Linearly Homomorphic Structure-Preserving Signatures and Their Applications

Benoît Libert[1], Thomas Peters[2⋆], Marc Joye[1], and Moti Yung[3]

[1] Technicolor (France)
[2] Université catholique de Louvain, Crypto Group (Belgium)
[3] Google Inc. and Columbia University (USA)

Abstract. Structure-preserving signatures (SPS) are signature schemes where messages, signatures and public keys all consist of elements of a group over which a bilinear map is efficiently computable. This property makes them useful in cryptographic protocols as they nicely compose with other algebraic tools (like the celebrated Groth-Sahai proof systems). In this paper, we consider SPS systems with homomorphic properties and suggest applications that have not been provided before (in particular, not by employing ordinary SPS). We build linearly homomorphic structure-preserving signatures under simple assumptions and show that the primitive makes it possible to verify the calculations performed by a server on outsourced encrypted data (*i.e.*, combining secure computation and authenticated computation to allow reliable and secure cloud storage and computation, while freeing the client from retaining cleartext storage). Then, we give a generic construction of non-malleable (and actually simulation-sound) commitment from any linearly homomorphic SPS. This notably provides the first constant-size non-malleable commitment to group elements.

Keywords: Structure-preserving cryptography, signatures, homomorphism, commitment schemes, non-malleability.

1 Introduction

Composability is an important cryptographic design notion for building systems and protocols. Inside protocols, cryptographic tools need to compose well with each other in order to be used in combination. Structure-preserving cryptography [3], in turn, is a recent paradigm that takes care of composing algebraic tools, and primarily within groups supporting bilinear maps to allow smooth composition with the Groth-Sahai proof systems [41]. The notion allows for modular and simplified designs of various cryptographic protocols and primitives. In the last three years, a large body of work has analyzed the feasibility and the efficiency of structure-preserving signatures (SPS) [40, 25, 34, 1, 3, 4, 17, 26, 44, 5, 6], public-key encryption [18] and commitments schemes [42, 2].

⋆ This author was supported by the CAMUS Walloon Region Project.

In this paper, we consider SPS schemes with linearly homomorphic properties and argue that such primitives have many applications, even independently of Groth-Sahai proofs. Let us next review our results and then review related work.

1.1 Our Contributions

Linearly Homomorphic Structure-Preserving Signatures. In this paper, we put forth the notion of linearly homomorphic structure-preserving signatures (homomorphic signatures and structure-preserving signatures have been defined before, as we review in the sequel, but the combination of the earlier notions is useful and non-trivial). These signature schemes function exactly like ordinary homomorphic signatures with the additional restriction that signatures and messages only consist of (vectors of) group elements whose discrete logarithms may not be available. We describe three constructions and prove their security under established complexity assumptions in symmetric bilinear groups.

Applications. As in all SPS systems, the structure-preserving property makes it possible to efficiently prove knowledge of a homomorphic signature on a committed vector. However, as indicated above, we describe applications of linearly homomorphic SPS beyond their compatibility with the Groth-Sahai techniques.

First, we show that the primitive enables verifiable computation mechanisms on encrypted data.[4] Specifically, it allows a client to store encrypted files on an untrusted remote server. While the dataset is encrypted using an additively homomorphic encryption scheme, the server is able to blindly compute linear functions on the original data and provide the client with a short homomorphically derived signature vouching for the correctness of the computation. This is achieved by having the client sign each ciphertext using a homomorphic SPS scheme and handing the resulting signatures to the server at the beginning. After this initial phase, the client only needs to store a short piece of information, no matter how large the file is. Still, he remains able to authenticate linear functions on his data and the whole process is fully non-interactive. The method extends when datasets are encrypted using a CCA1-secure encryption schemes. Indeed, we will observe that linearly homomorphic SPS schemes yield simple homomorphic IND-CCA1-secure cryptosystems with publicly verifiable ciphertexts.

As a second and perhaps more surprising application, we show that linearly homomorphic SPS schemes generically yield non-malleable [31] trapdoor commitments to group elements. We actually construct a simulation-sound trapdoor commitment [35] — a primitive known (by [35, 47]) to imply re-usable non-malleable commitments with respect to opening [28] — from any linearly homomorphic SPS satisfying a relatively mild condition. To our knowledge, we thus obtain the first constant-size trapdoor commitments to group elements providing re-usable non-malleability with respect to opening. Previous non-interactive

[4] Our goals are very different from those of [37], where verifiable computation on homomorphically encrypted data is also considered. We do not seek to outsource computation but rather save the client from storing large datasets.

commitments to group elements were either malleable [41, 42] or inherently length-increasing [32]: if we disregard the trivial solution consisting of hashing the message first (which is not an option when we want to allow for efficient proofs of knowledge of an opening), no general technique has been known, to date, for committing to many group elements at once using a short commitment.

In the structure-preserving case, our transformation is purely generic as it applies to a template which any linearly homomorphic SPS necessarily satisfies in symmetric bilinear groups. We also generalize the construction so as to build simulation-sound trapdoor commitments to vectors from any pairing-based (non-structure-preserving) linearly homomorphic signature. In this case, the conversion is only semi-generic as it imposes conditions which are only met by pairing-based systems for the time being: essentially, we need the underlying signature scheme to operate over groups of finite, public order. While only partially generic, this construction of non-malleable commitments from linearly homomorphic signatures is somewhat unexpected considering that the terms "non-malleability" and "homomorphism" are antagonistic, and may be considered incompatible.

Techniques and ideas. At first, the very name of our primitive may sound almost self-contradictory when it comes to formally define its security. Indeed, the security of a linearly homomorphic scheme [14] notably requires that it be infeasible to publicly compute a signature on a vector outside the linear span of originally signed vectors. The problem is that, when vector entries live in a discrete-logarithm hard group, deciding whether several vectors are independent or not is believed to be a hard problem. Yet, this will not prevent us from applying new techniques and constructing schemes with security proofs under simple assumptions and the reduction will be able to detect when the adversary has won by simply solving the problem instance it received as input.

Our first scheme's starting point is the one-time (regular) SPS scheme of Abe *et al.* [1]. By removing certain public key components, we obtain the desired linear homomorphism, and prove the security using information-theoretic arguments as in [1]. The key observation here is that, as long as the adversary does not output a signature on a linear combination of previously signed vectors, it will be unable to sign its target vector in the same way as the reduction would, because certain private key components will remain perfectly hidden.

Our initial scheme inherits the one-time restriction of the scheme in [1] in that only one linear subspace can be safely signed with a given public key. Nevertheless, we can extend it to build a full linearly homomorphic SPS system. To this end, we suitably combine our first scheme with Waters signatures [51]. Here, Waters signatures are used as a resting ground for fresh random exponents which are introduced in each signed vector and help us refresh the state of the system and apply each time the same argument as in the one-time scheme. We also present techniques to turn the scheme into a fully randomizable one, where a derived signature has the same distribution as a directly signed message.

In our simulation-sound commitments to group elements, the commitment generation technique appeals to the verification algorithm of the signature scheme, and proceeds by evaluating the corresponding pairing-product equations on the

message, but using random group elements instead of actual signatures. The binding and simulation-binding properties, in turn, stem from the infeasibility of forging signatures while the signature homomorphism allows equivocating fake commitments when simulating the view of an adversary. It was already known how to build simulation-sound and non-malleable commitments [35, 47, 28, 36, 21] from signature schemes with efficient Σ protocols. Our method is, in fact, different and immediately yields length-reducing structure-preserving commitments to vectors without using Σ protocols.

1.2 Related Work

Structure-Preserving Signatures. Signature schemes where messages only consist of group elements appeared for the first time — without the "structure-preserving" terminology — as ingredients of Groth's construction [40] of group signatures in the standard model. The scheme of [40] was mostly a proof of concept, with signatures consisting of thousands of group elements. More efficient realizations were given by Cathalo, Libert and Yung [25] and Fuchsbauer [34]. Abe, Haralambiev and Ohkubo [1, 3] subsequently showed how to sign messages of n group elements at once using $O(1)$-size signatures. Lower bounds on the size of structure-preserving signatures were given in [4] while Abe *et al.* [7] provided evidence that optimally short SPS necessarily rely on interactive assumptions. As an ingredient for their tightly secure cryptosystems, Hofheinz and Jager [44] gave constructions based on the Decision Linear assumption [13] while similar results were independently achieved in [17, 26]. Quite recently, Abe *et al.* [5, 6] obtained constant-size signatures without sacrificing the security guarantees offered by security proofs under simple assumptions.

Regarding primitives beyond signature schemes, Camenisch *et al.* [18] showed a structure-preserving variant of the Cramer-Shoup cryptosystem [27] and used it to implement oblivious third parties [19]. Groth [42] described length-reducing trapdoor commitments (*i.e.*, where the commitment is shorter than the committed message) to group elements whereas [2] showed the impossibility of realizing such commitments when the commitment string lives in the same group as the message. Sakai *et al.* [49] recently suggested to use structure-preserving identity-based encryption [50] systems to restrict the power of the opening authority in group signatures.

Linearly Homomorphic Signatures. The concept of homomorphic signatures can be traced back to Desmedt [30] while proper definitions remained lacking until the work of Johnson *et al.* [46]. Since then, constructions have appeared for various kinds of homomorphisms (see [8] and references therein).

Linearly homomorphic signatures are an important class of homomorphic signatures for arithmetic functions, whose study was initiated by Boneh, Freeman, Katz and Waters [14]. While initially motivated by applications to network coding [14], they are also useful in proofs of storage [9] or in verifiable computation mechanisms, when it comes to authenticate servers' computations on outsourced data (see, *e.g.*, [8]). The recent years, much attention was given to the notion

and a variety of constructions [38, 10, 15, 16, 23, 24, 33, 11, 12] based on various assumptions have been studied.

1.3 Organization

Section 2 first gives security definitions for linearly homomorphic SPS systems, for which efficient constructions are provided in Section 3. Their applications to verifiable computation on encrypted data are explained in Section 4 while Section 5 shows how to build simulation-sound commitments to group elements.

2 Background

2.1 Definitions for Linearly Homomorphic Signatures

Let $(\mathbb{G}, \mathbb{G}_T)$ be a configuration of (multiplicatively written) groups of prime order p over which a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is efficiently computable.

Following [1, 3], we say that a signature scheme is *structure-preserving* if messages, signature components and public keys live in the group $\mathbb{G}$.

We consider linearly homomorphic signatures for which the message space $\mathcal{M}$ consists of pairs $\mathcal{M} := \mathcal{T} \times \mathbb{G}^n$, for some $n \in \mathbb{N}$, where $\mathcal{T}$ is a tag space. We remark that, in the applications considered in this paper, tags do not need to be group elements. We thus allow them to be arbitrary strings.

Definition 1. *A linearly homomorphic structure-preserving signature scheme over $(\mathbb{G}, \mathbb{G}_T)$ is a set of efficient algorithms $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ for which the message space is $\mathcal{M} := \mathcal{T} \times \mathbb{G}^n$, for some $n \in \mathsf{poly}(\lambda)$ and some set $\mathcal{T}$, and with the following specifications.*

$\mathsf{Keygen}(\lambda, n)$: *is a randomized algorithm that takes in a security parameter $\lambda \in \mathbb{N}$ and an integer $n \in \mathsf{poly}(\lambda)$ denoting the dimension of vectors to be signed. It outputs a key pair $(\mathsf{pk}, \mathsf{sk})$ and the description of a tag (i.e., a file identifier) space $\mathcal{T}$.*

$\mathsf{Sign}(\mathsf{sk}, \tau, \vec{M})$: *is a possibly probabilistic algorithm that takes as input a private key sk, a file identifier $\tau \in \mathcal{T}$ and a vector $\vec{M} \in \mathbb{G}^n$. It outputs a signature $\sigma \in \mathbb{G}^{n_s}$, for some $n_s \in \mathsf{poly}(\lambda)$.*

$\mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell})$: *is a (possibly probabilistic) signature derivation algorithm. It takes as input a public key pk, a file identifier τ as well as ℓ pairs $(\omega_i, \sigma^{(i)})$, each of which consists of a weight $\omega_i \in \mathbb{Z}_p$ and a signature $\sigma^{(i)} \in \mathbb{G}^{n_s}$. The output is a signature $\sigma \in \mathbb{G}^{n_s}$ on the vector $\vec{M} = \prod_{i=1}^{\ell} \vec{M}_i{}^{\omega_i}$, where $\sigma^{(i)}$ is a signature on $\vec{M}_i$.*

$\mathsf{Verify}(\mathsf{pk}, \tau, \vec{M}, \sigma)$: *is a deterministic algorithm that takes in a public key pk, a file identifier $\tau \in \mathcal{T}$, a signature σ and a vector $\vec{M}$. It outputs 1 if σ is deemed valid and 0 otherwise.*

Correctness is expressed by imposing that, for all $\lambda \in \mathbb{N}$, all integers $n \in \mathsf{poly}(\lambda)$ and all triples $(\mathsf{pk}, \mathsf{sk}, \mathcal{T}) \leftarrow \mathsf{Keygen}(\lambda, n)$, the following holds:

1. For all $\tau \in \mathcal{T}$ and all n-vectors $\vec{M}$, if $\sigma = \mathsf{Sign}(\mathsf{sk}, \tau, \vec{M})$, then we have $\mathsf{Verify}(\mathsf{pk}, \tau, \vec{M}, \sigma) = 1$.
2. For all $\tau \in \mathcal{T}$, $\ell > 0$ and $\{(\omega_i, \sigma^{(i)}, \vec{M}_i)\}_{i=1}^{\ell}$, if $\mathsf{Verify}(\mathsf{pk}, \tau, \vec{M}_i, \sigma^{(i)}) = 1$ for each i, then $\mathsf{Verify}\big(\mathsf{pk}, \tau, \prod_{i=1}^{\ell} \vec{M}_i{}^{\omega_i}, \mathsf{SignDerive}(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell})\big) = 1$.

Security. In linearly homomorphic signatures, we use the same definition of unforgeability as in [11]. This definition implies security in the stronger model used by Freeman [33] since the adversary can interleave signing queries for individual vectors belonging to distinct subspaces. Moreover, file identifiers can be chosen by the adversary (which strengthens the definition of [14]) and are not assumed to be uniformly distributed. As a result, a file identifier can be a low-entropy, easy-to-remember string such as the name of the dataset's owner.

Definition 2. *A linearly homomorphic SPS scheme $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ is secure if no PPT adversary has non-negligible advantage in the game below:*

1. *The adversary $\mathcal{A}$ chooses an integer $n \in \mathbb{N}$ and sends it to the challenger who runs $\mathsf{Keygen}(\lambda, n)$ and obtains $(\mathsf{pk}, \mathsf{sk})$ before sending pk to $\mathcal{A}$.*
2. *On polynomially-many occasions, $\mathcal{A}$ can interleave the following kinds of queries.*
   - *Signing queries: $\mathcal{A}$ chooses a tag $\tau \in \mathcal{T}$ and a vector $\vec{M} \in \mathbb{G}^n$. The challenger picks a handle h and computes $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \tau, \vec{M})$. It stores $(\mathsf{h}, (\tau, \vec{M}, \sigma))$ in a table T and returns h.*
   - *Derivation queries: $\mathcal{A}$ chooses a vector of handles $\vec{\mathsf{h}} = (\mathsf{h}_1, \ldots, \mathsf{h}_k)$ and a set of coefficients $\{\omega_i\}_{i=1}^{k}$. The challenger retrieves $\{(\mathsf{h}_i, (\tau, \vec{M}_i), \sigma^{(i)})\}_{i=1}^{k}$ from T and returns $\perp$ if one of these does not exist or if there exists $i \in \{1, \ldots, k\}$ such that $\tau_i \neq \tau$. Otherwise, it computes $\vec{M} = \prod_{i=1}^{k} \vec{M}_i^{\omega_i}$ and runs $\sigma' \leftarrow \mathsf{SignDerive}\big(\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{k}\big)$. It also chooses a handle h', stores $(\mathsf{h}', (\tau, \vec{M}), \sigma')$ in T and returns h' to $\mathcal{A}$.*
   - *Reveal queries: $\mathcal{A}$ chooses a handle h. If no tuple of the form $(\mathsf{h}, (\tau, \vec{M}), \sigma')$ exists in T, the challenger returns $\perp$. Otherwise, it returns σ' to $\mathcal{A}$ and adds $((\tau, \vec{M}), \sigma')$ to the set Q.*
3. *$\mathcal{A}$ outputs an identifier $\tau^\star$, a signature $\sigma^\star$ and a vector $\vec{M}^\star \in \mathbb{G}^n$. The adversary $\mathcal{A}$ wins if $\mathsf{Verify}(\mathsf{pk}, \tau^\star, \vec{M}^\star, \sigma^\star) = 1$ and one of the conditions below is satisfied:*
   - *(Type I): $\tau^\star \neq \tau_i$ for any entry $(\vec{\tau}_i, .)$ in Q and $\vec{M}^\star \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$.*
   - *(Type II): $\tau^\star = \tau_i$ for $k_i > 0$ entries $(\tau_i, .)$ in Q and $\vec{M}^\star \notin V_i$, where V_i denotes the subspace spanned by all vectors $\vec{M}_1, \ldots, \vec{M}_{k_i}$ for which an entry of the form $(\tau^\star, \vec{M}_j)$, with $j \in \{1, \ldots, k_i\}$, appears in Q.*

$\mathcal{A}$'s advantage is its probability of success taken over all coin tosses.

In our first scheme, we will consider a weaker notion of *one-time* security. In this notion, the adversary is limited to obtain signatures for only *one* linear subspace. In this case, there is no need for file identifiers and we assume that all vectors are assigned the identifier $\tau = \varepsilon$.

In the following, the adversary will be said *independent* if

– For any given tag τ, it is restricted to only query signatures on linearly independent vectors.
– Each vector is only queried at most once.

Non-independent adversaries are not subject to the above restrictions. It will be necessary to consider these adversaries in our construction of non-malleable commitments. Nevertheless, security against independent adversaries suffices for many applications — including encrypted cloud storage — since the signer can always append unit vectors to each newly signed vector.

At first, one may wonder how Definition 2 can be satisfied at all given that the challenger may not have an efficient way to check whether the adversary is successful. Indeed, in cryptographically useful discrete-logarithm-hard groups $\mathbb{G}$, deciding whether vectors $\{\vec{M}_i\}_i$ of $\mathbb{G}^n$ are linearly dependent is believed to be difficult when $n > 2$. However, it may be possible using some trapdoor information embedded in pk, especially if the adversary additionally outputs signatures on $\{\vec{M}_i\}_i$.

2.2 Hardness Assumptions

We rely on the following hardness assumptions, the first of which implies the second one.

Definition 3 ([13]). *In a group $\mathbb{G}$ of prime order p, the* Decision Linear Problem *(DLIN), consists in distinguishing the distributions $(g^a, g^b, g^{ac}, g^{bd}, g^{c+d})$ and $(g^a, g^b, g^{ac}, g^{bd}, g^z)$, with $a, b, c, d \xleftarrow{R} \mathbb{Z}_p^*$, $z \xleftarrow{R} \mathbb{Z}_p^*$. The* Decision Linear Assumption *is the intractability of DLIN for any PPT distinguisher $\mathcal{D}$.*

Definition 4. *The* Simultaneous Double Pairing problem *(SDP) in $(\mathbb{G}, \mathbb{G}_T)$ is, given a tuple of elements $(g_z, g_r, h_z, h_u) \in_R \mathbb{G}^4$, to find a non-trivial triple $(z, r, u) \in \mathbb{G}^3 \backslash \{(1_\mathbb{G}, 1_\mathbb{G}, 1_\mathbb{G})\}$ satisfying the equalities $e(g_z, z) \cdot e(g_r, r) = 1_{\mathbb{G}_T}$ and $e(h_z, z) \cdot e(h_u, u) = 1_{\mathbb{G}_T}$.*

3 Constructions of Linearly Homomorphic Structure-Preserving Signatures

As a warm-up, we begin by describing a one-time homomorphic signature, where a given public key allows signing only *one* linear subspace.

3.1 A One-Time Linearly Homomorphic Construction

In the description hereunder, since only one linear subspace can be signed for each public key, no file identifier τ is used. We thus set τ to be the empty string ε in all algorithms.

Keygen(λ, n): given a security parameter λ and the dimension $n \in \mathbb{N}$ of the subspace to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Then, choose generators $h, g_z, g_r, h_z \xleftarrow{R} \mathbb{G}$. Pick $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$, for $i = 1$ to n. Then, for each $i \in \{1, \ldots, n\}$, compute $g_i = g_z^{\chi_i} g_r^{\gamma_i}$, $h_i = h_z^{\chi_i} h^{\delta_i}$. The private key is $\mathsf{sk} = \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n$ while the public key is defined to be

$$\mathsf{pk} = \big(g_z,\ h_r,\ h_z,\ h,\ \{g_i, h_i\}_{i=1}^n\big) \in \mathbb{G}^{2n+4}.$$

Sign($\mathsf{sk}, \tau, (M_1, \ldots, M_n)$): to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ associated with the identifier $\tau = \varepsilon$ using $sk = \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n$, compute the signature consists of $\sigma = (z, r, u) \in \mathbb{G}^3$, where

$$z = \prod_{i=1}^n M_i^{-\chi_i}, \qquad r = \prod_{i=1}^n M_i^{-\gamma_i}, \qquad u = \prod_{i=1}^n M_i^{-\delta_i}.$$

SignDerive($\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^\ell$): given a file identifier $\tau = \varepsilon$, the public key pk and ℓ tuples $(\omega_i, \sigma^{(i)})$, parse each $\sigma^{(i)}$ as $\sigma^{(i)} = \big(z_i, r_i, u_i\big) \in \mathbb{G}^3$ for $i = 1$ to ℓ. Compute and return $\sigma = (z, r, u) = \big(\prod_{i=1}^\ell z_i^{\omega_i}, \prod_{i=1}^\ell r_i^{\omega_i}, \prod_{i=1}^\ell u_i^{\omega_i}\big)$.

Verify($\mathsf{pk}, \sigma, \tau, (M_1, \ldots, M_n)$): given a signature $\sigma = (z, r, u) \in \mathbb{G}^3$, a vector $(M_1, \ldots, M_n)$ and a file identifier $\tau = \varepsilon$, return 1 if and only if it holds that $(M_1, \ldots, M_n) \neq (1_\mathbb{G}, \ldots, 1_\mathbb{G})$ and (z, r, u) satisfy

$$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i, M_i), \qquad 1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h, u) \cdot \prod_{i=1}^n e(h_i, M_i).$$

The security proof relies on the fact that, while the signing algorithm is deterministic, signatures are not unique. However, the reduction will be able to compute exactly one signature for each vector. At the same time, an adversary has no information about which signature the legitimate signer would compute on a vector outside the span of already signed vectors. Moreover, by obtaining two distinct signatures on a given vector, the reduction can solve a given SDP instance. The following theorem is proved in the full version of the paper.

Theorem 1. *The scheme is unforgeable if the SDP assumption holds in* $(\mathbb{G}, \mathbb{G}_T)$.

3.2 A Full-Fledged Linearly Homomorphic SPS Scheme

We upgrade our one-time construction so as to sign an arbitrary number of linear subspaces. Here, each file identifier τ is a L-bit string. The construction builds on the observation that, in the scheme of Section 3.1, signatures (z, r, u) could be re-randomized by computing $(z \cdot g_r^\theta, r \cdot g_z^{-\theta}, u \cdot h_z^{-\log_h(g_r)\cdot\theta})$, with $\theta \xleftarrow{R} \mathbb{Z}_p$, if $h_z^{-\log_h(g_r)}$ were available. Since publicizing $h_z^{-\log_h(g_r)}$ would render the scheme insecure, our idea is to use Waters signatures as a support for introducing extra randomizers in the exponent.

In the construction, the u component of each signature can be seen as an aggregation of the one-time signature of Section 3.1 with a Waters signature $(h_z^{\log_h(g_r)} \cdot H_\mathbb{G}(\tau)^{-\rho}, h^\rho)$ [51] on the tag τ.

Keygen(λ, n): given a security parameter λ and the dimension $n \in \mathbb{N}$ of the subspace to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$.

1. Choose $h \xleftarrow{R} \mathbb{G}$, $\alpha_z, \alpha_r, \beta_z \xleftarrow{R} \mathbb{Z}_p$. Define $g_z = h^{\alpha_z}$, $g_r = h^{\alpha_r}$, $h_z = h^{\beta_z}$.
2. For $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ and compute $g_i = g_z^{\chi_i} g_r^{\gamma_i}$, $h_i = h_z^{\chi_i} h^{\delta_i}$.
3. Choose a random vector $\overline{\mathbf{w}} = (w_0, w_1, \ldots, w_L) \xleftarrow{R} \mathbb{G}^{L+1}$. The latter defines a hash function $H_\mathbb{G} : \{0,1\}^L \to \mathbb{G}$ which maps any L-bit string $\tau = \tau[1] \ldots \tau[L] \in \{0,1\}^L$ to $H_\mathbb{G}(\tau) = w_0 \cdot \prod_{k=1}^{L} w_k^{\tau[k]}$.

The private key is $\mathsf{sk} = \big(h_z^{\alpha_r},\ \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n\big)$ while the public key consists of

$$\mathsf{pk} = \big(g_z,\ g_r,\ h_z,\ h,\ \{g_i, h_i\}_{i=1}^n,\ \overline{\mathbf{w}}\big) \in \mathbb{G}^{2n+4} \times \mathbb{G}^{L+1}.$$

Sign($\mathsf{sk}, \tau, (M_1, \ldots, M_n)$): to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ w.r.t. the file identifier τ using $sk = \big(h_z^{\alpha_r},\ \{\chi_i, \gamma_i, \delta_i\}_{i=1}^n\big)$, choose $\theta, \rho \xleftarrow{R} \mathbb{Z}_p$ and output $\sigma = (z, r, u, v) \in \mathbb{G}^4$, where

$$z = g_r^\theta \cdot \prod_{i=1}^{n} M_i^{-\chi_i} \qquad r = g_z^{-\theta} \cdot \prod_{i=1}^{n} M_i^{-\gamma_i}$$

$$u = (h_z^{\alpha_r})^{-\theta} \cdot \prod_{i=1}^{n} M_i^{-\delta_i} \cdot H_\mathbb{G}(\tau)^{-\rho} \qquad v = h^\rho$$

SignDerive($\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell}$): given pk, a file identifier τ and ℓ tuples $(\omega_i, \sigma^{(i)})$, parse $\sigma^{(i)}$ as $\sigma^{(i)} = \big(z_i, r_i, u_i, v_i\big) \in \mathbb{G}^4$ for $i = 1$ to ℓ. Then, choose $\rho' \xleftarrow{R} \mathbb{Z}_p$ and compute and return $\sigma = (z, r, u, v)$, where $z = \prod_{i=1}^{\ell} z_i^{\omega_i}$, $r = \prod_{i=1}^{\ell} r_i^{\omega_i}$, $u = \prod_{i=1}^{\ell} u_i^{\omega_i} \cdot H_\mathbb{G}(\tau)^{-\rho'}$ and $v = \prod_{i=1}^{\ell} v_i^{\omega_i} \cdot h^{\rho'}$.

Verify($\mathsf{pk}, \sigma, \tau, (M_1, \ldots, M_n)$): given a signature $\sigma = (z, r, u, v) \in \mathbb{G}^4$, a file identifier τ and a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$, return 1 if and only if $(M_1, \ldots, M_n) \neq (1_\mathbb{G}, \ldots, 1_\mathbb{G})$ and (z, r, u, v) satisfy

$$1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^{n} e(g_i, M_i), \tag{1}$$

$$1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h, u) \cdot e(H_\mathbb{G}(\tau), v) \cdot \prod_{i=1}^{n} e(h_i, M_i).$$

The security of the scheme against *non-independent* Type I adversaries is proved under the SDP assumption. In the case of Type II forgeries, we need to assume the adversary to be independent because, at some point, the simulator is only able to compute a signature for a unique value[5] of θ.

Theorem 2. *The scheme is unforgeable against independent adversaries if the SDP assumption holds in $(\mathbb{G}, \mathbb{G}_T)$. Moreover, the scheme is secure against non-independent Type I adversaries.*

[5] Note that this is not a problem since the signer can derive θ as a pseudorandom function of τ and $(M_1, \ldots, M_n)$ to make sure that a given vector is always signed using the same θ.

The proof of Theorem 2 is available in the full verison of the paper. It uses Waters signatures as a handle to randomize signatures. Whenever the reduction is able to compute a Waters signatures $(h_z^{\alpha_r} \cdot H_{\mathbb{G}}(\tau)^{-\rho}, h^{\rho})$ on the tag τ, it can inject a fresh extra randomizer $\theta \in \mathbb{Z}_p$ in the exponent for each vector associated with τ. By doing so, with non-negligible probability, the specific vector $(\chi_1, \ldots, \chi_n)$ used by the reduction will remain undetermined from $\mathcal{A}$'s view.

Since the signature component u cannot be publicly randomized, the scheme does not have fully randomizable signatures. In the full version of the paper, we describe a fully randomizable variant. In applications like non-malleable commitments to group elements, the above scheme is sufficient however.

4 Applications

4.1 Verifiable Computation for Encrypted Cloud Storage

Linearly homomorphic schemes are known (see, *e.g.*, [8]) to provide verifiable computation mechanisms for outsourced data. Suppose that a user has a dataset consisting of n samples $s_1, \ldots, s_n \in \mathbb{Z}_p$. The dataset can be encoded as vectors $\vec{v}_i = (\vec{e}_i | s_i) \in \mathbb{Z}_p^{n+1}$, where $\vec{e}_i \in \mathbb{Z}_p^n$ denotes the i-th unit vector for each $i \in \{1, \ldots, n\}$. The user then assigns a file identifier τ to $\{\vec{v}_i\}_{i=1}^n$, computes signatures $\sigma_i \leftarrow \mathsf{Sign}(\mathsf{sk}, \tau, \vec{v}_i)$ on the resulting vectors and stores $\{(\vec{v}_i, \sigma_i)\}_{i=1}^n$ at the server. When requested, the server can then evaluate a sum $s = \sum_{i=1}^n s_i$ and provide evidence that the latter computation is correct by deriving a signature on the vector $(1, 1, \ldots, 1, s) \in \mathbb{Z}_p^{n+1}$. Unless the server is able to forge a signature for a vector outside the span of $\{\vec{v}_i\}_{i=1}^n$, it is unable to fool the user. The above method readily extends to authenticate weighted sums or Fourier transforms.

One disadvantage of the above method is that it requires the server to retain the dataset $\{s_i\}_{i=1}^n$ in the clear. Using linearly homomorphic structure-preserving signatures, the user can apply the above technique on encrypted samples using the Boneh-Boyen-Shacham (BBS) cryptosystem [13].

The BBS cryptosystem involves a public key $(g, \tilde{g}, f = g^x, h = g^y) \in_R \mathbb{G}^4$, where $(x, y) \in \mathbb{Z}_p^2$ is the private key. The user (or anyone else knowing his public key) can first encrypt his samples $\{s_i\}_{i=1}^n$ by computing BBS encryptions $(C_{1,i}, C_{2,i}, C_{3,i}) = (f^{r_i}, h^{t_i}, \tilde{g}^{s_i} \cdot g^{r_i + t_i})$, with $r_i, t_i \xleftarrow{R} \mathbb{Z}_p$, for each $i \in \{1, \ldots, n\}$. If the user holds a linearly homomorphic structure preserving signature key pair for vectors of dimension $n + 3$, he can generate n signatures on vectors $((C_{1,i}, C_{2,i}, C_{3,i}) | \vec{E}_i) \in \mathbb{G}^{n+3}$, where $\vec{E}_i = (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}}, g, 1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}}) = g^{\vec{e}_i}$ for each $i \in \{1, \ldots, n\}$. The vectors $\{((C_{1,i}, C_{2,i}, C_{3,i}) | \vec{E}_i)\}_{i=1}^n$ are then archived in the cloud with their signatures $\{(z_i, r_i, u_i, v_i)\}_{i=1}^n$ in such a way that the server can publicly derive a signature on $\left(f^{\sum_i r_i}, h^{\sum_i t_i}, \tilde{g}^{\sum_i s_i} \cdot g^{\sum_i (r_i + t_i)}, g, g, \ldots, g\right) \in \mathbb{G}^{n+3}$ in order to convince the client that the encrypted sum was correctly computed. Using his private key (x, y), the client can then retrieve the sum $\sum_i s_i$ as long as it remains in a sufficiently small range.

The interest of the above solution lies in that the client can dispense with the need for storing the $O(n)$-size public key of his linearly homomorphic signature.

Indeed, he can simply retain the random seed that was used to generate pk and re-compute private key elements $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$ whenever he wants to verify the server's response. In this case, the verification equations (1) become

$$1_{\mathbb{G}_T} = e(g_z, z \cdot \prod_{i=1}^{n} M_i^{\chi_i}) \cdot e(g_r, r \cdot \prod_{i=1}^{n} M_i^{\gamma_i})$$
$$1_{\mathbb{G}_T} = e(h_z, z \cdot \prod_{i=1}^{n} M_i^{\chi_i}) \cdot e(h, u \cdot \prod_{i=1}^{n} M_i^{\delta_i}) \cdot e(H_{\mathbb{G}}(\tau), v),$$

so that the client only has to compute $O(1)$ pairings. Moreover, the client does not have to determine an upper bound on the size of his dataset when generating his public key. Initially, he only needs to generate $\{(g_j, h_j)\}_{j=1}^3$. When the i-th ciphertext $(C_{1,i}, C_{2,i}, C_{3,i})$ has to be stored, the client derives $(\chi_{i+3}, \gamma_{i+3}, \delta_{i+3})$ and (g_{i+3}, h_{i+3}) by applying a PRF to the index i. This will be sufficient to sign vectors of the form $((C_{1,i}, C_{2,i}, C_{3,i})|\vec{E}_i)$.

In order to hide all partial information about the original dataset, the server may want to re-randomize the derived signature and ciphertext before returning them. This can be achieved by having the client include signatures on the vectors $(f, 1_{\mathbb{G}}, g, 1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$, $(1_{\mathbb{G}}, h, g, 1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ in the outsourced dataset. Note that, in this case, the signature should be re-randomized as well. For this reason, our randomizable scheme described in the full version of the paper should be preferred.

Complete security models for "verifiable computation on encrypted data" are beyond the scope of this paper. Here, they would naturally combine the properties of secure homomorphic encryption and authenticated computing. It should be intuitively clear that a malicious server cannot trick a client into accepting an incorrect result (*i.e.*, one which differs from the actual defined linear function it is supposed to compute over the defined signed ciphertext inputs) without defeating the security of the underlying homomorphic signature.

4.2 Extension to CCA1-Encrypted Data

In the application of Section 4.1, the underlying crypotosystem has to be additively homomorphic, which prevents it from being secure against adaptive chosen-ciphertext attacks. On the other hand, the method is compatible with security against *non-adaptive* chosen ciphertext attacks. One possibility is to apply the "lite" Cramer-Shoup technique (in its variant based on DLIN) as it achieves CCA1-security while remaining homomorphic. Unfortunately, the validity of ciphertexts is not publicly verifiable, which may be annoying in applications like cloud storage or universally verifiable e-voting systems. Indeed, servers may be willing to have guarantees that they are actually storing encryptions of some message instead of random group elements.

Consider the system where $(C_1, C_2, C_3, C_4) = (f^r, h^t, g^{r+t}, \tilde{g}^m \cdot X_1^r \cdot X_2^t)$ is decrypted as $m = \log_{\tilde{g}}(C_4 \cdot C_1^{-x_1} C_2^{-x_2} C_3^{-z})$, where $X_1 = f^{x_1} g^z$ and $X_2 =$

$h^{x_2}g^z$ are part of the public key. In [45], such a system was made chosen-ciphertext secure using a *publicly* verifiable one-time simulation-sound proof that (f, h, g, C_1, C_2, C_3) forms a DLIN tuple. In the security proof, if the reduction is guaranteed not to leak $C_1^{-x_1}C_2^{-x_2}C_3^{-z}$ for an invalid triple (C_1, C_2, C_3) (*i.e.*, as long as the adversary is unable to generate a fake proof for this), the private key component z will remain perfectly hidden. Consequently, if the challenge ciphertext is computed by choosing $C_3^\star \in_R \mathbb{G}$ (so that $(f, h, g, C_1^\star, C_2^\star, C_3^\star)$ is not a DLIN tuple) and computing $C_4^\star = \tilde{g}^m \cdot C_1^{\star x_1} \cdot C_2^{\star x_2} \cdot C_3^{\star z}$, the plaintext m is independent of $\mathcal{A}$'s view. If we replace the one-time simulation-sound proofs by standard proofs of membership in the scheme of [45], we obtain a CCA1 homomorphic encryption scheme. Linearly homomorphic SPS schemes provide a simple and efficient way to do that.

The idea is to include in the public key the verification key of a one-time linearly homomorphic SPS — using the scheme of Section 3.1 — for $n = 3$ as well as signatures on the vectors $(f, 1_\mathbb{G}, g)$, $(1_\mathbb{G}, h, g) \in \mathbb{G}^3$. This will allow the sender to publicly derive a signature (z, r, u) on the vector $(C_1, C_2, C_3) = (f^r, h^t, g^{r+t})$. Each ciphertext thus consists of $(z, r, u, C_1, C_2, C_3, C_4)$. In the security proof, at each pre-challenge decryption query, the signature (z, r, u) serves as publicly verifiable evidence that (f, h, g, C_1, C_2, C_3) is a DLIN tuple. In the challenge phase, the reduction reveals another homomorphic signature $(z^\star, r^\star, u^\star)$ for a vector $(C_1^\star, C_2^\star, C_3^\star)$ that may be outside the span of $(f, 1_\mathbb{G}, g)$ and $(1_\mathbb{G}, h, g)$ but it does not matter since decryption queries are not allowed beyond this point.

We note that linearly homomorphic SPS can also be used to construct CCA1-secure homomorphic encryption schemes based on the Naor-Yung paradigm [48].

5 Non-Malleable Trapdoor Commitments to Group Elements from Linearly Homomorphic Structure-Preserving Signatures

As noted in [42, 43], some applications require to commit to group elements without knowing their discrete logarithms or destroying their algebraic structure by hashing them first. This section shows that, under a certain mild condition, linearly homomorphic SPS imply length-reducing non-malleable structure-preserving commitments to vectors of group elements.

As a result, we obtain the first length-reducing non-malleable structure-preserving trapdoor commitment. Our scheme is not *strictly*[6] structure-preserving (according to the terminology of [2]) because the commitment string lives in $\mathbb{G}_T$ rather than $\mathbb{G}$. Still, openings only consist of elements in $\mathbb{G}$, which makes it possible to generate efficient NIWI proofs that committed group elements satisfy certain properties. To our knowledge, the only known non-malleable commitment schemes whose openings only consist of group elements were described by

[6] We recall that strictly structure-preserving commitments cannot be length-reducing, as shown by Abe *et al.* [2], so that our scheme is essentially the best we can hope for if we aim at short commitment stings.

Fischlin *et al.* [32]. However, these constructions cannot be length-reducing as they achieve universal composability [20, 22].

Our schemes are obtained by first constructing simulation-sound trapdoor commitments (SSTC) [35, 47] to group elements. SSTC schemes were first suggested by Garay, MacKenzie and Yang [35] as a tool for constructing universally composable zero-knowledge proofs [20]. MacKenzie and Yang subsequently gave a simplified security definition which suffices to provide non-malleability with respect to opening in the sense of the definition of re-usable non-malleable commitments [28].

In a SSTC, each commitment is labeled with a tag. The definition of [47] requires that, even if the adversary can see equivocations of commitments to possibly distinct messages for several tags $tag_1, \ldots, tag_q$, it will not be able to break the binding property for a new tag $tag \notin \{tag_1, \ldots, tag_q\}$.

Definition 5 ([47]). *A simulation-sound trapdoor commitment is a tuple of algorithms* $(\mathsf{Setup}, \mathsf{Com}, \mathsf{FakeCom}, \mathsf{FakeOpen}, \mathsf{Verify})$ *where* $(\mathsf{Setup}, \mathsf{Com}, \mathsf{Verify})$ *forms a commitment scheme and* $(\mathsf{FakeCom}, \mathsf{FakeOpen})$ *are PPT algorithms with the following properties*

Trapdoor: *for any tag and any message* Msg*, the following distributions are computationally indistinguishable:*

$$D_{fake} := \{(pk, tk) \leftarrow \mathsf{Setup}(\lambda);\ (\widetilde{\mathsf{com}}, \mathsf{aux}) \leftarrow \mathsf{FakeCom}(pk, tk, tag);\ \widetilde{\mathsf{dec}} \leftarrow \mathsf{FakeOpen}(\mathsf{aux}, tk, \widetilde{\mathsf{com}}, \mathsf{Msg}) : (pk, tag, \mathsf{Msg}, \widetilde{\mathsf{com}}, \widetilde{\mathsf{dec}})\}$$

$$D_{real} := \{(pk, tk) \leftarrow \mathsf{Setup}(\lambda);\ (\mathsf{com}, \mathsf{dec}) \leftarrow \mathsf{Com}(pk, tag, \mathsf{Msg}) : (pk, tag, \mathsf{Msg}, \mathsf{com}, \mathsf{dec})\}$$

Simulation-sound binding: *for any PPT adversary* $\mathcal{A}$*, the following probability is negligible*

$$\Pr[\ (pk, tk) \leftarrow \mathsf{Setup}(\lambda);\ (\mathsf{com}, tag, \mathsf{Msg}_1, \mathsf{Msg}_2, \mathsf{dec}_1, \mathsf{dec}_2) \leftarrow \mathcal{A}^{\mathcal{O}_{tk,pk}}(pk) : \mathsf{Msg}_1 \neq \mathsf{Msg}_2 \wedge \mathsf{Verify}(pk, tag, \mathsf{Msg}_1, \mathsf{com}, \mathsf{dec}_1) = 1 \wedge \mathsf{Verify}(pk, tag, \mathsf{Msg}_2, \mathsf{com}, \mathsf{dec}_2) = 1 \wedge tag \notin Q],$$

where $\mathcal{O}_{tk,pk}$ *is an oracle that maintains an initially empty set* Q *and operates as follows:*

- *On input* (commit, tag)*, it runs* $(\widetilde{\mathsf{com}}, \mathsf{aux}) \leftarrow \mathsf{FakeCom}(pk, tk, tag)$*, stores* $(\widetilde{\mathsf{com}}, tag, \mathsf{aux})$*, returns* $\widetilde{\mathsf{com}}$ *and adds tag in* Q*.*
- *On input* $(\mathsf{decommit}, \widetilde{\mathsf{com}}, \mathsf{Msg})$*: if a tuple* $(\widetilde{\mathsf{com}}, tag, \mathsf{aux})$ *was previously stored, it computes* $\widetilde{\mathsf{dec}} \leftarrow \mathsf{FakeOpen}(\mathsf{aux}, tk, tag, \widetilde{\mathsf{com}}, \mathsf{Msg})$ *and returns* $\widetilde{\mathsf{dec}}$*. Otherwise,* $\mathcal{O}_{tk,pk}$ *returns* $\perp$*.*

While our SSTC to group elements will be proved secure in the above sense, a *non-adaptive* flavor of simulation-sound binding security is sufficient for constructing non-malleable commitments. Indeed, Gennaro used [36] such a relaxed

notion to achieve non-malleability from similar-looking multi-trapdoor commitments. In the non-adaptive notion, the adversary has to choose the set of tags $tag_1, \ldots, tag_\ell$ for which it wants to query $\mathcal{O}_{tk,pk}$ before seeing the public key pk.

5.1 Template of Linearly Homomorphic SPS Scheme

We first remark that *any* constant-size linearly homomorphic structure-preserving signature necessarily complies with the template below.

For simplicity, the template is described in terms of symmetric pairings but generalizations to asymmetric configurations are possible.

Keygen(λ, n): given λ and the dimension $n \in \mathbb{N}$ of the vectors to be signed, choose constants n_z, n_v, m. Among these, n_z and n_v will determine the signature length while m will be the number of verification equations. Then, choose $\{F_{j,\mu}\}_{j\in\{1,\ldots,m\},\mu\in\{1,\ldots,n_z\}}$, $\{G_{j,i}\}_{i\in\{1,\ldots,n\},\ j\in\{j,\ldots,m\}}$ in the group $\mathbb{G}$. The public key is $\mathsf{pk} = \big(\{F_{j,\mu}\}_{j\in\{1,\ldots,m\},\mu\in\{1,\ldots,n_z\}}, \{G_{j,i}\}_{i\in\{1,\ldots,n\},\ j\in\{j,\ldots,m\}}\big)$ while sk contains information about the representation of public elements w.r.t. specific bases.

Sign($\mathsf{sk}, \tau, (M_1, \ldots, M_n)$): Outputs $\sigma = \big(Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v}\big) \in \mathbb{G}^{n_z+n_v}$.

SignDerive($\mathsf{pk}, \tau, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell}$): parses $\sigma^{(i)}$ as $\big(Z_1^{(i)}, \ldots, Z_{n_z}^{(i)}, V_1^{(i)}, \ldots, V_{n_v}^{(i)}\big)$ for each $i \in \{1, \ldots, \ell\}$ and computes

$$Z_\mu = \prod_{i=1}^{\ell} Z_\mu^{(i)\ \omega_i} \qquad V_\nu = \prod_{i=1}^{\ell} V_\nu^{(i)\ \omega_i} \quad \mu \in \{1, \ldots, n_z\},\ \nu \in \{1, \ldots, n_v\}.$$

After possible extra re-randomizations, it outputs $\big(Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v}\big)$.

Verify($\mathsf{pk}, \sigma, \tau, (M_1, \ldots, M_n)$): given $\sigma = \big(Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v}\big) \in \mathbb{G}^{n_z+n_v}$, a tag τ and $(M_1, \ldots, M_n)$, return 0 if $(M_1, \ldots, M_n) = (1_\mathbb{G}, \ldots, 1_\mathbb{G})$. Otherwise, do the following.

1. For each $j \in \{1, \ldots, m\}$ and $\nu \in \{1, \ldots, n_v\}$, compute one-to-one[7] encodings $T_{j,\nu} \in \mathbb{G}$ of the tag τ as a group element.
2. Return 1 if and only if $c_j = 1_{\mathbb{G}_T}$ for $j = 1$ to m, where

$$c_j = \prod_{\mu=1}^{n_z} e(F_{j,\mu}, Z_\mu) \cdot \prod_{\nu=1}^{n_v} e(T_{j,\nu}, V_\nu) \cdot \prod_{i=1}^{n} e(G_{j,i}, M_i) \quad j \in \{1, \ldots, m\}. \quad (2)$$

We say that a linearly homomorphic SPS is *regular* if, for each file identifier τ, any non-trivial vector $(M_1, \ldots, M_n) \neq (1_\mathbb{G}, \ldots, 1_\mathbb{G})$ has a valid signature.

5.2 Construction of Simulation-Sound Structure-Preserving Trapdoor Commitments

Let $\Pi^{\mathsf{SPS}} = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ be a linearly homomorphic SPS. We construct a simulation-sound trapdoor commitment as follows.

[7] This condition can be relaxed to have collision-resistant deterministic encodings. Here, we assume injectivity for simplicity.

SSTC.Setup(λ, n): given the desired dimension $n \in \mathbb{N}$ of vectors, choose public parameters pp for the linearly homomorphic SPS scheme. Then, run $\Pi^{\mathsf{SPS}}.\mathsf{Keygen}(\lambda, n)$ to obtain a public key $\mathsf{pk} = \big(\{F_{j,\mu}\}_{j\in\{1,\ldots,m\},\mu\in\{1,\ldots,n_z\}}, \{G_{j,i}\}_{i\in\{1,\ldots,n\},\ j\in\{j,\ldots,m\}}\big)$, for some constants n_z, n_v, m, and a sk. The commitment key is $pk = \mathsf{pk}$ and the trapdoor tk consists of sk. Note that the public key defines a signature space $\mathbb{G}^{n_z+n_v}$, for constants n_z and n_v.

SSTC.Com($pk, tag, (M_1, \ldots, M_n)$): to commit to $(M_1, \ldots, M_n) \in \mathbb{G}^n$ with respect to the tag $tag = \tau$, choose $\big(Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v}\big) \xleftarrow{R} \mathbb{G}^{n_z+n_v}$ in the signature space. Then, run step 1 of the verification algorithm and evaluate the right-hand-side member of (2). Namely, compute

$$c_j = \prod_{\mu=1}^{n_z} e(F_{j,\mu}, Z_\mu) \cdot \prod_{\nu=1}^{n_v} e(T_{j,\nu}, V_\nu) \cdot \prod_{i=1}^{n} e(G_{j,i}, M_i) \qquad j \in \{1, \ldots, m\} \quad (3)$$

where $\{T_{j,\nu}\}_{j,\nu}$ form an injective encoding of $tag = \tau$ as a set of group elements. The commitment string is $\mathsf{com} = (c_1, \ldots, c_m)$ whereas the decommitment is $\mathsf{dec} = \big(Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v}\big)$.

SSTC.FakeCom(pk, tk, tag): proceeds like $\mathsf{SSTC.Com}$ with randomly chosen $(\hat{M}_1, \ldots, \hat{M}_n) \xleftarrow{R} \mathbb{G}^n$. If $(\hat{\mathsf{com}}, \hat{\mathsf{dec}})$ denotes the resulting pair, the algorithm outputs $\widetilde{\mathsf{com}} = \hat{\mathsf{com}}$ and the auxiliary information aux, which consists of the pair $\mathsf{aux} = ((\hat{M}_1, \ldots, \hat{M}_n), \hat{\mathsf{dec}})$ for $tag = \tau$.

SSTC.FakeOpen($\mathsf{aux}, tk, tag, \widetilde{\mathsf{com}}, (M_1, \ldots, M_n)$): parses $\widetilde{com}$ as $(\tilde{c}_1, \ldots, \tilde{c}_m)$ and aux as $\big((\hat{M}_1, \ldots, \hat{M}_n), (\hat{Z}_1, \ldots, \hat{Z}_{n_z}, \hat{V}_1, \ldots, \hat{V}_{n_v})\big)$. The algorithm first generates a linearly homomorphic signature on $(M_1/\hat{M}_1, \ldots, M_n/\hat{M}_n)$ for the tag $tag = \tau$. Namely, using the trapdoor $tk = \mathsf{sk}$, compute a signature $\sigma' = (Z'_1, \ldots, Z'_{n_z}, V'_1, \ldots, V'_{n_v}) \leftarrow \Pi^{\mathsf{SPS}}.\mathsf{Sign}\big(\mathsf{sk}, \tau, (M_1/\hat{M}_n, \ldots, M_n/\hat{M}_n)\big)$. Since $\mathsf{aux} = \big((\hat{M}_1, \ldots, \hat{M}_n), (\hat{Z}_1, \ldots, \hat{Z}_{n_z}, \hat{V}_1, \ldots, \hat{V}_{n_v})\big)$ satisfies

$$\tilde{c}_j = \prod_{\mu=1}^{n_z} e(F_{j,\mu}, \hat{Z}_\mu) \cdot \prod_{\nu=1}^{n_v} e(T_{j,\nu}, \hat{V}_\nu) \cdot \prod_{i=1}^{n} e(G_{j,i}, \hat{M}_i) \qquad j \in \{1, \ldots, m\}, \quad (4)$$

$\mathsf{FakeOpen}$ runs $(\tilde{Z}_1, \ldots, \tilde{Z}_{n_z}, \tilde{V}_1, \ldots, \tilde{V}_{n_v}) \leftarrow \mathsf{SignDerive}(\mathsf{pk}, \tau, \{(1, \sigma'), (1, \hat{\sigma})\})$, where $\hat{\sigma} = (\hat{Z}_1, \ldots, \hat{Z}_{n_z}, \hat{V}_1, \ldots, \hat{V}_{n_v})$. It outputs a valid de-commitment $\widetilde{\mathsf{dec}} = (\tilde{Z}_1, \ldots, \tilde{Z}_{n_z}, \tilde{V}_1, \ldots, \tilde{V}_{n_v})$ to $(M_1, \ldots, M_n)$ with respect to $tag = \tau$.

SSTC.Verify($pk, tag, (M_1, \ldots, M_n), \mathsf{com}, \mathsf{dec}$): parse com as $(c_1, \ldots, c_m) \in \mathbb{G}_T^m$ and dec as $\big(Z_1, \ldots, Z_{n_z}, V_1, \ldots, V_{n_v}\big) \in \mathbb{G}^{n_z+n_v}$ (if these values do not parse properly, return 0). Then, compute a one-to-one encoding $\{T_{j,\nu}\}_{j,\nu}$ of $tag = \tau$. Return 1 if relations (3) hold and 0 otherwise.

In the full version of the paper, we extend this construction so as to build simulation-sound trapdoor commitment to vectors from any linearly homomorphic signature that fits a certain template. As a result, we obtain a modular construction of constant-size non-malleable commitment to vectors which preserves the feasibility of efficiently proving properties about committed values.

Theorem 3. *Assuming that the underlying linearly homomorphic SPS is regular and secure against non-independent Type I adversaries, the above construction is a simulation-sound trapdoor commitment to group elements.* (The proof is given in the full version of the paper.

A standard technique (see [35, 36]) to construct a re-usable non-malleable commitment from a SSTC scheme is as follows. To commit to Msg, the sender generates a key-pair $(\mathsf{VK}, \mathsf{SK})$ for a one-time signature and generates $(\mathsf{com}, \mathsf{dec}) \leftarrow \mathsf{SSTC.Commit}(pk, \mathsf{VK}, \mathsf{MSg})$ using VK as a tag. The non-malleable commitment string is the pair $(\mathsf{com}, \mathsf{VK})$ and the opening is given by (dec, σ), where σ is a one-time signature on com, so that the receiver additionally checks the validity of σ. This construction is known to provide input independence [29] and thus non-malleability with respect to opening, as proved in [29, 39].

In our setting, we cannot compute σ as a signature of com, as it consists of $\mathbb{G}_T$ elements. However, we can sign the pair $(\mathsf{Msg}, \mathsf{dec})$ — whose components live in $\mathbb{G}$ — as long as it uniquely determines com. To this end, we can use the one-time structure-preserving of [1, Appendix C.1] as it allows signing messages of arbitrary length using a constant-size one-time public key. Like our scheme of Section 3.2, it relies on the SDP assumption and yields a non-malleable commitment based on this sole assumption. Alternatively, we can move σ in the commitment string (which becomes $(\mathsf{com}, \mathsf{VK}, \sigma)$), in which case the one-time signature does not need to be structure-preserving but it has to be strongly unforgeable (as can be observed from the definition of independent commitments [29]) while the standard notion of unforgeability suffices in the former case.

References

1. M. Abe, K. Haralambiev, M. Ohkubo. Signing on Elements in Bilinear Groups for Modular Protocol Design. Cryptology ePrint Archive: Report 2010/133, 2010.
2. M. Abe, K. Haralambiev, M. Ohkubo. Group to Group Commitments Do Not Shrink. In *Eurocrypt'12*, *LNCS* 7237, pp. 301–317, 2012.
3. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo. Structure-Preserving Signatures and Commitments to Group Elements. In *Crypto'10*, *LNCS* 6223, pp. 209–236, 2010.
4. M. Abe, J. Groth, K. Haralambiev, M. Ohkubo. Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups. In *Crypto'11*, *LNCS* 6841, pp. 649–666, 2011.
5. M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, M. Ohkubo. Constant-Size Structure-Preserving Signatures: Generic Constructions and Simple Assumptions. In *Asiacrypt'12*, *LNCS* 7658, pp. 4–24, 2012.
6. M. Abe, B. David, M. Kohlweiss, R. Nishimaki, M. Ohkubo. Tagged One-Time Signatures: Tight Security and Optimal Tag Size. In *PKC'13*, *LNCS* 7778, pp. 312–331, 2013.
7. M. Abe, J. Groth, M. Ohkubo. Separating Short Structure-Preserving Signatures from Non-interactive Assumptions. In *Asiacrypt'11*, *LNCS* 7073, pp. 628–646, 2011.
8. J.-H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, a. shelat, B. Waters. Computing on Authenticated Data. In *TCC 2012*, *LNCS* 7194, pp. 1–20, 2012.

9. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song. Provable data possession at untrusted stores. In *ACM-CCS 2007*, pp. 598–609, 2007.
10. N. Attrapadung, B. Libert. Homomorphic Network Coding Signatures in the Standard Model. In *PKC'11*, *LNCS* 6571, pp. 17–34, 2011.
11. N. Attrapadung, B. Libert, T. Peters. Computing on Authenticated Data: New Privacy Definitions and Constructions. In *Asiacrypt'12*, *LNCS* 7658, pp. 367–385, 2012.
12. N. Attrapadung, B. Libert, T. Peters. Efficient Completely Context-Hiding Quotable Signatures and Linearly Homomorphic Signatures. In *PKC'13*, *LNCS* 7778, pp. 367–385, pp. 386–404, 2013.
13. D. Boneh, X. Boyen, H. Shacham. Short Group Signatures. In *Crypto'04*, *LNCS* 3152, pp. 41–55. Springer, 2004.
14. D. Boneh, D. Freeman, J. Katz, B. Waters. Signing a Linear Subspace: Signature Schemes for Network Coding. In *PKC'09*, *LNCS* 5443, pp. 68–87, 2009.
15. D. Boneh, D. Freeman. Linearly Homomorphic Signatures over Binary Fields and New Tools for Lattice-Based Signatures. In *PKC'11*, *LNCS* 6571, pp. 1–16, 2011.
16. D. Boneh, D. Freeman. Homomorphic Signatures for Polynomial Functions. In *Eurocrypt'11*, *LNCS* 6632, pp. 149–168, 2011.
17. J. Camenisch, M. Dubovitskaya, K. Haralambiev. Efficient Structure-Preserving Signature Scheme from Standard Assumptions. In *Security and Cryptography for Networks 2012 (SCN 2012)*, *LNCS* 7485, pp. 76–94, 2012.
18. J. Camenisch, K. Haralambiev, M. Kohlweiss, J. Lapon, V. Naessens. Structure Preserving CCA Secure Encryption and Applications. In *Asiacrypt'11*, *LNCS* 7073, pp. 89–106, 2011.
19. J. Camenisch, T. Gross, T.-S. Heydt-Benjamin. Rethinking accountable privacy supporting services: extended abstract. In *Digital Identity Management 2008 (DIM'08)*, pp. 1–8, 2008.
20. R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *FOCS'01* pp. 136–145, 2001.
21. R. Canetti, Y. Dodis, R. Pass, S. Walfish. Universally Composable Security with Global Setup. In *TCC'07*, *LNCS* 4392, pp. 61–85, 2007.
22. R. Canetti, M. Fischlin. Universally Composable Commitments. In *Crypto'01*, *LNCS* 2139, pp. 19–40, 2001.
23. D. Catalano, D. Fiore, B. Warinschi. Adaptive Pseudo-free Groups and Applications. In *Eurocrypt'11*, *LNCS* 6632, pp. 207–223, 2011.
24. D. Catalano, D. Fiore, B. Warinschi. Efficient Network Coding Signatures in the Standard Model. In *PKC'12*, *LNCS* 7293, pp. 680–696, 2012.
25. J. Cathalo, B. Libert, M. Yung. Group Encryption: Non-Interactive Realization in the Standard Model. In *Asiacrypt'09*, *LNCS* 5912, pp. 179–196, 2009.
26. M. Chase, M. Kohlweiss. A New Hash-and-Sign Approach and Structure-Preserving Signatures from DLIN. In *Security and Cryptography for Networks 2012 (SCN 2012)*, *LNCS* 7485, pp. 131–148, 2012.
27. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Crypto'98*, LNCS 1462, pages 13–25, 1998.
28. I. Damgård, J. Groth. Non-interactive and reusable non-malleable commitment schemes. In *STOC'03*, pages 426–437, 2003.
29. G. Di Crescenzo, Y. Ishai, R. Ostrovsky. Non-Interactive and Non-Malleable Commitment. In *STOC'98*, pp. 141–150, 1998.

30. Y. Desmedt. Computer security by redefining what a computer is. In *New Security Paradigms Workshop (NSPW) 1993*, pp. 160–166, 1993.
31. D. Dolev, C. Dwork, M. Naor. Non-malleable cryptography. In *STOC'91*, pages 542–552. ACM Press, 1991.
32. M. Fischlin, B. Libert, M. Manulis. Non-interactive and Re-usable Universally Composable String Commitments with Adaptive Security. In *Asiacrypt'11*, *LNCS* 7073, pp. 468–485, 2011.
33. D. Freeman. Improved security for linearly homomorphic signatures: A generic framework. In *PKC'12*, *LNCS* 7293, pp. 697–714, 2012.
34. G. Fuchsbauer. Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. Cryptology ePrint Archive: Report 2009/320, 2009.
35. J. Garay, P. MacKenzie, K. Yang Strengthening Zero-Knowledge Protocols Using Signatures. In *Eurocrypt'03*, *LNCS* 2656, pp. 177–194, 2003.
36. R. Gennaro. Multi-trapdoor Commitments and Their Applications to Proofs of Knowledge Secure Under Concurrent Man-in-the-Middle Attacks. In *Crypto'04*, *LNCS* 3152, pp. 220–236, 2004.
37. R. Gennaro, C. Gentry, B. Parno. Non-interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers. In *Crypto 2010*, *LNCS* 6223, pp. 465–482, 2010.
38. R. Gennaro, J. Katz, H. Krawczyk, T. Rabin. Secure Network Coding over the Integers. In *PKC'10*, *LNCS* 6056, pp. 142–160, 2010.
39. R. Gennaro and S. Micali. Independent Zero-Knowledge Sets. In *ICALP'06*, LNCS 4052, pages 34–45, 2006.
40. J. Groth. Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In *Asiacrypt'06*, *LNCS* 4284, pp. 444–459, Springer, 2006.
41. J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pp. 415–432, 2008.
42. J. Groth. Homomorphic trapdoor commitments to group elements. Cryptology ePrint Archive: Report 2009/007, 2009.
43. J. Groth. Efficient Zero-Knowledge Arguments from Two-Tiered Homomorphic Commitments. In *Asiacrypt'11*, *LNCS* 7073, pp. 431–448, 2011.
44. D. Hofheinz, T. Jager. Tightly Secure Signatures and Public-Key Encryption. In *Crypto'12*, *LNCS* 7417, pp. 590–607, 2012.
45. B. Libert, M. Yung. Non-Interactive CCA2-Secure Threshold Cryptosystems with Adaptive Security: New Framework and Constructions. In *TCC 2012*, LNCS 7194, pp. 75–93, Springer, 2012.
46. R. Johnson, D. Molnar, D. Song, D. Wagner. Homomorphic Signature Schemes. In *CT-RSA'02*, LNCS 2271, pp. 244–262, 2002.
47. P. MacKenzie, K. Yang. On Simulation-Sound Trapdoor Commitments. In *Eurocrypt'04*, *LNCS* 3027, pp. 382–400, 2004.
48. M. Naor, M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC'90*, ACM Press, 1990.
49. Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai, T. Matsuda, and K. Omote. Group Signatures with Message-Dependent Opening. In *5th International Conference on Pairing-Based Cryptography (Pairing 2012)*, LNCS 7708, pp. 270–294, 2013.
50. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *Crypto'84*, *LNCS* 196, pp. 47–53, 1984.
51. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *Eurocrypt'05*, *LNCS* 3494, pp. 114–127, 2005.