syntactically define x_0 by $\mathbf{RF}_0(\epsilon)$ which is a fixed random element from $\mathbb{Z}_p$, and we have

Lemma 3.10 (H_3 to $\mathsf{H}_{4,0}$). *There exists an adversary $\mathcal{B}$ against CRS indistinguishability of* GS *with running time* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and* $\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}) \geq |\mathrm{AdvH}_{4,0} - \mathrm{AdvH}_3|$.

Proof. The only difference between $\mathsf{H}_{4,0}$ and H_3 is the simulation of $\mathbf{crs}_1$, which is generated by either BG (in H_3) or HG (in $\mathsf{H}_{4,0}$) since $\mathbf{RF}_0(\epsilon) = x_0$ and $\mu_j|_0 = \epsilon$ for all $j \in [q_s]$. From that, we obtain a straightforward reduction to CRS indistinguishability of GS. ■

Lemma 3.11 ($\mathsf{H}_{4,k}$ to $\mathsf{H}_{4,k+1}$). *There exist adversaries $\mathcal{B}_1$ against CRS indistinguishability of* GS *and $\mathcal{B}_2$ against* IND-mCPA *security of* PKE *with running times* $\mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{B}_2) \approx \mathbf{T}(\mathcal{A})$ *and* $\mathrm{AdvH}_{4,k} - \mathrm{AdvH}_{4,k+1} \leq 4\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}_1) + 6\mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{B}_2) + \frac{2q_s}{p}$

Proof. We define the games between $\mathsf{H}_{4,k}$ and $\mathsf{H}_{4,k+1}$ in Figure 5.

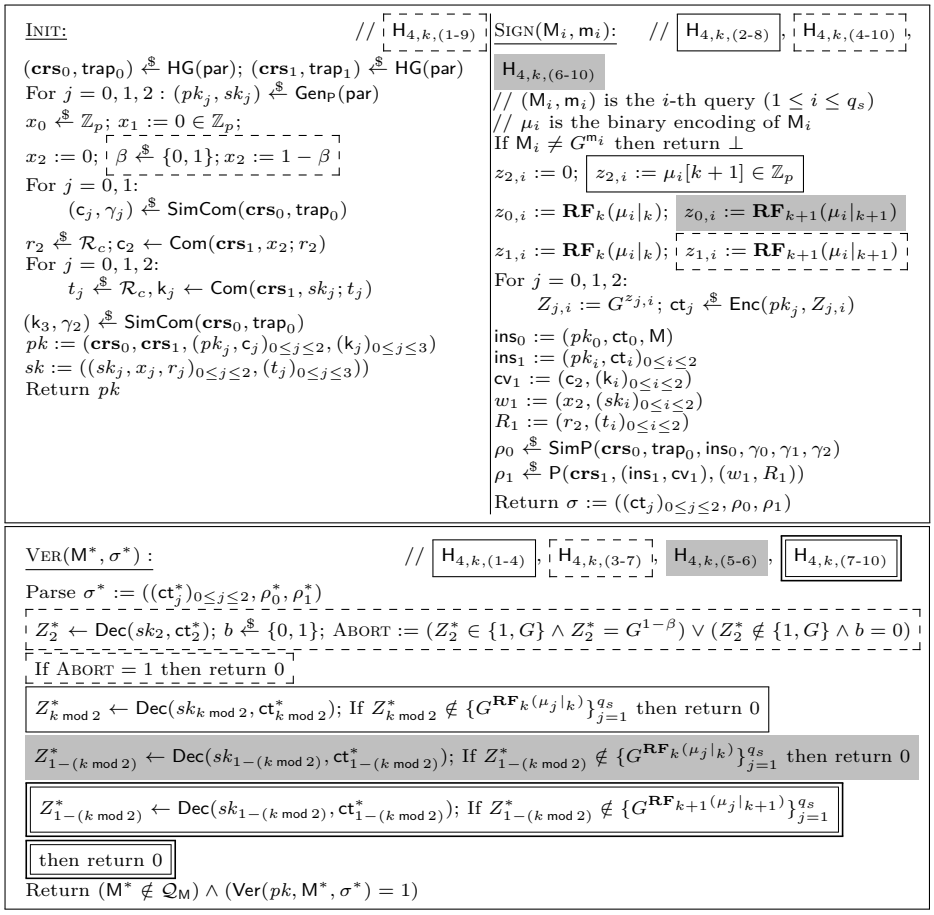

Fig. 5: Games $\mathsf{H}_{4,k,1}$-$\mathsf{H}_{4,k,10}$ for the proof of Lemma 3.11. $\mu[k]$ is the k-th bit of μ and $\mu|_k$ is the first k bits of μ. $\mathbf{RF}_{k+1} : \{0,1\}^{k+1} \to \mathbb{Z}_p$ is a truly random functions (defined by Equation (2)).

Lemma 3.12 ($\mathsf{H}_{4,k}$ to $\mathsf{H}_{4,k,1}$). $\mathrm{AdvH}_{4,k,1} = \mathrm{AdvH}_{4,k}$.

Proof. In $\mathsf{H}_{4,k,1}$, x_2 is switched from 0 to $1-\beta$, where $\beta \xleftarrow{\$} \{0,1\}$. Though $x_2 \neq z_{2,i}$ may happen in $\mathsf{H}_{4,k,1}$, still $z_{0,i} = z_{1,i}$ holds and hence ins_1 is in $\mathcal{L}_1$ in both games. Thus commitment $\mathsf{c}_2 \xleftarrow{\$} \mathsf{Com}(\mathbf{crs}_1, x_2)$ and proofs ρ_1 distribute identically in both games due to the witness indistinguishability under $\mathbf{crs}_1$ generated by $\mathsf{HG}(\mathsf{par})$. Thus, $\mathrm{AdvH}_{4,k,1} = \mathrm{AdvH}_{4,k}$. ■

Lemma 3.13 ($\mathsf{H}_{4,k,1}$ **to** $\mathsf{H}_{4,k,2}$)**.** *There exists an adversary* $\mathcal{B}$ *against* IND-mCPA *security of* PKE *with running time* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and* $\mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{B}) \geq |\mathrm{AdvH}_{4,k,2} - \mathrm{AdvH}_{4,k,1}|$.

Proof. In $\mathsf{H}_{4,k,2}$, ct_2 encrypts $Z_{2,i} = G^{\mu_i[k+1]}$, instead of $Z_{2,i} = G^0$. Observe that sk_2 is used only in making commitment k_2 and proof ρ_1 with $\mathbf{crs}_1$ generated by $\mathsf{HG}(\mathsf{par})$ in both games. Thus we can construct a straightforward reduction to bound the difference by IND-mCPA security of PKE by using perfect zero-knowledge simulator Sim for making ρ_1 and relevant commitments. ■

Lemma 3.14 ($\mathsf{H}_{4,k,2}$ **to** $\mathsf{H}_{4,k,3}$)**.** $\mathrm{AdvH}_{4,k,3} = \frac{1}{2}\mathrm{AdvH}_{4,k,2}$.

Proof. In $\mathsf{H}_{4,k,3}$, β and b are independent of adversary's view and chosen uniformly at random. c_2 perfectly hides β since $\mathbf{crs}_1$ is generated by $\mathsf{HG}(\mathsf{par})$ and the simulation of Sign is independent of β. Thus, the event Abort is independent of adversary's success event and

$$\begin{aligned}\Pr[\textsc{Abort}] &= \Pr[(z_2^* \in \{0,1\}) \wedge z_2^* = 1-\beta] + \Pr[z_2^* \notin \{0,1\} \wedge b = 0] \\ &= \frac{1}{2}\Pr[z_2^* \in \{0,1\}] + \frac{1}{2}(1 - \Pr[z_2^* \in \{0,1\}]) = \frac{1}{2},\end{aligned}$$

where z_2^* is the discrete log of Z_2^* based on G and independent of b. This only halves $\mathcal{A}$'s advantage. We note that, for all accepted forgeries in Games $\mathsf{H}_{4,k,3}$ to $\mathsf{H}_{4,k,8}$, the following equation holds:

$$z_2^* \neq x_2. \tag{1}$$

■

In the following games, we define the random function:

$$\mathbf{RF}_{k+1}(\mu|_{k+1}) := \begin{cases} \mathbf{RF}_k(\mu|_k) \ (\mu[k+1] = \beta) \\ \mathbf{RF}'_k(\mu|_k) \ (\mu[k+1] = 1-\beta) \end{cases}, \tag{2}$$

where $\mathbf{RF}_k$ and $\mathbf{RF}'_k$ are two independent random functions from $\{0,1\}^k \to \mathbb{Z}_p$. By the definition, we note that $\mathbf{RF}_{k+1} : \{0,1\}^{k+1} \to \mathbb{Z}_p$ is a random function.

Lemma 3.15 ($\mathsf{H}_{4,k,3}$ **to** $\mathsf{H}_{4,k,4}$)**.** *There exists an adversary* $\mathcal{B}$ *against* IND-mCPA *security of* PKE *with running time* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and* $\mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{B}) \geq |\mathrm{AdvH}_{4,k,4} - \mathrm{AdvH}_{4,k,3}|$.

Proof. In game $\mathsf{H}_{4,k,4}$, $x_2 = z_{2,i}$ holds if $\mu_i[k+1] \neq \beta$; otherwise $z_{0,i} = z_{1,i}$. If $\mu_i[k+1] = \beta$, then $z_{0,i} = z_{1,i} = \mathbf{RF}_k(\mu_i|_k)$, otherwise $x_2 = z_{2,i} = 1-\beta$ by Equation (2). Thus, in either case, $(z_{0,i} - z_{1,i})(x_2 - z_{2,i}) = 0$ holds and $\mathsf{ins}_1 \in \mathcal{L}_1$. Another difference between $\mathrm{AdvH}_{4,k,3}$ and $\mathsf{H}_{4,k,4}$ is that ct_1 is a ciphertext either of $Z_{1,i} = G^{\mathbf{RF}_{k+1}(\mu_i|_{k+1})}$ (in $\mathsf{H}_{4,k,4}$) or $Z_{1,i} = G^{\mathbf{RF}_k(\mu_i|_k)}$ (in $\mathrm{AdvH}_{4,k,3}$). Moreover, sk_1 is used only for making k_1 and ρ_1 with respect to $\mathbf{crs}_1$ generated by $\mathsf{HG}(\mathsf{par})$ in both games. Thus, as well as Lemma 3.13, we can construct a straightforward reduction to bound this difference by IND-mCPA-security of PKE using Sim for simulating ρ_1 and relevant commitments. Lemma 3.15 is concluded. ■

Lemma 3.16 ($\mathsf{H}_{4,k,4}$ **to** $\mathsf{H}_{4,k,5}$)**.** *There exists an adversary* $\mathcal{B}$ *against CRS indistinguishability of* GS *with running time* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and* $2\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}) \geq |\mathrm{AdvH}_{4,k,5} - \mathrm{AdvH}_{4,k,4}|$.

Proof. In $\mathsf{H}_{4,k,5}$, VER rejects a forgery if $Z^*_{1-(k \bmod 2)} \notin \{G^{\mathbf{RF}_k(\mu_j|_k)}\}_{j=1}^{q_s}$ instead of using $Z^*_{k \bmod 2}$. In these games, Equation (1) holds and we can switch $\mathbf{crs}_1$ to be binding and argue that $Z^*_{k \bmod 2} = Z^*_{1-(k \bmod 2)}$ by $z^*_2 \neq x_2$ and the perfect soundness of GS for language $\mathcal{L}_1$. More formally, we prove that via the game sequence in Figure 6. As shown in Lemma 3.15, ins_1 is always in $\mathcal{L}_1$ and we can

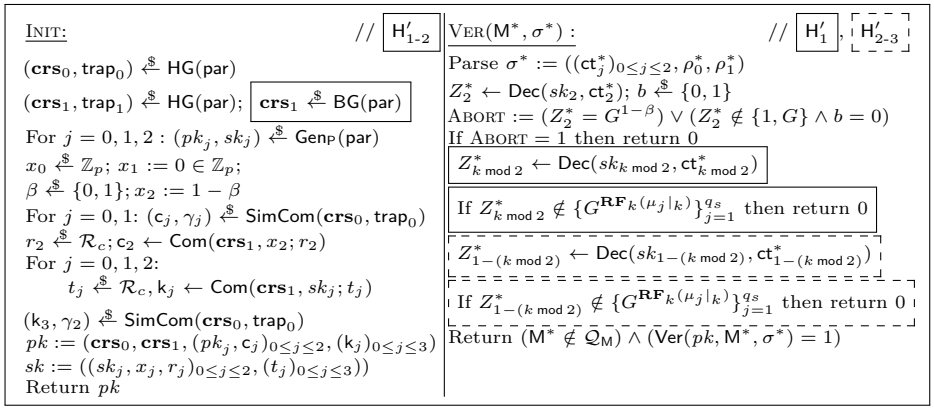

Fig. 6: Games H'_1-H'_3 for the proof of Lemma 3.16.

construct a straightforward reduction to show that there exists an adversary $\mathcal{B}$ against CRS indistinguishability of GS with

$$\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}) \geq |\mathrm{AdvH}'_1 - \mathrm{AdvH}_{4,k,4}|.$$

Since $\mathbf{crs}_1$ is binding in both H'_1 and H'_2, by the perfect soundness of GS and Equation (1), $Z^*_{k \bmod 2} = Z^*_{1-(k \bmod 2)}$ holds if ρ^*_1 gets verified. Hence, the changes between H'_1 and H'_2 are only conceptual, and thus $\mathrm{AdvH}'_2 = \mathrm{AdvH}'_1$. By the CRS indistinguishability of GS, we have $\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}) \geq |\mathrm{AdvH}'_3 - \mathrm{AdvH}'_2|$. It is clear that $\mathrm{AdvH}'_3 = \mathrm{AdvH}_{4,k,5}$ ■

Lemma 3.17 ($\mathsf{H}_{4,k,5}$ **to** $\mathsf{H}_{4,k,6}$)**.** *There exists an adversary* $\mathcal{B}$ *against* $\mathsf{IND\text{-}mCPA}$ *security of* PKE *with running time* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and* $\mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{B}) \geq |\mathrm{AdvH}_{4,k,6} - \mathrm{AdvH}_{4,k,5}|$.

Proof. In $\mathsf{H}_{4,k,6}$, $z_{0,i} = z_{1,i}$ is used as w_1. It holds that $(z_{0,i} - z_{1,i})(x_2 - z_{2,i}) = 0$ and $\mathsf{ins}_1 \in \mathcal{L}_1$ as the case in $\mathsf{H}_{4,k,5}$. In the signing oracle of $\mathsf{H}_{4,k,6}$, ct_0 encrypts $Z_{0,i} = G^{\mathbf{RF}_{k+1}(\mu_i|_{k+1})}$ instead of $Z_{0,i} = G^{\mathbf{RF}_k(\mu_i|_k)}$. Observe that sk_0 is used only

in making k_0 and ρ_1 with $\mathbf{crs}_1$ generated by $\mathsf{HG}(\mathsf{par})$ in both games. We thus can construct a straightforward reduction to bound the difference between $\mathsf{H}_{4,k,5}$ and $\mathsf{H}_{4,k,6}$ by IND-mCPA security using zero-knowledge simulator Sim for making ρ_1 and relevant commitments. ■

Lemma 3.18 ($\mathsf{H}_{4,k,6}$ **to** $\mathsf{H}_{4,k,7}$)**.** $\mathrm{Adv}\mathrm{H}_{4,k,6} \leq \mathrm{Adv}\mathrm{H}_{4,k,7} + \frac{q_s}{p}$.

Proof. According to Equation (2), the difference between $\mathsf{H}_{4,k,6}$ and $\mathsf{H}_{4,k,7}$ is that the accepted forgery with a $Z^*_{1-(k \bmod 2)}$ in either:

$$\begin{aligned}\mathcal{Z}_6 &:= \{G^{\mathbf{RF}_k(\mu_j|_k)}\}_{j=1}^{q_s} \\ &= \underbrace{\{G^{\mathbf{RF}_k(\mu_j|_k)} : \mu_j[k+1] = \beta\}_{j=1}^{q_s}}_{=:\mathcal{S}_1} \cup \{G^{\mathbf{RF}_k(\mu_j|_k)} : \mu_j[k+1] = 1-\beta\}_{j=1}^{q_s} \\ &\quad (\text{in } \mathsf{H}_{4,k,6})\end{aligned}$$

or

$$\mathcal{Z}_7 := \{G^{\mathbf{RF}_{k+1}(\mu_j|_{k+1})}\}_{j=1}^{q_s} = \mathcal{S}_1 \cup \{G^{\mathbf{RF}'_k(\mu_j|_k)} : \mu_j[k+1] = 1-\beta\}_{j=1}^{q_s} (\text{in } \mathsf{H}_{4,k,7}).$$

We note that, for those messages M where $\mu[k+1] = 1-\beta$ and $\mu|_k \in \mathcal{CM} := \{\mu_j|_k : \mu_j[k+1] = \beta\}_{j=1}^{q_s}$, the value $G^{\mathbf{RF}_k(\mu|_k)} \in \mathcal{S}_1$. Namely,

$$\begin{aligned}\mathcal{S}' &:= \mathcal{S}_1 \bigcap \{G^{\mathbf{RF}_k(\mu_j|_k)} : \mu_j[k+1] = 1-\beta\}_{j=1}^{q_s} \\ &= \{G^{\mathbf{RF}_k(\mu_j|_k)} : \mu_j[k+1] = 1-\beta \wedge \mu_j|_k \in \mathcal{CM}\}_{j=1}^{q_s}.\end{aligned}$$

We note that $\mathcal{S}'$ is not empty, since each element $G^{\mathbf{RF}_k(\mu_j|_k)}$ depends on k-*bit prefix* of μ_j. Thus, we can rewrite

$$\mathcal{Z}_6 = \mathcal{S}_1 \cup \underbrace{\{G^{\mathbf{RF}_k(\mu_j|_k)} : \mu_j[k+1] = 1-\beta \wedge \mu_j|_k \notin \mathcal{CM}\}_{j=1}^{q_s}}_{=:\mathcal{S}_2}.$$

We define the following game $\mathsf{H}_{4,k,6'}$ between $\mathsf{H}_{4,k,6}$ and $\mathsf{H}_{4,k,7}$. $\mathsf{H}_{4,k,6'}$ simulates Init and Sign as in $\mathsf{H}_{4,k,6}$, but differs in simulating Ver, where it only accepts forgery with $Z^*_{1-(k \bmod 2)} \in \mathcal{S}_1$. Precisely, $\mathsf{H}_{4,k,6'}$ simulates Ver as follows:

- Parse $\sigma^* := ((\mathsf{ct}^*_j)_{0 \leq j \leq 2}, \rho^*_0, \rho^*_1)$.
- $Z^*_2 \leftarrow \mathsf{Dec}(sk_2, \mathsf{ct}^*_2)$. If $Z^*_2 \neq G^\beta$ then return 0.
- $Z^*_{1-(k \bmod 2)} \leftarrow \mathsf{Dec}(sk_{1-(k \bmod 2)}, \mathsf{ct}^*_{1-(k \bmod 2)})$. If $Z^*_{1-(k \bmod 2)} \notin \mathcal{S}_1$ then return 0.
- Return $(\mathsf{M}^* \notin \mathcal{Q}_\mathsf{M}) \wedge (\mathsf{Ver}(pk, \mathsf{M}^*, \sigma^*) = 1)$.

We note that the value $\mathbf{RF}_k(\mu|_k)$ is perfectly hidden from $\mathcal{A}$ for $\mu[k+1] = 1-\beta$ and $\mu|_k \notin \mathcal{CM}$ since $\mathcal{A}$ only learns $\mathbf{RF}'_k(\mu|_k)$ from Sign by Equation (2) and $\mathbf{RF}$ and $\mathbf{RF}'$ are two independent random functions. Thus, even an unbounded adversary $\mathcal{A}$ can output a value in $\mathcal{S}_2$ with probability at most q_s/p and the following holds,

$$\mathrm{Adv}\mathrm{H}_{4,k,6} - \mathrm{Adv}\mathrm{H}_{4,k,6'} \leq \frac{q_s}{p}.$$

Compared to $\mathsf{H}_{4,k,6'}$, there are more valid forgeries in $\mathsf{H}_{4,k,7}$ and we have

$$\mathrm{AdvH}_{4,k,6'} \leq \mathrm{AdvH}_{4,k,7}.$$

Thus, $\mathrm{AdvH}_{4,k,6} - \mathrm{AdvH}_{4,k,7} \leq \frac{q_s}{p}$ and we conclude the lemma. ■

Lemma 3.19 ($\mathsf{H}_{4,k,7}$ **to** $\mathsf{H}_{4,k,8}$)**.** $\mathrm{AdvH}_{4,k,8} = 2\mathrm{AdvH}_{4,k,7}$.

Proof. $\mathsf{H}_{4,k,8}$ accepts a forgery no matter if $\textsc{Abort} = 1$ or not. By the same argument as in Lemma 3.14, this doubles the advantage of $\mathcal{A}$. ■

Note that we have stopped using sk_2 in $\mathsf{H}_{4,k,8}$. In $\mathsf{H}_{4,k,9}$, ct_2 encrypts $Z_{2,i} = G^0$ instead of $Z_{2,i} = G^{\mu_i[k+1]}$. By the same argument as Lemma 3.13, we have

Lemma 3.20 ($\mathsf{H}_{4,k,8}$ **to** $\mathsf{H}_{4,k,9}$)**.** *There exists an adversary* $\mathcal{B}$ *against* IND-mCPA *security of* PKE *with running time* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and* $\mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{B}) \geq |\mathrm{AdvH}_{4,k,9} - \mathrm{AdvH}_{4,k,8}|$.

Lemma 3.21 ($\mathsf{H}_{4,k,9}$ **to** $\mathsf{H}_{4,k,10}$)**.** $\mathrm{AdvH}_{4,k,10} = \mathrm{AdvH}_{4,k,9}$.

Proof. In $\mathsf{H}_{4,k,10}$, x_2 is switched from $1-\beta$ to 0 and ρ_1 is generated by using P instead of Sim. Since $\mathbf{crs}_1$ is generated by $\mathsf{HG}(\mathsf{par})$, $\mathsf{c}_2 \xleftarrow{\$} \mathsf{Com}(\mathbf{crs}_1, x_2)$ is distributed the same in both $\mathsf{H}_{4,k,9}$ and $\mathsf{H}_{4,k,10}$. So is ρ_1 by the perfect zero-knowledge property. Thus, $\mathrm{AdvH}_{4,k,10} = \mathrm{AdvH}_{4,k,19}$. ■

Lemma 3.22 ($\mathsf{H}_{4,k,10}$ **to** $\mathsf{H}_{4,k+1}$)**.** $\mathrm{AdvH}_{4,k+1} = \mathrm{AdvH}_{4,k,10}$.

Proof. $\mathsf{H}_{4,k,10}$ simulates $\textsc{Init}$ and $\textsc{Ver}$ the same as in $\mathsf{H}_{4,k}$ and $z_{0,i} = z_{1,i} = \mathbf{RF}_{k+1}(\mu_i|_{k+1})$. Thus, $\mathrm{AdvH}_{4,k,10} = \mathrm{AdvH}_{4,k+1}$. ■

From Lemmata 3.12 to 3.17, we have

$$\mathrm{AdvH}_{4,k} - 2\mathrm{AdvH}_{4,k,6} \leq |\mathrm{AdvH}_{4,k} - 2\mathrm{AdvH}_{4,k,6}| \leq 4\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}_1) + 5\mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{B}_2).$$

From Lemmata 3.19 to 3.22, we have

$$2\mathrm{AdvH}_{4,k,7} - \mathrm{AdvH}_{4,k+1} \leq |2\mathrm{AdvH}_{4,k,7} - \mathrm{AdvH}_{4,k+1}| \leq \mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{B}_2).$$

As $2\mathrm{AdvH}_{4,k,6} \leq 2\mathrm{AdvH}_{4,k,7} + \frac{2q_s}{p}$ (Lemma 3.18), we conclude Lemma 3.11 as

$$\mathrm{AdvH}_{4,k} - \mathrm{AdvH}_{4,k+1} \leq 4\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}_1) + 6\mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{B}_2) + 2q_s/p.$$

■

We syntactically define $\mathbf{F}(\mathsf{M}_i) := \mathbf{RF}_L(\mu_i)$ in G_1 since the binary representation of a group element is unique and have

Lemma 3.23 ($\mathsf{H}_{4,L}$ **to** G_1)**.** *There exists an adversary* $\mathcal{B}$ *against CRS indistinguishability of* GS *with running time* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and* $\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}) \geq |\mathrm{AdvG}_1 - \mathrm{AdvH}_{4,L}|$.

Proof. We note that L is the smallest even integer that is equal or larger than the bit size of p (namely, $L \bmod 2 = 0$). The only difference between G_1 and $\mathsf{H}_{4,L}$ is the simulation of $\mathbf{crs}_1$, which is generated by either BG (in G_1) or HG (in $\mathsf{H}_{4,L}$) since $\mathbf{F}(\mathsf{M}_i) = \mathbf{RF}_L(\mu_i)$. From that, we obtain a straightforward reduction to CRS indistinguishability of GS. ■

Combining Lemma 3.6 to 3.11 and Lemma 3.23, we have $\mathrm{AdvG}_0 \leq \mathrm{AdvG}_1 + 3\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}_1) + L \cdot (4\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{B}_1) + 6\mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{B}_2) + \frac{2q_s}{p})$ and conclude Lemma 3.2.

4 Instantiation

We instantiate our generic construction in Type-III bilinear groups under the SXDH assumption. Throughout this section, we denote group elements in $\mathbb{G}_1$ with plain upper-case letters, such as X, and elements in $\mathbb{G}_2$ such letters with tilde, such as $\tilde{X}$. Scalar values in $\mathbb{Z}_p$ are denoted with lower-case letters. We may also put a tilde to scalar values or other objects when they are related to group elements in $\mathbb{G}_2$ in a way that is clear from the context.

We begin with optimizations in Section 4.1 made on top of the generic construction. We then present a concrete scheme for signing unilateral messages in Section 4.2 and for bilateral messages in Section 4.3 followed by full details of the Groth-Sahai proofs in Section 4.4.

4.1 ElGamal Encryption with Common Randomness

Observe that relation $(z_0 - z_1)(x_2 - z_2) = 0$ in $\mathcal{L}_1$ is a quadratic equation and it can be proved efficiently by a GS proof if z_0 and z_1 are committed in the same group and z_2 is committed in the other group. Relevant encryptions should follow the deployment of groups. We thus build the first two ciphertexts, ct_0 and ct_1 in $\mathbb{G}_1$, and ct_2 in $\mathbb{G}_2$.

To gain efficiency, we consider using the same randomness for making ct_0 and ct_1. For this to be done without spoiling the security proof, it is sufficient that one of the ciphertext ct_b is perfectly simulated given the other ciphertext ct_{1-b}. Formally, we assume that there exists a function, say SimEnc, such that, for any key pairs $(pk, sk) \xleftarrow{\$} \mathsf{Gen}_{\mathsf{P}}(\mathsf{par})$ and $(pk', sk') \xleftarrow{\$} \mathsf{Gen}_{\mathsf{P}}(\mathsf{par})$, any messages m and m' in the legitimate message space, and any randomness s, it holds that $\mathsf{Enc}(pk', m'; s) = \mathsf{SimEnc}(sk', m', \mathsf{Enc}(pk, m; s))$. In [10], Bellare et al. formally defined such a property as *reproducibility*. Given reproducible PKE and its ciphertext $\mathsf{ct}_b \xleftarrow{\$} \mathsf{Enc}(pk_b, G^{z_b}; s)$, we can compute another ciphertext $\mathsf{ct}_{1-b} \xleftarrow{\$} \mathsf{SimEnc}(sk_{1-b}, G^{z_{1-b}}, \mathsf{ct}_b)$ without knowing sk_b or s. All reduction steps with respect to the CPA security of PKE go through using SimEnc and simulated GS proofs. Precisely, we use SimEnc in Lemma 3.15 to compute ct_0 from given ct_1. Similar adjustment applies to Lemma 3.17.

As shown in [10], ElGamal encryption (EG) is reproducible. Let (y, G^y) and $(y', G^{y'}) \in \mathbb{Z}_p \times \mathbb{G}_1$ be two key pairs of ElGamal encryption. Given ciphertext $(M \cdot (G^y)^s, G^s)$ of message M with s and public key G^y, one can compute

$(M' \cdot (G^s)^{y'}, G^s)$ for any M' using secret key y'. It is exactly the same ciphertext obtained from the regular encryption with common randomness s. We thus encrypt z_0 and z_1 with ElGamal encryption in $\mathbb{G}_1$ using the same randomness and removing redundant G^s. For encrypting z_2, we also use ElGamal but in $\mathbb{G}_2$. Bellare et al. show that the multi-message chosen-plaintext security for each encryption holds under the DDH assumption in respective groups, which is directly implied by the SXDH assumption [9]. We thus have:

Theorem 4.1. *For all adversaries $\mathcal{A}$ against* IND-mCPA *security of* EG*, there exists an adversary $\mathcal{C}$ against the SXDH assumption with running time $\mathbf{T}(\mathcal{C}) \approx \mathbf{T}(\mathcal{A})$ and $\mathrm{Adv}^{\mathsf{mcpa}}_{\mathsf{PKE}}(\mathcal{A}) \leq 2\,\mathrm{Adv}^{\mathsf{sxdh}}_{\mathsf{PGGen}}(\mathcal{C}) + \frac{1}{p}$.*

4.2 Concrete Scheme for Unilateral Messages

We present a concrete scheme, SPSu1, for signing messages in $\mathbb{G}_1$. We use a structure-preserving one-time signature scheme, POSu1, taken from the results of Abe et al. [2], and the SXDH-based instantiation of GS proof system. The description of POSu1 is blended into the description of SPSu1. For the GS proofs, however, we only show concrete relations in this section and present details of computation in Section 4.4.

We use notations $[x]_i$ and $[\tilde{x}]_1$ as a shorthand of $\mathsf{Com}(\mathbf{crs}_i, x)$ and $\mathsf{Com}(\widetilde{\mathbf{crs}}_1, x)$, respectively. We abuse these notations to present witnesses in a relation. It is indeed useful to keep track which CRS and which source group is used to commit to a witness. This notational convention is used in the rest of the paper.

Scheme SPSu1**:** Let $\mathsf{par} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, \tilde{G})$ be a description of Type-III bilinear groups generated by $\mathsf{PGGen}(1^\lambda)$.

SPSu1.Gen(par). Generates $\mathbf{crs}_0$, and $(\mathbf{crs}_1, \widetilde{\mathbf{crs}}_1)$ as shown in (13). Picks $x_0 \xleftarrow{\$} \mathbb{Z}_p$ and set $x_1 = x_2 := 0$. Generates three ElGamal keys $\tilde{Y}_0 := \tilde{G}^{y_0}$, $\tilde{Y}_1 := \tilde{G}^{y_1}$, and $Y_2 := G^{y_2}$ where $y_i \xleftarrow{\$} \mathbb{Z}_p$ for $i = 0, 1, 2$. Then computes commitments

$$
\begin{aligned}
[x_0]_0 &:= \mathsf{Com}(\mathbf{crs}_0, x_0; r_{x_{00}}), & [x_1]_0 &:= \mathsf{Com}(\mathbf{crs}_0, x_1; r_{x_{10}}),\\
[y_0]_0 &:= \mathsf{Com}(\mathbf{crs}_0, y_0; r_{y_{00}}), & [\tilde{x_2}]_1 &:= \mathsf{Com}(\widetilde{\mathbf{crs}}_1, x_2; r_{x_{21}}),\\
[y_0]_1 &:= \mathsf{Com}(\mathbf{crs}_1, y_0; r_{y_{01}}), & [y_1]_1 &:= \mathsf{Com}(\mathbf{crs}_1, y_1; r_{y_{11}}),\\
[\tilde{y_2}]_1 &:= \mathsf{Com}(\widetilde{\mathbf{crs}}_1, y_2; r_{y_{21}})
\end{aligned}
$$

as shown in Equation (14). Generates a persistent key pair of POSu1 by $w \xleftarrow{\$} \mathbb{Z}_p^*$, $\gamma_i \xleftarrow{\$} \mathbb{Z}_p^*$, $\tilde{G}_r := \tilde{G}^w$, and $\tilde{G}_i := \tilde{G}_r^{\gamma_i}$ for $i = 1, \ldots, n_1$. Outputs pk and sk defined as $pk := (G, \tilde{G}, \mathbf{crs}_0, \mathbf{crs}_1, \widetilde{\mathbf{crs}}_1, \tilde{Y}_0, \tilde{Y}_1, Y_2, [x_0]_0, [x_1]_0, [\tilde{x_2}]_1, [y_0]_0, [y_0]_1, [y_1]_1, [\tilde{y_2}]_1, \tilde{G}_r, \tilde{G}_1, \ldots, \tilde{G}_{n_1})$, and $sk := (x_0, y_0, y_1, y_2, r_{x_{00}}, r_{x_{10}}, r_{x_{21}}, r_{y_{00}}, r_{y_{01}}, r_{y_{11}}, r_{y_{21}}, w, \gamma_1, \ldots, \gamma_{n_1})$, where par and pk are implicitly included in pk and sk, respectively.

SPSu1.Sign(sk, M). Given sk as defined above and $\mathsf{M} =: (M_1, \ldots, M_{n_1}) \in \mathbb{G}_1^{n_1}$, proceeds as follows.

- Generate one-time $\mathsf{POSu1}$ key pair $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ and $\tilde{A} := \tilde{G}^{\alpha}$, and compute a one-time signature, (Z, R), by

$$Z := G^{\alpha - \rho\, w} \quad \text{and} \quad R := G^{\rho} \prod_{i=1}^{n_1} M_i^{-\gamma_i}\ , \tag{3}$$

  where $w, \gamma_1, \ldots, \gamma_{n_1}$ are taken from sk, and ρ is chosen uniformly from $\mathbb{Z}_p$.
- Encrypt $z_0 = z_1 := x_0$, and $z_2 := 0$ as $(\tilde{E}_{z_0}, \tilde{E}_{z_1}, \tilde{E}_s) := (\tilde{G}^{z_0}\tilde{Y}_0^s,\ \tilde{G}^{z_1}\tilde{Y}_1^s,\ \tilde{G}^s)$ and $(E_{z_2}, E_t) := (G^{z_2}Y_2^t, G^t)$, where $s, t \xleftarrow{\$} \mathbb{Z}_p$.
- Commit to z_0, z_1, and z_2 by $[z_0]_0$, $[z_0]_1$, $[z_1]_1$, and $[\tilde{z_2}]_1$, as described in equation (14).
- Using $\mathbf{crs}_0$, commitments $[x_0]_0$, $[x_1]_0$, and $[y_0]_0$ in pk, and default commitment $[1]_0$ computed with randomness $0 \in \mathbb{Z}_p$, as shown in equation (15), compute GS proofs $\rho_{0,0}$ and $\rho_{0,1}$ for relations

$$\rho_{0,0} : \tilde{G}^{[z_0]_0}(\tilde{G}^{-1})^{[x_0]_0}(\tilde{A}^{-1})^{[x_1]_0} = 1, \text{ and} \qquad \text{(linear MSE in } \mathbb{G}_2) \tag{4}$$

$$\rho_{0,1} : \tilde{E}_{z_0}^{[1]_0}(\tilde{G}^{-1})^{[z_0]_0}(\tilde{E}_s^{-1})^{[y_0]_0} = 1 \qquad \text{(linear MSE in } \mathbb{G}_2) \tag{5}$$

  that correspond to clauses $\tilde{G}^{z_0} = \tilde{G}^{x_0} \cdot \tilde{\mathsf{M}}^{x_1}$ for $\tilde{\mathsf{M}} := \tilde{A}$ and $(\tilde{E}_{z_0}, \tilde{E}_s) \in \mathsf{Enc}(\tilde{Y}_0, \tilde{G}^{z_0})$ in $\mathcal{L}_0$, respectively.
- Similarly, using $(\mathbf{crs}_1, \widetilde{\mathbf{crs}}_1)$ and default commitments $[1]_1$ and $[\tilde{1}]_1$, computes GS proofs $\rho_{1,0}$, $\rho_{1,1}$, $\rho_{1,2}$, and $\rho_{1,3}$ for relations

$$\rho_{1,0} : ([\tilde{x_2}]_1 - [\tilde{z_2}]_1)([z_0]_1 - [z_1]_1) = 0, \qquad \text{(non-linear QE)} \tag{6}$$

$$\rho_{1,1} : \tilde{E}_{z_0}^{[1]_1}(\tilde{G}^{-1})^{[z_0]_1}(\tilde{E}_s^{-1})^{[y_0]_1} = 1, \qquad \text{(linear MSE in } \mathbb{G}_2) \tag{7}$$

$$\rho_{1,2} : \tilde{E}_{z_1}^{[1]_1}(\tilde{G}^{-1})^{[z_1]_1}(\tilde{E}_s^{-1})^{[y_1]_1} = 1, \text{ and} \qquad \text{(linear MSE in } \mathbb{G}_2) \tag{8}$$

$$\rho_{1,3} : E_{z_2}^{[\tilde{1}]_1}(G^{-1})^{[\tilde{z_2}]_1}(E_t^{-1})^{[\tilde{y_2}]_1} = 1, \qquad \text{(linear MSE in } \mathbb{G}_1) \tag{9}$$

  that correspond to clauses in $\mathcal{L}_1$.
- Output a signature $\sigma := (\tilde{A}, Z, R, \tilde{E}_{z_0}, \tilde{E}_{z_1}, \tilde{E}_s, E_{z_2}, E_t, [z_0]_0, [z_0]_1, [z_1]_1, [\tilde{z_2}]_1, \rho_{0,0}, \rho_{0,1}, \rho_{1,0}, \rho_{1,1}, \rho_{1,2}, \rho_{1,3})$.

$\underline{\mathsf{SPSu1.Ver}(pk, \mathsf{M}, \sigma).}$ Return 1 if all the following verifications are passed. Return 0, otherwise.

- Verify signature (Z, R) of $\mathsf{POSu1}$ for $\mathsf{M} = (M_1, \ldots, M_{n_1})$ with one-time key $\tilde{A}$ by

$$e(G, \tilde{A}) = e(Z, \tilde{G})\, e(R, \tilde{G}_r) \prod_{i=1}^{n_1} e(M_i, \tilde{G}_i). \tag{10}$$

- Verify all GS proofs $\rho_{0,0}, \rho_{0,1}, \rho_{1,0}, \rho_{1,1}, \rho_{1,2}, \rho_{1,3}$ with commitments $[z_0]_0$, $[z_0]_1$, $[z_1]_1$, $[\tilde{z_2}]_1$, and ciphertext $\tilde{E}_{z_0}$, $\tilde{E}_{z_1}$, $\tilde{E}_s$, E_{z_2}, E_t in σ, using $[x_0]_0$, $[x_1]_0$, $[y_0]_0$, $[\tilde{x_2}]_1$, $[y_0]_1$, $[y_1]_1$, $[\tilde{y_2}]_1$ in pk, as expressed in equations (17) and (19). Default commitments $[1]_1$ and $[\tilde{1}]_1$ are built on-the-fly following equation (15).

This completes the description of SPSu1.

PERFORMANCE. In Tables 1 and 2, we summarize the performance of SPSu1. Since computational cost largely depends on available resources and implementation, we only present basic dominant parameters. In bach verification, we consider the most aggressive case where all equations are wrapped into one. See Section 4.4 for more details about batch verification.

SECURITY. Regarding POSu1 used in the above construction, the following statement is proven in [2].

Theorem 4.2 ([2]). POSu1 *is* OT-nCMA *secure if the* DDH_2 *assumption holds with respect to* PGGen. *In particular, for all polynomial-time algorithms* $\mathcal{A}$ *there exists a polynomial-time algorithm* $\mathcal{B}$ *with* $\mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B})$ *and* $\mathrm{Adv}^{\mathsf{ncma}}_{\mathsf{POSu1}}(\mathcal{A}) \leq \mathrm{Adv}^{\mathsf{ddh}_2}_{\mathsf{PGGen}}(\mathcal{B}) + 1/p$.

With asymmetric pairing groups, CRS indistinguishability of GS proof system is tightly reduced from the SXDH assumption. Namely, the following theorem holds.

Theorem 4.3 ([31]). *For all adversaries* $\mathcal{A}$ *against CRS indistinguishability of* GS, *there exists an adversary* $\mathcal{B}$ *with running time* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and* $\mathrm{Adv}^{\mathsf{crsind}}_{\mathsf{GS}}(\mathcal{A}) \leq 2 \cdot \mathrm{Adv}^{\mathsf{sxdh}}_{\mathsf{PGGen}}(\mathcal{B})$.

Combining Theorems 2.1, 3.1, 4.1, 4.2, and 4.3, we have the following theorem.

Theorem 4.4. SPSu1 *is* UF-CMA *if the* SXDH *assumption holds with respect to* PGGen. *In particular, for any polynomial-time algorithm* $\mathcal{A}$, *there exists a polynomial-time algorithm* $\mathcal{B}$ *that runs in almost the same as* $\mathcal{A}$ *and*

$$\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SPSu1}}(\mathcal{A}) \leq (40L + 13) \cdot \mathrm{Adv}^{\mathsf{sxdh}}_{\mathsf{PGGen}}(\mathcal{B}) + \frac{4L(q_s + 3) + 1}{p}. \tag{11}$$

If we have $L = \log_2 p = 256$ for the targeted 128-bit security level, for instance, the security loss of SPSu1 is approximately in 13 bits ($2^{13.3}$).

4.3 Concrete Scheme for Bilateral Messages

To sign bilateral messages $(\mathsf{M}_1, \mathsf{M}_2) \in \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2}$, we use SPSu1 in the previous section to sign $\mathsf{M}_1 \in \mathbb{G}_1^{n_1}$ and combine it with another POS, say POSu2, that signs $\mathsf{M}_2 \in \mathbb{G}_2^{n_2}$. Since a one-time public key of POSu2 is in $\mathbb{G}_1$, it can be appended to M_1 and authenticated by SPSu1 by extending the message space to $\mathbb{G}_2^{n_1+1}$. The signing and verification procedure of POSu2 is analogous to POSu1 shown in the construction of SPSu1 with $\mathbb{G}_1$ and $\mathbb{G}_2$ interchanged. POSu2 is OT-nCMA if DDH_1 holds. Therefore, for the resulting scheme, that we denote SPSb, the following theorem holds by combining Theorem 4.4 and Theorem 4.2 for POSu2.

PERFORMANCE. Regarding the performance of SPSb, the only difference from SPSu2 is the cost due to POSu2. Concrete numbers obtained by inspection of the scheme are shown in Tables 1 and 2.

Object	#(elements)	#(s.mult)	Verification	
			#(equations)	#(pairings)
CRS in $\mathbb{G}_1$	(3, 0)	(3, 0)	-	-
CRS in $\mathbb{G}_2$	(0, 3)	(0, 3)	-	-
Commitment $[w]$ for $w \in \mathbb{Z}_p$	(2, 0)	(3, 0)	-	-
Commitment $[\tilde{w}]$ for $w \in \mathbb{Z}_p$	(0, 2)	(0, 3)	-	-
Commitment $[b]$ for $b \in \{0,1\}$	(2, 0)	(2, 0)	-	-
Commitment $[\tilde{b}]$for $b \in \{0,1\}$	(0, 2)	(0, 2)	-	-
Proof of linear MSE in $\mathbb{G}_1$	(1, 0)	(1.5, 0)	2	4
Proof of linear MSE in $\mathbb{G}_2$	(0, 1)	(0, 1.5)	2	4
Proof of non-linear QE	(2, 2)	(3, 3)	4	16

Table 3: Sizes and computational costs for GS proofs in the SXDH assumption setting for relations used in our construction. Default generators G and $\tilde{G}$ are *not* included in CRS. Column #(s.mult) indicates number of scalar multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$ for generating object by counting multi-scalar multiplication as 1.5. Linear MSE and non-linear QE are specific to relations in Equation (4) to (9).

SECURITY. Theorem 4.2 holds for $\mathsf{POSu2}$ under the DDH_1 assumption. Combining it with Theorem 4.4, we obtain the following.

Theorem 4.5. SPSb *is* $\mathsf{UF\text{-}CMA}$ *if the* SXDH *assumption holds with respect to* PGGen. *In particular, for any polynomial-time algorithm* $\mathcal{A}$*, there exists an algorithm* $\mathcal{B}$ *with* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ *and*

$$\mathrm{Adv}^{\mathsf{uf\text{-}cma}}_{\mathsf{SPSb}}(\mathcal{A}) \leq (40L+14) \cdot \mathrm{Adv}^{\mathsf{sxdh}}_{\mathsf{PGGen}}(\mathcal{B}) + \frac{4L(q_s+3)+2}{p}. \tag{12}$$

4.4 Specific Groth-Sahai Proofs under SXDH

Among wide variations of relations that are provable with GS proofs, our instantiation involves only three types of relations; linear multiscalar multiplication equations (MSEs) in $\mathbb{G}_1$ and $\mathbb{G}_2$, and non-linear quadratic equations (QEs). Witnesses are committed in either $\mathbb{G}_1$ or $\mathbb{G}_2$ depending on the relations to prove. We summarize the space and computation complexity in Table 3 and give details in the sequel.

CRS Generation: Our construction includes three independent common reference strings, $\mathbf{crs}_0$ and $(\mathbf{crs}_1, \widetilde{\mathbf{crs}}_1)$ generated in the binding mode as

$$\mathbf{crs}_0 := \begin{pmatrix} G & Q_0 \\ U_0 & V_0 \end{pmatrix}, \quad \mathbf{crs}_1 := \begin{pmatrix} G & Q_1 \\ U_1 & V_1 \end{pmatrix}, \quad \widetilde{\mathbf{crs}}_1 := \begin{pmatrix} \tilde{G} & \tilde{Q}_1 \\ \tilde{U}_1 & \tilde{V}_1 \end{pmatrix}, \tag{13}$$

where, for $\chi_0, \xi_0, \chi_1, \xi_1, \tilde{\chi}_1, \tilde{\xi}_1 \xleftarrow{\$} \mathbb{Z}_p^*$, $Q_i := G^{\chi_i}$, $U_i := G^{\xi_i}$, $V_i := G^{\chi_i \xi_i}$ for $i = 0, 1$ and $\tilde{Q}_1 := \tilde{G}^{\tilde{\chi}_1}$, $\tilde{U}_1 := \tilde{G}^{\tilde{\xi}_1}$, $\tilde{V}_1 := \tilde{G}^{\tilde{\chi}_1 \tilde{\xi}_1}$.

Scalar Commitments: To commit to $x \in \mathbb{Z}_p$ under $\mathbf{crs}_i$, compute

$$[x]_i := \mathsf{Com}(\mathbf{crs}_i, x; r) := (U_i^x\, G^r, (V_i\, G)^x\, Q_i^r), \tag{14}$$

where $r \in \mathbb{Z}_p$ is a fresh randomness. A default commitment of $1 \in \mathbb{Z}_p$ uses $0 \in \mathbb{Z}_p$ as a randomness, namely,

$$[1]_i := \mathsf{Com}(\mathbf{crs}_i, 1; 0) := (U_i, V_i\, G). \tag{15}$$

Commitment $[\tilde{x}]_1$ is computed analogously using elements in $\widetilde{\mathbf{crs}}_1$.

Proof of Scalar MSE: Proof $\rho_{0,0}$ for relation (4) as a linear MSE in $\mathbb{G}_1$ consists of a single element $\pi_{0,0} \in \mathbb{G}_2$ computed as

$$\pi_{0,0} := \tilde{G}^{r_{z_0}} (\tilde{G}^{-1})^{r_{x_0}} (\tilde{A}^{-1})^{r_{x_1}}, \tag{16}$$

where r_{z_0}, r_{x_0}, and r_{x_1} are random coins used to commit to z_0, x_0, x_1 by $[\tilde{z_0}]_0$, $[\tilde{x_0}]_0$, $[\tilde{x_1}]_0$, respectively. It is verified by evaluating

$$\begin{aligned} &e(C_{z_0,1}, \tilde{G})\, e(C_{x_0,1}, \tilde{G}^{-1})\, e(C_{x_1,1}, \tilde{A}^{-1}) = e(G, \pi_{0,0}), \text{ and} \\ &e(C_{z_0,2}, \tilde{G})\, e(C_{x_0,2}, \tilde{G}^{-1})\, e(C_{x_1,2}, \tilde{A}^{-1}) = e(Q_0, \pi_{0,0}), \end{aligned} \tag{17}$$

where $(C_{\mathsf{x},1}, C_{\mathsf{x},2}) := [\mathsf{x}]_0$ for $\mathsf{x} \in \{z_0, x_0, x_1\}$, and $\tilde{G}$ and Q_0 are taken from $\mathbf{crs}_0$.

Proofs $\rho_{0,1}$, $\rho_{1,1}$, and $\rho_{1,2}$, are for linear MSEs in exactly the same form as equation (4). They are generated and verified in the same manner as above.

Proof of Non-Linear QE: Proof $\rho_{1,0}$ for non-linear QE (6) consists of $(\theta_{1,0,1}, \theta_{1,0,2}, \pi_{1,0,1}, \pi_{1,0,2}) \in \mathbb{G}_1^2 \times \mathbb{G}_2^2$ that, $\psi \xleftarrow{\$} \mathbb{Z}_p$,

$$\begin{aligned} \theta_{1,0,1} &:= U_1^{z_0(r_{x_2} - r_{z_2}) - z_1(r_{x_2} - r_{z_2})}\, G^{(x_2 - z_2)(z_0 - z_1) - \psi}, \\ \theta_{1,0,2} &:= (V_1 G)^{z_0(r_{x_2} - r_{z_2}) - z_1(r_{x_2} - r_{z_2})}\, Q_1^{(x_2 - z_2)(z_0 - z_1) - \psi}, \\ \pi_{1,0,1} &:= \tilde{U}_1^{x_2(r_{z_0} - r_{z_1}) - z_2(r_{z_0} - r_{z_1})}\, \tilde{G}^{\psi}, \text{ and} \\ \pi_{1,0,2} &:= (\tilde{V}_1 \tilde{G})^{x_2(r_{z_0} - r_{z_1}) - z_2(r_{z_0} - r_{z_1})}\, \tilde{Q}_1^{\psi}, \end{aligned} \tag{18}$$

where r_{x} is a random coin used to commit to x. The verification evaluates

$$\begin{aligned} &e(C_{z_0,1} C_{z_1,1}^{-1}, \tilde{D}_{x_2,1})\, e(C_{z_0,1} C_{z_1,1}^{-1}, \tilde{D}_{z_2,1}^{-1}) = e(G, \pi_{1,0,1})\, e(\theta_{1,0,1}, \tilde{G}), \\ &e(C_{z_0,2} C_{z_1,2}^{-1}, \tilde{D}_{x_2,1})\, e(C_{z_0,2} C_{z_1,2}^{-1}, \tilde{D}_{z_2,1}^{-1}) = e(Q_1, \pi_{1,0,1})\, e(\theta_{1,0,2}, \tilde{G}), \\ &e(C_{z_0,1} C_{z_1,1}^{-1}, \tilde{D}_{x_2,2})\, e(C_{z_0,1} C_{z_1,1}^{-1}, \tilde{D}_{z_2,2}^{-1}) = e(G, \pi_{1,0,2})\, e(\theta_{1,0,1}, \tilde{Q}_1), \text{ and} \\ &e(C_{z_0,2} C_{z_1,2}^{-1}, \tilde{D}_{x_2,2})\, e(C_{z_0,2} C_{z_1,2}^{-1}, \tilde{D}_{z_2,2}^{-1}) = e(Q_1, \pi_{1,0,2})\, e(\theta_{1,0,2}, \tilde{Q}_1), \end{aligned} \tag{19}$$

where $(C_{\mathsf{x},1}, C_{\mathsf{x},2}) := [\mathsf{x}]_1$ for $\mathsf{x} \in \{z_0, z_1\}$, $(\tilde{D}_{\mathsf{y},1}, \tilde{D}_{\mathsf{y},2}) := [\tilde{\mathsf{y}}]_1$ for $\mathsf{y} \in \{x_2, z_2\}$, and other group elements are taken from $(\mathbf{crs}_1, \widetilde{\mathbf{crs}}_1)$.

Batch Verification: The number of pairing computations in equations (17) and (19) can be reduced when verifying proofs $\rho_{0,0}, \rho_{0,1}, \rho_{1,0}, \rho_{1,1}, \rho_{1,2}$ and $\rho_{1,3}$ at once by batch verification. By merging pairings with respect to G, $\tilde{G}$, Q_0, Q_1, $\tilde{Q}_1$, $\tilde{A}$, $\tilde{E}_{z_0}$, $\tilde{E}_s$, $\tilde{D}_{x_2,1}$, $\tilde{D}_{x_2,2}$, $\tilde{D}_{z_2,1}$, $\tilde{D}_{z_2,2}$, $\tilde{E}_{z_1}$, E_{z_2}, and E_t, we have a single pairing product equation consisting of 15 pairings. It will be merged further with the verification equations for the POS part that includes pairings involving G and $\tilde{G}$. For $\mathsf{SPSu1}$, the batch verification equation consists of $n_1 + 16$ pairings, of which $n_1 + 1$ pairings are from $\mathsf{POSu1}$. For SPSb, it consists of $n_1 + n_2 + 18$ pairings, of which $n_1 + n_2 + 3$ pairings are from POSb.

References

1. Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. *EUROCRYPT 2012*, *LNCS* 7237, pp. 572–590. Springer, 2012. 3
2. Masayuki Abe, Melissa Chase, Bernardo David, Markulf Kohlweiss, Ryo Nishimaki, and Miyako Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. *Journal of Cryptology*, 29(4):833–878, 2016. 2, 8, 9, 14, 25, 27
3. Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. *Journal of Cryptology*, 29(2):363–421, 2016. 1, 2, 4
4. Masayuki Abe, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. *CRYPTO 2011*, *LNCS* 6841, pp. 649–666. Springer, 2011. 2
5. Tolga Acar, Kristin Lauter, Michael Naehrig, and Daniel Shumow. Affine pairings on ARM. *PAIRING 2012*, *LNCS* 7708, pp. 203–209, Springer, 2012. 4
6. Diego F. Aranha, Laura Fuentes-Castañeda, Edward Knapp, Alfred Menezes, and Francisco Rodríguez-Henríquez. Implementing pairings at the 192-bit security level. *PAIRING 2012*, *LNCS* 7708, pp. 177–195, Springer, 2012. 3
7. Nuttapong Attrapadung, Goichiro Hanaoka, and Shota Yamada. A framework for identity-based encryption with almost tight security. *ASIACRYPT 2015, Part I*, *LNCS* 9452, pp. 521–549. Springer, 2015. 6
8. Paulo S. L. M. Barreto, Craig Costello, Rafael Misoczki, Michael Naehrig, Geovandro C. C. F. Pereira, and Gustavo Zanon. Subgroup security in pairing-based cryptography. *LATINCRYPT 2015*, *LNCS* 9230, pp. 245–265. Springer, 2015. 4
9. Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. *EUROCRYPT 2000*, *LNCS* 1807, pp. 259–274. Springer, 2000. 10, 25
10. Mihir Bellare and Alexandra Boldyreva and Jessica Staddon. Randomness Re-use in Multi-recipient Encryption Schemeas. *PKC 2003*, *LNCS* 2567, pp. 85–99. Springer, 2003. 24
11. Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. *EUROCRYPT 1996*, *LNCS* 1070, pp. 399–416. Springer, 1996. 3
12. Mihir Bellare and Sarah Shoup. Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. *PKC 2007*, *LNCS* 4450, pp. 201–216. Springer, 2007. 8, 9
13. Olivier Blazy, Georg Fuchsbauer, Malika Izabachène, Amandine Jambert, Hervé Sibert, and Damien Vergnaud. Batch Groth-Sahai. *ACNS 2010*, *LNCS* 6123, pp. 218–235. Springer, 2010. 3
14. Olivier Blazy, Eike Kiltz, and Jiaxin Pan. (Hierarchical) identity-based encryption from affine message authentication. *CRYPTO 2014, Part I*, *LNCS* 8616, pp. 408–425. Springer, 2014. 6
15. Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. *CRYPTO 2004*, *LNCS* 3152, pp. 443–459. Springer, 2004. 3
16. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. *CRYPTO 2004*, *LNCS* 3152, pp. 41–55. Springer, 2004. 10
17. Jan Camenisch, Maria Dubovitskaya, and Kristiyan Haralambiev. Efficient structure-preserving signature scheme from standard assumptions. *SCN 2012*, *LNCS* 7485, pp. 76–94. Springer, 2012. 2

18. Jan Camenisch, Maria Dubovitskaya, Kristiyan Haralambiev, and Markulf Kohlweiss. Composable and modular anonymous credentials: Definitions and practical constructions. *ASIACRYPT 2015, Part II*, *LNCS* 9453, pp. 262–288. Springer, 2015. 4
19. Julien Cathalo, Benoît Libert, and Moti Yung. Group encryption: Non-interactive realization in the standard model. *ASIACRYPT 2009*, *LNCS* 5912, pp. 179–196. Springer, 2009. 2
20. Melissa Chase and Markulf Kohlweiss. A new hash-and-sign approach and structure-preserving signatures from DLIN. *SCN 2012*, *LNCS* 7485, pp. 131–148. Springer, 2012. 2
21. Sanjit Chatterjee, Neal Koblitz, Alfred Menezes, and Palash Sarkar. Another look at tightness II: Practical issues in cryptography. Cryptology ePrint Archive, Report 2016/360, 2016. http://eprint.iacr.org/2016/360. 2
22. Jie Chen and Hoeteck Wee. Fully, (almost) tightly secure IBE and dual system groups. *CRYPTO 2013, Part II*, *LNCS* 8043, pp. 435–460. Springer, 2013. 3, 5, 6
23. Benoît Chevallier-Mames. An efficient CDH-based signature scheme with a tight security reduction. *CRYPTO 2005*, *LNCS* 3621, pp. 511–526. Springer, 2005. 3
24. Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *CRYPTO 1884*, *LNCS* 196, pp. 10–18. Springer, 1984. 10
25. Andreas Enge and Jérôme Milan. Implementing cryptographic pairings at standard security levels. *SPACE 2004*, *LNCS* 8804, pp. 28–46. Springer, 1984. 3, 4
26. Alex Escala and Jens Groth. Fine-tuning Groth-Sahai proofs. *PKC 2014*, *LNCS* 8383, pp. 630–649. Springer, 2014. 6, 10, 11
27. Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Tightly CCA-secure encryption without pairings. *EUROCRYPT 2016, Part I*, *LNCS* 9665, pp. 1–27. Springer, 2016. 6
28. Robert Granger, Dan Page, and Nigel P. Smart. High security pairing-based cryptography revisited. *ANTS-VII 2006*, *LNCS* 4076, pp. 480–494.Springer, 2006. 3
29. Gurleen Grewal, Reza Azarderakhsh, Patrick Longa, Shi Hu, and David Jao. Efficient implementation of bilinear pairings on ARM processors. *SAC 2012*, *LNCS* 7707, pp. 149–165. Springer, 2013. 4
30. Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. *ASIACRYPT 2006*, *LNCS* 4284, pp. 444–459. Springer, 2006. 2
31. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. *SIAM J. Comput.*, 41(5):1193–1232, 2012. 1, 27
32. Dennis Hofheinz. Algebraic partitioning: Fully compact and (almost) tightly secure cryptography. *TCC 2016-A, Part I*, *LNCS* 9562, pp. 251–281. Springer, 2016. 3, 6
33. D. Hofheinz. Adaptive partitioning. *EUROCRYPT 2017, Part III*, *LNCS* 9562, pp. 489–518. Springer, 2017. 5, 6
34. Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. *Des. Codes Cryptography*, 80(1):29–61, 2016. 2, 10
35. Charanjit S. Jutla and Arnab Roy. Improved structure preserving signatures under standard bilinear assumptions. Cryptology ePrint Archive, Report 2017/025, 2017. 2, 3
36. Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. *ACM CCS 2003*, pp. 155–164. ACM Press, 2003. 3
37. Eike Kiltz, Jiaxin Pan, and Hoeteck Wee. Structure-preserving signatures from standard assumptions, revisited. *CRYPTO 2015, Part II*, *LNCS* 9216, pp. 275–295. Springer, 2015. 2, 3, 8

38. Taechan Kim and Razvan Barbulescu. Extended tower number field sieve: A new complexity for the medium prime case. *CRYPTO 2016, Part I*, *LNCS* 9814, pp. 543–571. Springer, 2016. 5
39. Benoît Libert, Marc Joye, Moti Yung, and Thomas Peters. Concise multi-challenge CCA-secure encryption and signatures with almost tight security. *ASIACRYPT 2014, Part II*, *LNCS* 8874, pp. 1–21. Springer, 2014. 3
40. Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. *ASIACRYPT 2015, Part I*, *LNCS* 9452, pp. 681–707. Springer, 2015. 6
41. Benoît Libert, Thomas Peters, and Moti Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. *CRYPTO 2015, Part II*, *LNCS* 9216, pp. 296–316. Springer, 2015. 2, 4
42. Sven Schäge. Tight proofs for signature schemes without random oracles. *EUROCRYPT 2011*, *LNCS* 6632, pp. 189–206. Springer, 2011. 3
43. Michael Scott. On the efficient implementation of pairing-based protocols. *IMACC 2011*, *LNCS* 7089, pp. 296–308. Springer, 2011. 4
44. Rajeev Verma. *Efficient Implementations of Pairing-Based Cryptography on Embedded Systems*. PhD thesis, Rochester Institute of Technology, New York, USA, 2015. 4