Structure-Preserving Chosen-Ciphertext Security with Shorter Verifiable Ciphertexts

Benoît Libert, Thomas Peters, Chen Qian

Structure-Preserving Chosen-Ciphertext Security With Shorter Verifiable Ciphertexts

Benoît Libert[1], Thomas Peters[2], and Chen Qian[3]

[1] CNRS, Laboratoire LIP (CNRS, ENSL, U. Lyon, Inria, UCBL), ENS de Lyon (France)
[2] FNRS & UCLouvain, ICTEAM (Belgium)
[3] IRISA, Rennes (France)

Abstract. Structure-preserving cryptography is a world where messages, signatures, ciphertexts and public keys are entirely made of elements of a group over which a bilinear map is efficiently computable. While structure-preserving signatures have received much attention the last 6 years, structure-preserving encryption schemes have undergone slower development. In particular, the best known structure-preserving cryptosystems with chosen-ciphertext (IND-CCA2) security either rely on symmetric pairings or require long ciphertexts comprised of hundreds of group elements or do not provide publicly verifiable ciphertexts. We provide a publicly verifiable construction based on the SXDH assumption in asymmetric bilinear groups $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$, which features relatively short ciphertexts. For typical parameters, our ciphertext size amounts to less than 40 elements of $\mathbb{G}$. As a second contribution, we provide a structure-preserving encryption scheme with perfectly randomizable ciphertexts and replayable chosen-ciphertext security. Our new RCCA-secure system significantly improves upon the best known system featuring similar properties in terms of ciphertext size.

Keywords. Structure-preserving encryption, chosen-ciphertext security, RCCA security, public ciphertext verifiability.

1 Introduction

Structure-preserving cryptography is a paradigm where handled objects all live in discrete-log-hard abelian groups over which a bilinear map is efficiently computable. The structure-preserving property allows for a smooth interaction of the considered primitives with Groth-Sahai (GS) proof systems [36], making them very powerful tools for the modular design of privacy-preserving cryptographic protocols [3, 8, 16, 17, 19, 27, 32, 37, 44, 51].

In structure-preserving signatures (SPS) [6, 8], messages, signatures, public keys all live in the source groups $(\mathbb{G}, \hat{\mathbb{G}})$ of a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$. The roots of SPS schemes can be traced back to the work of Groth [34], which initiated a line of work seeking to obtain short signatures [4–6, 23, 40, 45], security under standard assumptions [4, 18, 24, 37, 40, 45], tight security proofs [5, 37] or lower

bounds [1,7]. Beyond signatures, structure-preserving cryptography was also developed in the context of commitment schemes [6,9,10,35,42], public-key [5,16] and identity-based encryption [41,52] as well as in deterministic primitives [2].

Structure-preserving encryption. Camenisch *et al.* [16] came up with the first chosen-ciphertext-secure (IND-CCA2) structure-preserving public-key encryption scheme. Structure-preserving CCA2 security is motivated by applications in the realization of oblivious third parties protocols [20] or proofs of knowledge of leakage-resilient signatures [28]. Among the use cases of structure-preserving CCA-secure encryption, [16] mentions various settings where a user, who has a ciphertext and a Groth-Sahai proof of its well-formedness, wants to convince a third party that it is in possession of such a ciphertext without revealing it. Structure-preserving encryption also allows two users to jointly compute an encryption (of a function) of two plaintexts such that neither player learns the plaintext of the other player and only one of them obtains the ciphertext.

As pointed out in [16], structure-preserving encryption should make it possible to efficiently and non-interactively prove possession of a valid ciphertext, which rules out the use of standard techniques – like hash functions [26] or ordinary (i.e., non-structure-preserving) one-time signatures [21,29,50] – that are typically used to achieve chosen-ciphertext security [49] in the standard model. In particular, the original Cramer-Shoup cryptosystem [26] does not provide the sought-after structure-preserving property and neither do direct applications of the Canetti-Halevi-Katz paradigm [21]: for example, merely combining Kiltz's tag-based encryption [39] with a one-time SPS does not work as the security proof of [39] requires (hashed) verification keys to be encoded as exponents. Nevertheless, Camenisch *et al.* [16] managed to twist the design principle of Cramer-Shoup [26] so as to obtain a variant of the scheme that only resorts to algebraic operations when it comes to tying all ciphertexts components altogether in a non-malleable manner.

While efficient and based on the standard Decision Linear assumption [14], the initial construction of [16] still suffers from certain disadvantages. In the first variant of their scheme, for example, one of the ciphertext components lives in the target group $\mathbb{G}_T$ of a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \rightarrow \mathbb{G}_T$ which complicates its use in applications requiring to prove knowledge of a ciphertext: recall that Groth-Sahai proofs require witnesses to live in the *source* group of a bilinear (i.e., they need *strictly* structure-preserving components in the sense of [9]). While Camenisch *et al.* [16] suggested a technique of moving all ciphertext components to the source groups in their scheme, this is only known to be possible using symmetric bilinear groups (where $\mathbb{G} = \hat{\mathbb{G}}$) as it relies on the one-sided pairing randomization technique of [8]. Another limitation of [16] is that, analogously to the original Cramer-Shoup system [26], valid ciphertexts (i.e., which lie in the range of the legitimate encryption algorithm) are not publicly recognizable. As a result, only the sender of a ciphertext (who knows the random encryption coins) can generate a proof that this particular ciphertext is indeed a valid ciphertext without revealing it. Ideally, any ciphertext observer should be able to commit to that ciphertext and prove statements about it without any interaction with

the sender, which would be possible with publicly verifiable ciphertexts.

Abe *et al.* [5] provided several constructions of structure-preserving CCA2-secure encryption with publicly verifiable ciphertexts. On the downside, their solutions incur substantially longer ciphertexts than [16]: under the Decision Linear assumption, the most efficient solution of [5] entails 321 group elements per ciphertext. Moreover, it was only described in terms of symmetric pairings.

In addition, symmetric pairings have become significantly less efficient (see, e.g., [31]) as the use of small-characteristic fields is now considered insecure [11]. This motivates the search for efficient structure-preserving CCA2-secure systems which provide shorter ciphertexts and can operate in asymmetric pairings.

Our Contributions. We provide a new CCA2-secure structure-preserving encryption scheme wherein the validity of ciphertexts is publicly verifiable and ciphertexts only consist of 16 elements of $\mathbb{G}$ and 11 elements of $\hat{\mathbb{G}}$. By "public verifiability", we mean that ciphertexts which are rejected by the decryption algorithm should be recognizable given the public key. While stronger definitions of verifiability could be used[4], this notions suffices to ensure confidentiality in settings – like threshold decryption [13, 46, 54] – where potentially harmful decryption queries should be publicly detectable. In particular, our first scheme readily implies a CCA2-secure structure-preserving cryptosystem that enables threshold decryption in the adaptive corruption setting.

In our first scheme, the ciphertext size amounts to 38 elements of $\mathbb{G}$ assuming that each element of $\hat{\mathbb{G}}$ has a representation which is twice as large as the representation of $\mathbb{G}$ elements. The security is proved under the standard symmetric eXternal Diffie-Hellman (SXDH) assumption [53] in asymmetric bilinear maps.

As a second contribution, we provide a different structure-preserving cryptosystem which features perfectly re-randomizable ciphertexts and replayable chosen-ciphertext (RCCA) security. As defined by Canetti, Krawczyk and Nielsen [22], RCCA security is a meaningful relaxation of CCA2 security that tolerates a "benign" form of malleability: namely, anyone should be able to randomize a given ciphertext into another encryption of the same plaintext. Under the SXDH assumption, our construction features statistically randomizable ciphertexts which only consist of 34 elements of $\mathbb{G}$ and 18 elements of $\hat{\mathbb{G}}$. Under the same[5] assumption, the best known RCCA-secure realization thus far was the scheme of Chase *et al.* [25] which costs 49 elements of $\mathbb{G}$ and 20 elements of $\hat{\mathbb{G}}$.

Our techniques. Our structure-preserving CCA2 secure cryptosystem builds on a public-key encryption scheme suggested by Libert and Yung [46], which is not structure-preserving in its original form. Our starting observation is that, unlike Kiltz's tag-based encryption scheme [39], the security proof of [46] does not require to interpret one-time signature verification keys as exponents. The

[4] For example, we could additionally require that all ciphertexts outside the range of the decryption algorithm are rejected by the decryption procedure.

[5] The authors of [25] only described a construction from the DLIN assumption with 93 elements per ciphertext. Their approach extends to the SXDH assumption and happens to provide structure-preserving schemes.

construction of [46] is obtained by tweaking the Cramer-Shoup paradigm [26] and replacing the designated verifier NIZK proofs of ciphertext validity by a universally verifiable Groth-Sahai proof. In order to obtain publicly verifiable proofs with the desired security property called *simulation-soundness* [50], the authors of [46] used Groth-Sahai common reference strings (CRSes) which depend on the verification key of a one-time signature. In the security proof, the key idea was to enable the simulation of fake NIZK proofs of ciphertext validity while making it impossible for the adversary to create such a fake proof himself. In Groth-Sahai proofs, this can be achieved by programming the Groth-Sahai CRSes in such a way that they form a linear subspace of dimension 1 in the challenge ciphertext whereas adversarially-generated ciphertexts involve CRSes of dimension 2 (which are perfectly sound CRSes).

We build on the observation that the approach of [46] still works if one-time verification keys consist of group elements instead of exponents. One difficulty is that we need one-time signature verification keys comprised of a single group element while the best known one-time SPS [6] have longer verification keys. Our solution is to "hash" the one-time verification keys of [6] in a structure-preserving manner. For this purpose, we apply a strictly structure-preserving commitment scheme proposed by Abe *et al.* [10] as if it was a chameleon hash function: namely, we replace the hash value by a commitment to the one-time verification key while the corresponding de-commitment information is included in the ciphertext. One caveat is that [10] considers a relaxed security notion for strictly structure-preserving commitments, called *chosen-message target collision-resistance*, which appears insufficient for our purposes. We actually need a stronger notion, called *enhanced chosen-message target collision-resistance* (ECM-TCR), where the adversary should also be able to come up with a different opening to the same message for a given commitment. Fortunately, we can prove that the strictly structure-preserving commitment of [10] *does* provide ECM-TCR security under the SXDH assumption.

The security proof of our construction addresses another technical hurdle which arises from the fact that ciphertexts contain elements from both sources groups $\mathbb{G}$ and $\hat{\mathbb{G}}$. Directly adapting the security proof of [46] would require to sign all elements of $\mathbb{G}$ and $\hat{\mathbb{G}}$ that are contained in the ciphertext, which would require a one-time SPS where messages contain elements of both groups $(\mathbb{G}, \hat{\mathbb{G}})$. While such schemes exist [4], they are less efficient than one-time SPS schemes for unilateral messages. Our solution to this problem is to modify the security proof of Libert and Yung [46] in such a way that not all ciphertexts components have to be signed using the one-time signature. In short, we leverage the fact that only Groth-Sahai commitments have to live in the group $\hat{\mathbb{G}}$: proof elements and other components of the ciphertext can indeed dwell in $\mathbb{G}$. In GS commitments for linear multi-exponentiation equations, we notice that Groth-Sahai commitments are uniquely determined by the proof elements and the statement. For this reason, even if the adversary tampers with the GS commitments of the challenge ciphertext, it will be unable to create another ciphertext that will be accepted by the decryption oracle. This saves us from having to one-time-sign

the Groth-Sahai commitments in the encryption algorithm, which is the reason why we only need such a system for unilateral messages.

Our construction of RCCA-secure encryption extends the ideas of Chase *et al.* [25]. In a nutshell, the RCCA-secure scheme of [25] combines a semantically secure encryption scheme and a randomizable witness indistinguishable proof of a statement of the form "Either I know the plaintext OR a signature of a ciphertext that this ciphertext is a randomization of". Our construction proceeds in an analogous way by demonstrating a statement of the form "Either I know the plaintext OR this ciphertext is a randomization of the challenge ciphertext".

In a high level, for the two branches of the statement we rely on proofs which nicely share a common structure to optimize our OR-proof. On the one hand, for the knowledge of the plaintext we use a quasi-adaptive NIZK (QA-NIZK) proof, which are NIZK proofs introduced by [38] where the CRS may depend on the specific language for which proofs have to be generated. Our QA-NIZK is built from the one-time structure-preserving linearly homomorphic signature (LHSPS) of Libert, Peters, Joye and Yung [42]. On the other hand, for the one-time signature we use the strongly unforgeable one-time SPS of Abe et al. [5] that we make re-randomizable thanks to LHSPS. These tools allows to combine some of the verification equations for which Groth-Sahai proofs of satisfiability are included in ciphertexts.

Related Work. Several different approaches [15, 30, 47, 48] were taken to reconcile chosen-ciphertext-security and homomorphism. Relaxed flavors of chosen-ciphertext security [22] opened the way to perfectly randomizable encryption schemes offering stronger guarantees than just semantic security. Groth described [33] a weakly RCCA secure variant of Cramer-Shoup which only encrypts messages in a bit-by-bit manner. Prabhakaran and Rosulek [47] showed how to more efficiently encrypt many bits at once in a RCCA-secure realization from the DDH assumption. While their solution features shorter ciphertexts than our RCCA-secure scheme, it is not structure-preserving as it cannot be readily instantiated in groups with a bilinear maps. On the other hand, unlike our scheme and the one of [25], it allows re-randomizing ciphertexts without knowing under which public key they were encrypted.

Prabhakaran and Rosulek subsequently generalized the RCCA security notion [22] into a model [48] of homomorphic encryption that only supports a limited form of malleability. Boneh, Segev and Waters [15] took a different approach aiming for restricted malleability properties. Chase *et al.* [25] considered a modular design of HCCA-secure encryption [48] based on malleable proof systems. Their proposals turn out to be the only known HCCA/RCCA-secure structure-preserving candidates thus far.

2 Background and Definitions

2.1 Hardness Assumptions

We consider groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime-order p endowed with a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$.

Definition 1. *The* **Diffie-Hellman problem** *(DDH) in* $\mathbb{G}$*, is to distinguish the distributions* (g, g^a, g^b, g^{ab}) *and* (g, g^a, g^b, g^c) *with* $a, b, c \xleftarrow{R} \mathbb{Z}_p$*. The Diffie-Hellman assumption asserts the intractability of DDH for any PPT distinguisher.*

In the asymmetric setting $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$, we consider the SXDH assumption, which posits that the DDH assumption holds in both $\mathbb{G}$ and $\hat{\mathbb{G}}$.

Definition 2. *The* **Double Pairing problem** *(DP) in* $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ *is, given a pair of group elements* $(\hat{g}_z, \hat{g}_r) \in \hat{\mathbb{G}}^2$*, to find a non-trivial triple* $(z, r) \in \mathbb{G}^2 \backslash \{(1_{\mathbb{G}}, 1_{\mathbb{G}})\}$ *such that* $e(z, \hat{g}_z) \cdot e(r, \hat{g}_r) = 1_{\mathbb{G}_T}$*.*

It is known [8] that the DP assumption is implied by the DDH assumption in $\mathbb{G}$. By exchanging the roles of $\mathbb{G}$ and $\hat{\mathbb{G}}$ in the definition of DP, we obtain a variant of the assumption which implies the hardness of DDH in $\hat{\mathbb{G}}$.

2.2 One-Time Structure-Preserving Signatures

Structure-preserving signatures (SPS) [6, 8] are signature schemes where messages and public keys all consist of elements of a group over which a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ is efficiently computable. Constructions based on simple assumptions were put forth in [4, 5].

In the forthcoming sections, we will rely on one-time SPS schemes.

Definition 3. *A one-time signature scheme is a tuple of efficient algorithms* $\mathcal{OTS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *where:*

Setup(λ) : *This algorithm takes as input a security parameter* λ *and generates the public parameters* PP *for the scheme.*

KeyGen (PP) : *This algorithm takes as input* PP *and generates a one-time secret key* osk *and a one-time verification key* ovk*.*

Sign$(\mathsf{PP}, \mathsf{osk}, \boldsymbol{M})$: *Given as input* $(\mathsf{PP}, \mathsf{osk})$ *and a message* $\boldsymbol{M}$*, this algorithm produces a signature* σ *for* $\boldsymbol{M}$*.*

Verify$(\mathsf{PP}, \mathsf{ovk}, \boldsymbol{M}, \sigma)$: *The verification algorithm takes* $(\mathsf{PP}, \mathsf{ovk}, \boldsymbol{M}, \sigma)$ *and returns* 1 *or* 0*.*

Correctness mandates that, for any $\lambda \in \mathbb{N}$, any $\mathsf{PP} \leftarrow \mathsf{Setup}(\lambda)$, any pair $(\mathsf{osk}, \mathsf{ovk}) \leftarrow \mathsf{KeyGen}(\mathsf{PP})$, we have $\mathsf{Verify}(\mathsf{PP}, \mathsf{ovk}, \boldsymbol{M}, \mathsf{Sign}(\mathsf{PP}, \mathsf{osk}, \boldsymbol{M})) = 1$ for any message $\boldsymbol{M}$.

In addition, a one-time signature is said *structure-preserving* if the components of ovk, $\boldsymbol{M}$ and σ all live in the source groups $(\mathbb{G}, \hat{\mathbb{G}})$ of a configuration $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of bilinear groups.

Definition 4. *A one-time signature scheme* $\mathcal{OTS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is strongly unforgeable against chosen message attack* $(SUF\text{-}CMA)$ *if*

$$\mathbf{Adv}^{SUF\text{-}CMA}_{OTS,\mathcal{A}} = \Pr\left[\begin{array}{l|l} (m^\star, \sigma^\star) \notin Q_{\mathsf{Sign}^{OT}} \wedge & \mathsf{PP} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{Verify}(\mathsf{ovk}, m^\star, \sigma^\star) = 1 & (\mathsf{ovk}, \mathsf{osk}) \leftarrow \mathsf{KeyGen}(\mathsf{PP}) \\ & (m^\star, \sigma^\star) \leftarrow \mathcal{A}^{\mathsf{Sign}^{OT}_{\mathsf{osk}}(\cdot)}(\mathsf{ovk}) \end{array}\right]$$

is negligible against any PPT adversary $\mathcal{A}$. Here, $\mathsf{Sign}^{OT}_{\mathsf{osk}}(\cdot)$ is a signing oracle which allows the adversary to obtain a signature σ_m of only one message m for which (m, σ_m) is stored in $Q_{\mathsf{Sign}^{OT}}$.

We recall a construction of the one-time Structure-Preserving Signature scheme which was proposed in [5].

Setup(λ) : Choose asymmetric bilinear groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ and output $\mathsf{PP} = (\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$.

KeyGen(PP) : Generates the signing key osk and the verification key ovk using the security parameter λ and the number n of messages to be signed.

1. Choose $\hat{g}_z, \hat{g}_r, g \xleftarrow{R} \hat{\mathbb{G}}$.
2. For $i = 1$ to n, pick $(\chi_i, \gamma_i) \xleftarrow{R} \mathbb{Z}_p^2$ and compute $\hat{g}_i = \hat{g}_z^{\chi_i} \hat{g}_r^{\gamma_i}$.
3. Pick $(\zeta, \rho) \xleftarrow{R} \mathbb{Z}_p^2$ and compute $\hat{A} = g_z^\zeta \cdot g_r^\rho$.
4. Set $\mathsf{osk} = (\{(\chi_i, \gamma_i)\}_{i=1}^n, \zeta, \rho) \in \mathbb{G}^{2n+2}$ and

$$\mathsf{ovk} = (\hat{g}_z, \hat{g}_r, \{\hat{g}_i\}_{i=1}^n, \hat{A}) \in \hat{\mathbb{G}}^{n+3}.$$

Sign$(\mathsf{osk}, \boldsymbol{M} = (M_1, \ldots, M_n))$: In order to sign $\boldsymbol{M} = (M_1, \ldots, M_n) \in \mathbb{G}^n$, compute $z = g^\zeta \prod_{i=1}^n M_i^{\chi_i}$ and $r = g^\rho \prod_{i=1}^n M_i^{\gamma_i}$. Output $\sigma = (z, r)$.

Verify$(\mathsf{ovk}, \boldsymbol{M} = (M_1, \ldots, M_n), \sigma = (z, r))$: Return 1 if and only if the following equations are satisfied: $e(z, \hat{g}_z) \cdot e(r, \hat{g}_r) = e(g, \hat{A}) \cdot \prod_{i=1}^n e(M_i, \hat{g}_i)$.

2.3 Partial One-time Signature

A special case of the one-time signature presented in Section 2.2 is called Partial One-Time Signature (POTS) [12]. In a such scheme, part of the verification key can be re-used in multiple signatures and the remaining part must be refreshed at every signature generation.

Definition 5. *A partial one-time signature (POTS) scheme is a tuple of algorithms $POTS = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{OKeyGen}, \mathsf{Sign}, \mathsf{Verify})$.*

Setup($\boldsymbol{\lambda}$) : *The setup algorithm takes as input a security parameter λ and generates the public parameters PP for the scheme.*

KeyGen(PP) : *The key generation algorithm takes as input the public parameters PP and generates the long-term signing key sk and long-term verification key vk.*

OKeyGen(PP) : *The key generation algorithm takes PP and generates the one-time signing key osk and the one-time verification key ovk.*

Sign($\mathsf{PP}, \mathsf{sk}, \mathsf{osk}, \boldsymbol{M}$) : *The signature algorithm uses the $(\mathsf{PP}, \mathsf{osk})$ to produce a valid signature σ for the message vector $\boldsymbol{M}$.*

Verify($\mathsf{PP}, \mathsf{vk}, \mathsf{ovk}, \boldsymbol{M}, \sigma$) : *The verification algorithm takes $(\mathsf{PP}, \mathsf{vk}, \mathsf{ovk}, \boldsymbol{M}, \sigma)$ and returns 1 or 0.*

Correctness requires that, for any $\mathsf{PP} \leftarrow \mathsf{Setup}(\lambda)$, $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(\mathsf{PP})$ and $(\mathsf{osk}, \mathsf{ovk}) \leftarrow \mathsf{OKeyGen}(\mathsf{PP})$, the partial one-time signature scheme is correct if and only if $\mathsf{Verify}(\mathsf{PP}, \mathsf{vk}, \mathsf{ovk}, \boldsymbol{M}, \mathsf{Sign}(\mathsf{PP}, \mathsf{sk}, \mathsf{osk}, \boldsymbol{M})) = 1$.

We focus on the strong unforgeability against one-time chosen-message attack of our POTS.

Definition 6. *A POTS scheme* $POTS = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{OKeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is strongly unforgeable against one-time chosen-message attack (or OT-CMA secure) if:*

$$\mathbf{Adv}^{OT\text{-}SU\text{-}CMA}_{POTS,\mathcal{A}}(\lambda) = \Pr\left[\begin{array}{l|l} \exists\ (m', \sigma') \text{ s.t. } (\mathsf{ovk}^\star, \sigma', m') \in Q & \mathsf{PP} \leftarrow \mathsf{Setup}(1^\lambda) \\ \wedge\ (\mathsf{ovk}^\star, \sigma^\star, m^\star) \notin Q & (\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{PP}) \\ \wedge\ \mathsf{Verify}(\mathsf{vk}, \mathsf{ovk}^\star, m^\star, \sigma^\star) = 1 & (\mathsf{ovk}^\star, \sigma^\star, m^\star) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{sk}}}(\mathsf{PP}, \mathsf{vk}) \end{array}\right]$$

is negligible for any PPT adversary $\mathcal{A}$*. Here, the signing oracle takes as input a message* m*, generates* $(\mathsf{ovk}, \mathsf{osk}) \leftarrow \mathsf{OKeyGen}(\mathsf{PP})$*,* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{osk}, m)$*. Then, it records* (ovk, m) *to* Q *and returns* (σ, ovk)*.*

Here, we recall an instantiation of the POTS scheme [4], which is strongly unforgeable against the one-time chosen-message attack (SU-OTCMA) under the DP assumption.

$\mathsf{Setup}(\lambda, \ell)$: On input of a security parameter λ and an integer $\ell \in poly(\lambda)$, the setup algorithm chooses a large prime $p > 2^\lambda$, asymmetric groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime order p, with a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ and the corresponding generators $(g, \hat{g}) \in \mathbb{G} \times \hat{\mathbb{G}}$. The algorithm outputs

$$\mathsf{PP} = (p, \mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, e, g, \hat{g}, \ell).$$

$\mathsf{KeyGen}(\mathsf{PP})$: Parse PP as $(p, \mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, e, g, \hat{g}, \ell)$. Choose $w_z \xleftarrow{R} \mathbb{Z}_p^*$ and compute $g_z \leftarrow g^{w_z}$. For $i \in \{1, \ldots, \ell\}$, choose $\chi_i \xleftarrow{R} \mathbb{Z}_p$ and compute $g_i \leftarrow g^{\chi_i}$. Return

$$\mathsf{vk} = (g_z, g_1, \ldots, g_\ell) \in \mathbb{G}^{\ell+1} \qquad \mathsf{sk} = (w_z, \chi_1, \ldots, \chi_\ell) \in \mathbb{Z}_p^{\ell+1}$$

$\mathsf{OKeyGen}(\mathsf{PP})$: Parse PP, choose $a \leftarrow \mathbb{Z}_p$, compute $A \leftarrow g^a$ and output

$$\mathsf{ovk} = A \qquad \mathsf{osk} = a$$

$\mathsf{Sign}(\mathsf{sk}, \mathsf{osk}, \hat{\boldsymbol{M}})$: Parse $\hat{\boldsymbol{M}}$ as $(\hat{M}_1, \ldots, \hat{M}_\ell) \in \hat{\mathbb{G}}^\ell$. Parse sk and osk, choose $\zeta \xleftarrow{R} \mathbb{Z}_p^*$,then compute and output

$$\hat{Z} = \hat{g}^\zeta \qquad \hat{R} = \hat{g}^{a - \zeta w_z} \prod_{i=1}^{\ell} \hat{M}_i^{-\chi_i}.$$

$\mathsf{Verify}(\mathsf{vk}, \mathsf{ovk}, \hat{\boldsymbol{M}}, \boldsymbol{\sigma})$: Parse σ as $(\hat{Z}, \hat{R}) \in \hat{\mathbb{G}}^2$, $\hat{\boldsymbol{M}}$ as $(\hat{M}_1, \ldots, \hat{M}_\ell) \in \hat{\mathbb{G}}^\ell$ and ovk as $A \in \mathbb{G}$. The algorithm returns 1 if the following equation holds:

$$e(A, \hat{g}) = e(g_z, \hat{Z}) \cdot e(g, \hat{R}) \cdot \prod_{i=1}^{\ell} e(g_i, \hat{M}_i)$$

otherwise the algorithm returns 0.

2.4 One-Time Linearly Homomorphic Structure-Preserving Signatures

Libert *el. al.* [42] considered structure-preserving with linear homomorphic properties (see Appendix B.1 for formal definitions). This section recalls the one-time linearly homomorphic structure-preserving signature (LHSPS) of [42].

Keygen(λ, n): Given a security parameter λ and the dimension $n \in \mathbb{N}$ of the subspace to be signed, choose bilinear group $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime order p. Then, choose $\hat{g_z}, \hat{g_r} \xleftarrow{R} \hat{\mathbb{G}}$. For $i = 1$ to n, pick $\chi_i, \gamma_i \xleftarrow{R} \mathbb{Z}_p$ and compute $\hat{g}_i = \hat{g_z}^{\chi_i} \hat{g_r}^{\gamma_i}$. The private key is defined to be $\mathsf{sk} = \{(\chi_i, \gamma_i)\}_{i=1}^n$ while the public key is $\mathsf{pk} = \left(\hat{g_z},\ \hat{g_r},\ \{\hat{g}_i\}_{i=1}^n\right) \in \hat{\mathbb{G}}^{n+2}$.

Sign($\mathsf{sk}, (M_1, \ldots, M_n)$): To sign a $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $\mathsf{sk} = \{(\chi_i, \gamma_i)\}_{i=1}^n$, output $\sigma = (z, r) \in \mathbb{G}^2$, where $z = \prod_{i=1}^n M_i^{\chi_i}$, $r = \prod_{i=1}^n, M_i^{\gamma_i}$.

SignDerive($\mathsf{pk}, \{(\omega_i, \sigma^{(i)})\}_{i=1}^{\ell}$): given pk as well as ℓ tuples $(\omega_i, \sigma^{(i)})$, parse $\sigma^{(i)}$ as $\sigma^{(i)} = \left(z_i, r_i\right)$ for $i = 1$ to ℓ. Compute and return $\sigma = (z, r)$, where $z = \prod_{i=1}^{\ell} z_i^{\omega_i}$, $r = \prod_{i=1}^{\ell} r_i^{\omega_i}$.

Verify($\mathsf{pk}, \sigma, (M_1, \ldots, M_n)$): Given a signature $\sigma = (z, r) \in \mathbb{G}^2$ and a vector $(M_1, \ldots, M_n)$, return 1 iff $(M_1, \ldots, M_n) \neq (1_{\mathbb{G}}, \ldots, 1_{\mathbb{G}})$ and (z, r) satisfy

$$e(z, \hat{g_z}) \cdot e(r, \hat{g_r}) = \prod_{i=1}^{n} e(M_i, \hat{g}_i).$$

The one-time security of the scheme (in the sense of Definition 9 in Appendix B.1) was proved [42] under the DP assumption. In short, the security notion implies the infeasibility of deriving a signature on a vector outside the subspace spanned by the vectors authenticated by the signer. Here, "one-time" security means that a given public key allows signing only one subspace.

We remark that the one-time structure-preserving signature of Section 2.2 can be seen as a special case of the above LHSPS scheme, in which we fix the first element of the vector to be signed. The one-time security of this signature scheme can be directly deduced from the security of the LHSPS scheme.

2.5 Strictly Structure-Preserving (Trapdoor) Commitments

In this section, we recall the notion of Chosen-Message Target Collision Trapdoor Commitment as it was defined by Abe *el. al.* [10].

Definition 7. *A non-interactive commitment scheme is a tuple of polynomial-time algorithms* $\{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Commit}, \mathsf{Verify}\}$ *that:*

Setup(λ)**:** *The parameter generation algorithm takes the security parameter* λ *and outputs a public parameter* PP*.*

KeyGen(PP)**:** *The key generation algorithm takes* PP *and outputs the commitment key* ck*.*

Com($\mathsf{PP}, \mathsf{ck}, m$)**:** *The commitment algorithm takes* $(\mathsf{PP}, \mathsf{ck})$ *and a message* m*, then it outputs a commitment* com *and an opening information* open*.*

Verify$(\mathsf{PP}, \mathsf{com}, m, \mathsf{open})$: *The verification algorithm takes* $(\mathsf{PP}, \mathsf{com}, m, \mathsf{open})$ *and outputs* 1 *or* 0*.*

In trapdoor commitment schemes, the Setup algorithm additionally outputs a trapdoor tk which, on input of a message m and random coins r such that $c = \mathsf{Com}(\mathsf{PP}, \mathsf{ck}, m; r)$, allows opening the commitment c to any message m'. In our construction, we need a length-reducing commitment scheme which satisfies a stronger notion of Chosen-Message Target Collision Resistance (CM-TCR) than the one considered in [10, Definition 10].

Definition 8. *A Commitment Scheme provides* **enhanced chosen-message target collision-resistance** *(ECM-TCR) if the advantage*

$$\mathbf{Adv}^{ECM\text{-}TCR}_{\mathcal{A}}(\lambda) = \Pr\left[\begin{array}{l|l} \exists(m^\dagger, \mathsf{open}^\dagger) \text{ s.t. } (\mathsf{com}^\star, m^\dagger, \mathsf{open}^\dagger) \in Q & \mathsf{PP} \leftarrow \mathsf{Setup}(1^\lambda) \\ \quad \wedge\ (\mathsf{com}^\star, m^\star, \mathsf{open}^\star) \notin Q & \mathsf{ck} \leftarrow \mathsf{KeyGen}(\mathsf{PP}) \\ \quad \wedge\ \mathsf{Verify}(\mathsf{ck}, \mathsf{com}^\star, m^\star, \mathsf{open}^\star) = 1 & (\mathsf{com}^\star, m^\star, \mathsf{open}^\star) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{ck}}}(\mathsf{ck}) \end{array}\right]$$

is negligible for any PPT adversary $\mathcal{A}$*. Here,* $\mathcal{O}_{\mathsf{ck}}$ *is an oracle that, given a message* m*, executes* $(\mathsf{com}, \mathsf{open}) \leftarrow \mathsf{Com}(\mathsf{PP}, \mathsf{ck}, m)$*, records* $(\mathsf{com}, m, \mathsf{open})$ *in* Q *and returns* $(\mathsf{com}, \mathsf{open})$*.*

We note that Definition 8 captures a stronger requirement than the original definition [10, Definition 10] in that the latter only requires that the adversary be unable to open a target commitment $\mathsf{com}^\star$ to a different message than the one queried to the oracle $\mathcal{O}_{\mathsf{ck}}$. Here, the adversary is also considered successful if it provides a different opening $\mathsf{open}^\star \neq \mathsf{open}'$ of $\mathsf{com}^\star$ to the same message $m^\star = m^\dagger$ as the one queried to $\mathcal{O}_{\mathsf{ck}}$.

We now recall the Strictly Structure-Preserving Trapdoor Commitment of Abe *el. al.* [10] and show that it actually satisfies our stronger notion of ECM-TCR security.

***TC*.Setup$(\boldsymbol{\lambda}, \boldsymbol{\ell})$** : On input of a security parameter λ and an integer $\ell \in \mathsf{poly}(\lambda)$, the public parameters are generated by choosing a large prime $p > 2^\lambda$, asymmetric groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime order p, with a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ and group generators $(g, \hat{g}) \in \mathbb{G} \times \hat{\mathbb{G}}$. The algorithm outputs

$$\mathsf{PP} = (p, \mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, e, g, \hat{g}, \ell).$$

***TC*.KeyGen(PP)** : For $i = 1, \ldots, \ell + 2$, choose $\rho_i \xleftarrow{R} \mathbb{Z}_p^*$ and compute

$$\hat{X}_i \leftarrow \hat{g}^{\rho_i} \qquad \forall i \in \{1, \ldots, \ell + 2\}.$$

Output the commitment key $\mathsf{ck} := \{\hat{X}_i\}_{i=1}^{\ell+2}$. Optionally, the algorithm may output the trapdoor $tk := \{\rho_i\}_i^{\ell+2}$.

***TC*.Commit$(\mathsf{PP}, \mathsf{ck}, \boldsymbol{M})$** : To commit to $\hat{\boldsymbol{M}} = (\hat{M}_1, \ldots, \hat{M}_\ell) \in \hat{\mathbb{G}}^\ell$, conduct the following step.

1. Generate a key pair $(\mathsf{vk}_{pots}, \mathsf{sk}_{pots})$ for the partial one-time signature of Section 2.3 . Namely, choose $\mathsf{sk}_{pots} \xleftarrow{R} (w_z, \chi_1, \ldots, \chi_\ell) \in \mathbb{Z}_p^{\ell+1}$ and set

$$\mathsf{vk}_{pots} = (g_z, g_1, \ldots, g_\ell) = (g^{w_z}, g^{\chi_1}, \ldots, g^{\chi_\ell}) \in \mathbb{G}^{\ell+1}.$$

2. Choose $a \xleftarrow{R} \mathbb{Z}_p$ and compute $\mathsf{ovk}_{pots} = A = g^a$ and $\mathsf{osk}_{pots} = a$.
3. Using sk_{pots}, generate a partial one-time signature on the message $\hat{\boldsymbol{M}}$ w.r.t. to the one-time secret key osk_{pots}. To this end,
   a. Pick $\zeta_1 \in \mathbb{Z}_p$.
   b. Compute $(\hat{Z}, \hat{R}) \in \hat{\mathbb{G}}^2$ as a partial one-time signature of $\hat{\boldsymbol{M}}$ as

$$\hat{Z} = \hat{g}^{\zeta_1} \qquad\qquad \hat{R} = \hat{g}^{a-\zeta_1 w_z} \prod_{i=1}^{\ell} \hat{M}_i^{\chi_i}$$

4. Generate a commitment to the message.
   a. Set $(m_1, \ldots, m_{\ell+2}) \leftarrow (\chi_1, \ldots, \chi_\ell, w_z, a)$
   b. Parse ck as $(\hat{X}_1, \ldots, \hat{X}_{\ell+2})$.
   c. Choose a random value $\zeta_2 \leftarrow \mathbb{Z}_p^*$ and compute:

$$\hat{C} = \hat{g}^{\zeta_2} \cdot \prod_{i=1}^{\ell+2} \hat{X}_i^{m_i} \qquad\qquad D = g^{\zeta_2}$$

5. Output the commitment $\hat{\mathsf{com}} = \hat{C}$ as well as the opening information

$$\mathsf{open} = \big(D, g_z, g_1, \ldots, g_\ell, A = g^a, \hat{Z}, \hat{R}\big) \in \mathbb{G}^{\ell+3} \times \hat{\mathbb{G}}^2. \tag{1}$$

***TC*.Verify($\mathsf{ck}, \hat{\mathsf{com}}, \hat{\boldsymbol{M}}, \mathsf{open}$):** Given $\hat{\mathsf{com}} = \hat{C} \in \hat{\mathbb{G}}$, parse $\hat{\boldsymbol{M}}$ as $(\hat{M}_1, \ldots, \hat{M}_\ell)$ and open as in (1).

1. Set $\boldsymbol{N} = (N_1, \ldots, N_{\ell+2}) = (g_1, \ldots, g_\ell, g_z, A)$
2. Using $\mathsf{ovk}_{pots} = A \in \mathbb{G}$, return 1 if the following equalities hold:

$$e(g, \hat{C}) = e(D, \hat{g}) \cdot \prod_{i=1}^{\ell+2} e(N_i, \hat{X}_i) \tag{2}$$

$$e(A, \hat{g}) = e(g_z, \hat{Z}) \cdot e(g, \hat{R}) \cdot \prod_{i=1}^{\ell} e(g_i, \hat{M}_i).$$

   Otherwise, return 0.

Using $tk := \{\rho_i\}_i^{\ell+2}$, it is possible to trapdoor-open a commitment $\hat{\mathsf{com}} = \hat{C}$ in the same way as a Pedersen commitment since $\hat{C}$ is nothing but a Pedersen commitment to $(\mathsf{sk}_{pots}, \mathsf{osk}_{pots})$.

We now prove that the above commitment does not only provide CM-TCR security as defined in [10], but also ECM-TCR security. The proof builds on the

same ideas as that of [10] but also takes advantage of the strong unforgeability[6] of the underlying partial one-time signature.

Theorem 1. *The scheme provides ECM-CTR security under the SXDH assumption.*

Proof. For the sake of contradiction, let us assume that a PPT adversary $\mathcal{A}$ can win the game of Definition 8 with noticeable probability. We observe that the adversary can only win in two mutually exclusive cases.

I. $\mathcal{A}$ outputs a commitment $\hat{C}^\star \in \hat{\mathbb{G}}$ for which it provides an opening

$$\begin{aligned} \boldsymbol{M}^\star &= (M_1^\star, \ldots, M_n^\star) \\ \mathsf{open}^\star &= \big(D^\star, g_z^\star, g_1^\star, \ldots, g_\ell^\star, A^\star, \hat{Z}^\star, \hat{R}^\star\big), \end{aligned}$$

where $(D^\star, g_z^\star, g_1^\star, \ldots, g_\ell^\star, A^\star)$ differs from the tuple $(D^\dagger, g_z^\dagger, g_1^\dagger, \ldots, g_\ell^\dagger, A^\dagger)$ returned by $\mathcal{O}_{\mathsf{ck}}$ as part of the opening

$$\mathsf{open}^\dagger = \big(D^\dagger, g_z^\dagger, g_1^\dagger, \ldots, g_\ell^\dagger, A^\dagger, \hat{Z}^\dagger, \hat{R}^\dagger\big),$$

of $\hat{C}^\star$ when $\mathcal{A}$ queried $\mathcal{O}_{\mathsf{ck}}$ to obtain a commitment to $\hat{\boldsymbol{M}}^\dagger = (\hat{M}_1^\dagger, \ldots, \hat{M}_\ell^\dagger)$.

II. $\mathcal{A}$ outputs a commitment $\hat{C}^\star \in \hat{\mathbb{G}}$ which it opens by revealing a pair

$$\begin{aligned} \boldsymbol{M}^\star &= (M_1^\star, \ldots, M_n^\star) \\ \mathsf{open}^\star &= \big(D^\star, g_z^\star, g_1^\star, \ldots, g_\ell^\star, A^\star, \hat{Z}^\star, \hat{R}^\star\big), \end{aligned}$$

such that $(D^\dagger, g_z^\dagger, g_1^\dagger, \ldots, g_\ell^\dagger, A^\dagger) = (D^\star, g_z^\star, g_1^\star, \ldots, g_\ell^\star, A^\star)$. In this case, we must have either $\boldsymbol{M}^\star \neq \boldsymbol{M}^\dagger$ or $(\hat{Z}^\star, \hat{R}^\star) \neq (\hat{Z}^\dagger, \hat{R}^\dagger)$.

Let us first assume that situation I occurs with noticeable probability. We show that $\mathcal{A}$ can be turned into an algorithm $\mathcal{B}_I$ that breaks the DDH assumption in $\hat{G}$ by finding a pair (Z, R) such that $e(Z, \hat{g}) \cdot e(R, \hat{h}) = 1_{\mathbb{G}_T}$ for a given pair $(\hat{g}, \hat{h}) \in \hat{\mathbb{G}}^2$. This algorithm $\mathcal{B}_I$ proceeds in the same way as in [10]. Namely, it creates the commitment key ck by choosing $\rho_i, \theta_i \xleftarrow{R} \mathbb{Z}_p$ and setting $\hat{X}_i = \hat{g}^{\rho_i} \cdot \hat{h}^{\theta_i}$ for each $i \in \{1, \ldots, \ell + 2\}$. It faithfully answers all queries made by $\mathcal{A}$ to $\mathcal{O}_{\mathsf{ck}}$. By hypothesis, $\mathcal{A}$ outputs a commitment $\hat{C}^\star \in \hat{\mathbb{G}}$ as well as an opening $(\boldsymbol{M}^\star, \mathsf{open}^\star)$ which satisfy the conditions of situation I. In particular, $\mathsf{open}^\star = \big(D^\star, g_z^\star, g_1^\star, \ldots, g_\ell^\star, A^\star, \hat{Z}^\star, \hat{R}^\star\big)$ satisfies

$$e(g, \hat{C}^\star) = e(D^\star, \hat{g}) \cdot \prod_{i=1}^{\ell} e(g_i^\star, \hat{X}_i) \cdot e(g_z^\star, \hat{X}_{\ell+1}) \cdot e(A^\star, \hat{X}_{\ell+2}) \tag{3}$$

[6] Note that, while [4] only considered the standard notion of unforgeability, it is straightforward that their scheme also provides strong unforgeability.

and the set Q must contain $\mathsf{open}^\dagger = \big(D^\dagger, g_z^\dagger, g_1^\dagger, \ldots, g_\ell^\dagger, A^\dagger, \hat{Z}^\dagger, \hat{R}^\dagger\big)$ such that

$$e(g, \hat{C}^\star) = e(D^\dagger, \hat{g}) \cdot \prod_{i=1}^{\ell} e(g_i^\dagger, \hat{X}_i) \cdot e(g_z^\dagger, \hat{X}_{\ell+1}) \cdot e(A^\dagger, \hat{X}_{\ell+2}). \tag{4}$$

Dividing (4) out of (3), we find that the pair

$$Z = \Big(\frac{D^\star}{D^\dagger}\Big) \cdot \Big(\frac{g_z^\star}{g_z^\dagger}\Big)^{\rho_{\ell+1}} \cdot \Big(\frac{A^\star}{A^\dagger}\Big)^{\rho_{\ell+2}} \cdot \prod_{i=1}^{\ell} \Big(\frac{g_i^\star}{g_i^\dagger}\Big)^{\rho_i}$$
$$R = \Big(\frac{D^\star}{D^\dagger}\Big) \cdot \Big(\frac{g_z^\star}{g_z^\dagger}\Big)^{\theta_{\ell+1}} \cdot \Big(\frac{A^\star}{A^\dagger}\Big)^{\theta_{\ell+2}} \cdot \prod_{i=1}^{\ell} \Big(\frac{g_i^\star}{g_i^\dagger}\Big)^{\theta_i}$$

satisfies $e(Z, \hat{g}) \cdot e(R, \hat{h}) = 1_{\mathbb{G}_T}$. Moreover, we have $Z \neq 1_{\mathbb{G}}$ with all but negligible probability since $\{\rho_i\}_{i=1}^{\ell}$ are completely independent of $\mathcal{A}$'s view.

We now turn to situation II and show that it implies an algorithm $\mathcal{B}_{II}$ that defeats the strong unforgeability of the partial one-time signature scheme. Algorithm $\mathcal{B}_{II}$ takes as input a POTS verification key $\mathsf{vk}_{pots} = (g_z^\dagger, g_1^\dagger, \ldots, g_\ell^\dagger)$ supplied by its own challenger in the POTS security game. It generates $\mathsf{ck} = \{\hat{X}_i\}_{i=1}^{\ell+2}$ by picking $\rho_i \xleftarrow{R} \mathbb{Z}_p$ and defining $\hat{X}_i = \hat{g}^{\rho_i}$ for each $i \in \{1, \ldots, \ell+2\}$. Letting $Q_c \in \mathsf{poly}(\lambda)$ denote the number of queries made by $\mathcal{A}$ to $\mathcal{O}_{\mathsf{ck}}$, $\mathcal{B}_{II}$ draws a random index $k^\star \xleftarrow{R} \{1, \ldots, Q_c\}$ as a guess that $\mathcal{A}$ will choose to equivocate the commitment $\hat{C}^\dagger$ returned as the output of the $k^\star$-th query. It answers all queries to $\mathcal{O}_{\mathsf{ck}}$ as follows. For each $k \in \{1, \ldots, Q_c\} \backslash \{k^\star\}$, the k-th query is answered by faithfully running the commitment algorithm. When the $k^\star$-th query occurs, $\mathcal{B}_{II}$ embeds $\mathsf{vk}_{pots} = (g_z^\dagger, g_1^\dagger, \ldots, g_\ell^\dagger)$ into the opening of the $k^\star$-th commitment. To this end, it chooses $\zeta \xleftarrow{R} \mathbb{Z}_p^*$ and computes $\hat{C}^\dagger = \hat{g}^\zeta$.

Next, $\mathcal{B}_{II}$ queries its own POTS challenger to obtain a signature $(A^\dagger, (\hat{Z}, \hat{R}))$ on the message $\hat{\boldsymbol{M}} = (\hat{M}_1, \ldots, \hat{M}_\ell) \in \hat{\mathbb{G}}^\ell$ queried by $\mathcal{A}$ at this $k^\star$-th query. Upon receiving a partial one-time signature $(A^\dagger, (\hat{Z}^\dagger, \hat{R}^\dagger))$ from its POTS challenger, $\mathcal{B}_{II}$ defines $(N_1, \ldots, N_\ell, N_{\ell+1}, N_{\ell+2}) = (g_1^\dagger, \ldots, g_\ell^\dagger, g_z^\dagger, A^\dagger)$ and computes

$$D^\dagger = g^\zeta \cdot \prod_{i=1}^{\ell+2} N_i^{-\rho_i} \in \mathbb{G},$$

which satisfies $e(g, \hat{C}^\dagger) = e(D^\dagger, \hat{g}) \cdot \prod_{i=1}^{\ell+2} e(N_i, \hat{X}_i)$. Given that $(A^\dagger, (\hat{Z}^\dagger, \hat{R}^\dagger))$ satisfies the second verification equation of (2) by construction, we observe that

$$\mathsf{open}^\dagger = \big(D^\dagger, g_z^\dagger, g_1^\dagger, \ldots, g_\ell^\dagger, A^\dagger, \hat{Z}^\dagger, \hat{R}^\dagger\big)$$

forms a valid opening of $\hat{C}^\dagger$. When $\mathcal{A}$ halts, we know that, with probability $1/Q_c$, it chooses to output a pair $(\boldsymbol{M}^\star, \mathsf{open}^\star)$ which opens $\hat{C}^\star = \hat{C}^\dagger$. Given that $(D^\star, g_z^\star, g_1^\star, \ldots, g_\ell^\star, A^\star) = (D^\dagger, g_z^\dagger, g_1^\dagger, \ldots, g_\ell^\dagger, A^\dagger)$ and since we must have

$(\boldsymbol{M}^\star, \mathsf{open}^\star) \neq (\boldsymbol{M}^\dagger, \mathsf{open}^\dagger)$ by the definition of ECM-TCR security, we know that $(\boldsymbol{M}^\star, (\hat{Z}^\star, \hat{R}^\star)) \neq (\boldsymbol{M}^\dagger, (\hat{Z}^\dagger, \hat{R}^\dagger))$. This means that $\mathcal{B}_{II}$ can win the game against its POTS challenger by outputting $(\boldsymbol{M}^\star, (A^\star, \hat{Z}^\star, \hat{R}^\star))$. In turn, the result of [4] implies that $\mathcal{B}_{II}$ would contradict the DDH assumption in $\mathbb{G}$. □

3 A Structure-Preserving CCA2-Secure Public-Key Cryptosystem With Shorter Publicly Verifiable Ciphertexts

In this section, we use the all-but-one hash proof systems of [46] and combine them with the structure-preserving commitment scheme of Section 2.5 and a strongly unforgeable signature scheme. We show that the ECMTCR property of the commitment scheme suffices to construct the sought-after CCA2-secure structure preserving encryption scheme with publicly verifiable ciphertexts.

In the notations hereafter, for any vector $\hat{\boldsymbol{h}} = (\hat{h}_1, \hat{h}_2) \in \hat{\mathbb{G}}^2$ and any $g \in \mathbb{G}$, we denote by $E(g, \hat{\boldsymbol{h}})$ the vector $(e(g, \hat{h}_1), e(g, \hat{h}_2))$. For any vectors $\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2 \in \hat{\mathbb{G}}^2$, the product $\hat{\boldsymbol{u}}_1 \cdot \hat{\boldsymbol{u}}_2 \in \hat{\mathbb{G}}^2$ refers to the component-wise multiplication in $\hat{\mathbb{G}}$.

KeyGen(λ): 1. Run the setup algorithm of the commitment scheme in Section 2.5 to obtain $\mathsf{PP} = (p, \mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, e, g, \hat{g}, \ell = 6) \leftarrow TC.\mathsf{Setup}(\lambda, 6)$, which will be used to commit to messages in $\hat{\mathbb{G}}^6$.

2. Generate $(\mathsf{ck}, \mathsf{tk}) \leftarrow TC.\mathsf{KeyGen}(\mathsf{PP})$, where $\mathsf{ck} \in \hat{\mathbb{G}}^8$ is the commitment key and $\mathsf{tk} \in \mathbb{Z}_p^8$ is the trapdoor key which can be erased.
3. Choose also group generators $g_1, g_2 \xleftarrow{R} \mathbb{G}$ and random values $x_1, x_2 \xleftarrow{R} \mathbb{Z}_p$ and set $X = g_1^{x_1} g_2^{x_2}$.
4. Choose $\rho_u \xleftarrow{R} \mathbb{Z}_p$ and $\hat{h} \xleftarrow{R} \hat{\mathbb{G}}^2$ at random.
5. Define $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ with $\hat{\boldsymbol{u}}_1 = (\hat{g}, \hat{h}) \in \hat{\mathbb{G}}^2$ and $\hat{\boldsymbol{u}}_2 = (\hat{g}^{\rho_u}, \hat{h}^{\rho_u}) \in \hat{\mathbb{G}}^2$. Note that $\hat{\boldsymbol{u}}_1$ and $\hat{\boldsymbol{u}}_2$ are linearly dependent.
6. Define $\mathsf{SK} = (x_1, x_2)$ and

$$\mathsf{PK} = (g_1, g_2, \hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, X, \mathsf{PP}, \mathsf{ck}).$$

Encrypt($\boldsymbol{M}$, PK): To encrypt $M \in \mathbb{G}$, conduct the following steps.

1. Generate a key pair $(\mathsf{SSK}, \mathsf{SVK}) \leftarrow OT1.\mathsf{KeyGen}(\mathsf{PP}, 5)$ for the one-time SPS of Section 2.2 so as to sign messages in $\mathbb{G}^5$. Let the resulting key pair consist of $\mathsf{SSK} = \big(\{\chi_i, \gamma_i\}_{i=1}^5, \zeta, \rho\big) \in \mathbb{Z}_p^{14}$ and $\mathsf{SVK} = \big(\{\hat{g}_i\}_{i=1}^5, \hat{A}\big) \in \hat{\mathbb{G}}^6$, where $\hat{g}_i = \hat{g}_z^{\chi_i} \cdot \hat{g}_r^{\delta_i}$ and $\hat{A} = \hat{g}_z^{\zeta} \cdot \hat{g}_r^{\rho}$.
2. Choose $\theta \xleftarrow{R} \mathbb{Z}_p$ and compute

$$C_0 = M \cdot X^\theta, \qquad C_1 = g_1^\theta, \qquad C_2 = g_2^\theta.$$

3. Generate a commitment to $\mathsf{SVK} = (\{\hat{g}_i\}_{i=1}^5, \hat{A})$ and let

$$(\hat{\mathsf{com}}, \mathsf{open}) \leftarrow TC.\mathsf{Commit}(\mathsf{PP}, ck, \mathsf{SVK}) \in \hat{\mathbb{G}} \times (\mathbb{G}^9 \times \hat{\mathbb{G}}^2)$$

be the resulting commitment/opening pair.

4. Define vector $\hat{\boldsymbol{u}}_{\hat{\mathsf{com}}} = \hat{\boldsymbol{u}}_2 \cdot (1, \hat{\mathsf{com}}) \in \hat{\mathbb{G}}^2$ as well as the Groth-Sahai CRS $\hat{\mathbf{u}}_{\hat{\mathsf{com}}} = (\hat{\boldsymbol{u}}_{\hat{\mathsf{com}}}, \hat{\boldsymbol{u}}_1) \in \hat{\mathbb{G}}^2$.
5. Pick $r \xleftarrow{R} \mathbb{Z}_p$. Compute $\hat{\boldsymbol{C}}_\theta = \hat{\boldsymbol{u}}_{\hat{\mathsf{com}}}^{\theta} \cdot \hat{\boldsymbol{u}}_1^{r}$.
6. Using the randomness of the commitment $\boldsymbol{C}_\theta$, generate proof elements $\boldsymbol{\pi} = (\pi_1, \pi_2) = (g_1^r, g_2^r) \in \mathbb{G}^2$ showing that the committed $\theta \in \mathbb{Z}_p$ satisfies the multi-exponentiation equations

$$C_1 = g_1^{\theta} \qquad\qquad C_2 = g_2^{\theta}$$

7. Output the ciphertext

$$\boldsymbol{C} = (\mathsf{SVK}, \hat{\mathsf{com}}, \mathsf{open}, C_0, C_1, C_2, \hat{\boldsymbol{C}}_\theta, \boldsymbol{\pi}, \boldsymbol{\sigma}) \in \mathbb{G}^{16} \times \hat{\mathbb{G}}^{11} \tag{5}$$

where $\boldsymbol{\sigma} \leftarrow OT1.\mathsf{Sign}(\mathsf{SSK}, (C_0, C_1, C_2, \pi_1, \pi_2)) \in \mathbb{G}^2$.

Decrypt($\mathsf{PK}, \boldsymbol{C}, \mathsf{SK}$): Parse the ciphertext $\boldsymbol{C}$ as in (5). Then, conduct the following steps.

1. Parse PK as $(g_1, g_2, X, \mathsf{PP}, \mathsf{ck})$ and SK as (x_1, x_2).
2. Return $\perp$ if $OT1.\mathsf{Verify}(\mathsf{SVK}, (C_0, C_1, C_2, \pi_1, \pi_2), \sigma) = 0$.
3. Return $\perp$ if $\hat{\mathsf{com}} = 1_{\hat{\mathbb{G}}}$ or $TC.\mathsf{Verify}(\mathsf{ck}, \hat{\mathsf{com}}, \mathsf{SVK}, \mathsf{open}) = 0$.
4. Verify that $\boldsymbol{\pi} = (\pi_1, \pi_2)$ is a valid Groth-Sahai proof w.r.t. $(C_1, C_2, \boldsymbol{C}_\theta, \hat{\mathsf{com}})$. Namely, it should satisfy

$$\begin{aligned} E(g_1, \hat{\boldsymbol{C}}_\theta) &= E(C_1, \hat{\boldsymbol{u}}_{\hat{\mathsf{com}}}) \cdot E(\pi_1, \hat{\boldsymbol{u}}_1) \\ E(g_2, \hat{\boldsymbol{C}}_\theta) &= E(C_2, \hat{\boldsymbol{u}}_{\hat{\mathsf{com}}}) \cdot E(\pi_2, \hat{\boldsymbol{u}}_1) \end{aligned} \tag{6}$$

5. If the above verifications all succeed, output $M = C_0/(C_1^{x_1} \cdot C_2^{x_2})$.

Note that, in step 3 of the decryption algorithm, the condition $\hat{\mathsf{com}} \neq 1_{\hat{\mathbb{G}}}$ ensures that vectors $(\hat{\boldsymbol{u}}_{\hat{\mathsf{com}}}, \hat{\boldsymbol{u}}_1)$ form a perfectly sound Groth-Sahai CRS, so that ciphertexts such that $\log_{g_1}(C_1) \neq \log_{g_2}(C_2)$ are always rejected.

The proof of the following theorem follows the strategy of [46] with additional arguments showing that omitting to sign the Groth-Sahai commitments does not affect the security of the scheme.

Theorem 2. *The scheme provides IND-CCA2 security under the SXDH assumption. More precisely,* $\mathbf{Adv}^{\mathsf{CCA}}(\lambda) \leq 5 \times \mathbf{Adv}^{\mathsf{SXDH}}(\lambda) + q_d \times 2^{-\lambda}$.

Proof. The proof proceeds with a sequence of games that begins with the real game and ends with a game where no advantage is left to the adversary whatsoever. In each game, we call W_i the event that the experiment outputs 1. The security parameter λ is implicitly given in all the games. Let q_d denote the number of decryption queries made by the adversary.

Game 0**:** This is the real game. The adversary is given the public key PK which contains vectors $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ such that

$$\hat{\boldsymbol{u}}_1 = (\hat{g}, \hat{h}) \in \hat{\mathbb{G}}^2 \qquad\qquad \hat{\boldsymbol{u}}_2 = (\hat{g}^{\rho_u}, \hat{h}^{\rho_u}) \in \hat{\mathbb{G}}^2, \tag{7}$$

where $\hat{g}, \hat{h} \xleftarrow{R} \hat{\mathbb{G}}$, $\rho_u \xleftarrow{R} \mathbb{Z}_p$. In the challenge phase, it chooses two messages $M_0, M_1 \in \mathbb{G}$ and obtains a challenge ciphertext

$$\boldsymbol{C}^\star = (\mathsf{SVK}^\star, \hat{\mathsf{com}}^\star, \mathsf{open}^\star, C_0^\star, C_1^\star, C_2^\star, \hat{\boldsymbol{C}}_\theta^\star, \boldsymbol{\pi}^\star, \boldsymbol{\sigma}^\star)$$

where, for some random bit $\beta \xleftarrow{R} \{0,1\}$,

$$C_0^\star = M_\beta \cdot X^{\theta^\star}, \qquad\qquad C_1^\star = g_1^{\theta^\star}, \qquad\qquad C_2^\star = g_2^{\theta^\star},$$

as well as $(\hat{\mathsf{com}}, \mathsf{open}) \leftarrow TC.\mathsf{Commit}(\mathsf{PP}_{TC}, ck, \mathsf{SVK})$, $\hat{\boldsymbol{C}}_\theta^\star = \hat{\boldsymbol{u}}_{\hat{\mathsf{com}}^\star}^{\theta^\star} \cdot \hat{\boldsymbol{u}}_1^{r^\star}$ and $\boldsymbol{\pi}^\star = (\pi_1^\star, \pi_2^\star) = (g_1^{r^\star}, g_2^{r^\star})$, where $\hat{\boldsymbol{u}}_{\hat{\mathsf{com}}^\star} = \hat{\boldsymbol{u}}_2 \cdot (1, \hat{\mathsf{com}}^\star)$. We assume w.l.o.g. that $\mathsf{SVK}^\star$ and $\hat{\mathsf{com}}^\star = \hat{C}^\star$ are generated at the outset of the game.
The adversary's decryption queries are always faithfully answered by the challenger. When the adversary halts, it outputs $\beta' \in \{0,1\}$ and wins if $\beta' = \beta$. In this case, the experiment outputs 1. Otherwise, it outputs 0. The adversary's advantage is thus $|\Pr[W_0] - 1/2|$.

Game 1**:** This game is like Game 0 except that, if the adversary makes a pre-challenge decryption query $\boldsymbol{C} = (\mathsf{SVK}, \hat{\mathsf{com}}, \mathsf{open}, C_0, C_1, C_2, \hat{\boldsymbol{C}}_\theta, \boldsymbol{\pi}, \boldsymbol{\sigma})$ such that $\hat{\mathsf{com}} = \hat{\mathsf{com}}^\star$, the experiment halts and outputs a random bit. Since Game 1 is identical to Game 0 until this event F_1 occurs, we have the inequality $|\Pr[W_1] - \Pr[W_0]| \leq \Pr[F_1]$. Moreover, since $\hat{\mathsf{com}}^\star$ was chosen uniformly in $\hat{\mathbb{G}}$ and remains independent of $\mathcal{A}$'s view until the challenge phase, we have $|\Pr[W_1] - \Pr[W_0]| \leq \Pr[F_1] \leq q_d/p$.

Game 2**:** In this game, we modify the generation of the public key and define

$$\begin{aligned} \hat{\boldsymbol{u}}_1 &= (\hat{g}, \hat{h}) \in \hat{\mathbb{G}}^2 \\ \hat{\boldsymbol{u}}_2 &= (\hat{g}^{\rho_u}, \hat{h}^{\rho_u}) \cdot (1, \hat{\mathsf{com}}^\star)^{-1} \in \hat{\mathbb{G}}^2, \end{aligned} \tag{8}$$

for a random $\rho_u \xleftarrow{R} \mathbb{Z}_p$, instead of computing $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ as in (7). Note that $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ are now linearly independent and $\hat{\mathsf{com}}^\star$ is no longer statistically hidden before the challenge phase. However, a straightforward argument based on the semantic security of ElGamal (and thus the DDH assumption in $\hat{\mathbb{G}}$) shows that this modification does not affect the adversary's view. We have $|\Pr[W_2] - \Pr[W_1]| \leq 2 \times \mathbf{Adv}^{\mathrm{DDH}}_{\hat{\mathbb{G}},\mathcal{B}}(\lambda)$.

Game 3**:** This game is like Game 2 but we modify the decryption oracle. Namely, if the adversary makes a post-challenge decryption query for a valid ciphertext $\boldsymbol{C} = (\mathsf{SVK}, \hat{\mathsf{com}}, \mathsf{open}, C_0, C_1, C_2, \hat{\boldsymbol{C}}_\theta, \boldsymbol{\pi}, \boldsymbol{\sigma})$ such that $\hat{\mathsf{com}} = \hat{\mathsf{com}}^\star$ but $(\mathsf{SVK}, \mathsf{open}) \neq (\mathsf{SVK}^\star, \mathsf{open}^\star)$, the experiment halts and outputs a random bit. If we call F_3 the latter event, we have $|\Pr[W_3] - \Pr[W_2]| \leq \Pr[F_3]$. As

shown by Lemma 1, event F_3 implies an adversary $\mathcal{B}_3$ against the ECM-TCR property (as formalized by Definition 8) of the trapdoor commitment in Section 2.5, which contradicts the SXDH assumption. We thus have $|\Pr[W_3] - \Pr[W_2]| \leq \mathbf{Adv}^{\mathsf{ECM\text{-}TCR}}_{TC,\mathcal{B}_3}(\lambda) \leq \mathbf{Adv}^{\mathrm{SXDH}}_{\mathcal{B}_3}(\lambda)$.

Game 4**:** We modify again the decryption oracle in post-challenge decryption queries. After the challenge phase, if the adversary $\mathcal{A}$ queries the decryption of a ciphertext $\boldsymbol{C} = (\mathsf{SVK}, \hat{\mathsf{com}}, \mathsf{open}, C_0, C_1, C_2, \hat{\boldsymbol{C}}_\theta, \boldsymbol{\pi}, \boldsymbol{\sigma})$ such that we have $(\hat{\mathsf{com}}, \mathsf{open}) = (\hat{\mathsf{com}}^\star, \mathsf{open}^\star)$ but $(C_0, C_1, C_2, \pi_1, \pi_2) \neq (C_0^\star, C_1^\star, C_2^\star, \pi_1^\star, \pi_2^\star)$, the experiment halts and outputs a random bit. If we call F_4 this event, we have the inequality $|\Pr[W_4] - \Pr[W_3]| \leq \Pr[F_4]$ since Game 4 is identical to Game 3 until F_4 occurs. Moreover, F_4 would contradict the strong unforgeability of the one-time structure-preserving signature and thus the DP assumption. This implies $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}^{\mathsf{SUF\text{-}OTS}}_{\mathcal{B}}(\lambda) \leq \mathbf{Adv}^{\mathrm{DP}}_{\mathcal{B}}(\lambda)$.

Game 5**:** We introduce another modification in the decryption oracle. We reject all ciphertexts $\boldsymbol{C} = (\mathsf{SVK}, \hat{\mathsf{com}}, \mathsf{open}, C_0, C_1, C_2, \hat{\boldsymbol{C}}_\theta, \boldsymbol{\pi}, \boldsymbol{\sigma})$ such that

$$(\hat{\mathsf{com}}, \mathsf{open}) = (\hat{\mathsf{com}}^\star, \mathsf{open}^\star) \quad \wedge \\ (C_0, C_1, C_2, \pi_1, \pi_2) = (C_0^\star, C_1^\star, C_2^\star, \pi_1^\star, \pi_2^\star) \quad \wedge \quad \hat{\boldsymbol{C}}_\theta \neq \hat{\boldsymbol{C}}^\star_\theta. \quad (9)$$

Let F_5 be the event that the decryption oracle rejects a ciphertext that would not have been rejected in Game 4. We argue that $\Pr[W_5] = \Pr[W_4]$ since Game 5 is identical to Game 4 until event F_5 occurs and we have $\Pr[F_5] = 0$. Indeed, for a given $(C_1^\star, C_2^\star, \pi_1^\star, \pi_2^\star) \in \mathbb{G}^4$, there exists only one commitment $\hat{\boldsymbol{C}}^\star_\theta \in \hat{\mathbb{G}}^2$ that satisfies the equalities (6). This follows from the fact that, since $(C_1^\star, C_2^\star, \pi_1^\star, \pi_2^\star) = (g_1^{\theta^\star}, g_2^{\theta^\star}, g_1^{r^\star}, g_2^{r^\star})$, relations (6) can be written

$$E(g_1, \hat{\boldsymbol{C}}^\star_\theta) = E(g_1^{\theta^\star}, \hat{\boldsymbol{u}}_{\hat{\mathsf{com}}}) \cdot E(g_1^{r^\star}, \hat{\boldsymbol{u}}_1) = E(g_1, \hat{\boldsymbol{u}}_{\hat{\mathsf{com}}}^{\theta^\star}) \cdot E(g_1, \hat{\boldsymbol{u}}_1^{r^\star}) \\ E(g_2, \hat{\boldsymbol{C}}^\star_\theta) = E(g_2^{\theta^\star}, \hat{\boldsymbol{u}}_{\hat{\mathsf{com}}}) \cdot E(g_2^{r^\star}, \hat{\boldsymbol{u}}_1) = E(g_2, \hat{\boldsymbol{u}}_{\hat{\mathsf{com}}}^{\theta^\star}) \cdot E(g_2, \hat{\boldsymbol{u}}_1^{r^\star})$$

which uniquely determines the only commitment $\hat{\boldsymbol{C}}^\star_\theta = \hat{\boldsymbol{u}}_{\hat{\mathsf{com}}}^{\theta^\star} \cdot \hat{\boldsymbol{u}}_1^{r^\star} \in \hat{\mathbb{G}}^2$ that satisfies (6). This shows that $\Pr[F_5] = 0$, as claimed.

Game 6**:** In this game, we modify the distribution of the public key. Namely, instead of generating the vectors $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ as in (8), we set

$$\hat{\boldsymbol{u}}_1 = (\hat{g}, \hat{h}) \in \hat{\mathbb{G}}^2 \qquad \hat{\boldsymbol{u}}_2 = (\hat{g}^{\rho_u}, \hat{h}^{\rho_u}) \cdot (1, \hat{C}^{\star -1}) \in \hat{\mathbb{G}}^2. \quad (10)$$

Said otherwise, $\hat{\boldsymbol{u}}_2$ is now the product of two terms, the first one of which lives in the one-dimensional subspace spanned by $\hat{\boldsymbol{u}}_1$. Under the DDH assumption in $\hat{\mathbb{G}}$, this modified distribution of PK should have not noticeable impact on the adversary's behavior. A straightforward reduction shows that $|\Pr[W_6] - \Pr[W_5]| \leq \mathbf{Adv}^{\mathrm{DDH}}_{\mathcal{B}}(\lambda)$. Note that, although the vectors $(\hat{\boldsymbol{u}}_{\hat{\mathsf{com}}^\star}, \hat{\boldsymbol{u}}_1) \in \hat{\mathbb{G}}^2$ are no longer linearly independent, $\hat{\boldsymbol{C}}^\star_\theta = \hat{\boldsymbol{u}}_1^{\rho_u \cdot \theta^\star + r^\star}$ remains the only commitment that satisfies the verification equations for a given tuple $(C_1^\star, C_2^\star, \pi_1^\star, \pi_2^\star)$.

Game 7**:** In this game, we modify the challenge ciphertext and replace the NIZK proof $\boldsymbol{\pi}^\star = (\pi_1^\star, \pi_2^\star) \in \mathbb{G}^2$ by a simulated proof which is produced using $\rho_u \in \mathbb{Z}_p$ as a simulation trapdoor. Namely, $(\hat{\boldsymbol{C}}_\theta^\star, \boldsymbol{\pi}^\star)$ is obtained by picking $r \xleftarrow{R} \mathbb{Z}_p$ and computing

$$\hat{\boldsymbol{C}}_\theta^\star = \boldsymbol{u}_1^r, \qquad \pi_1^\star = g_1^r \cdot C_1^{\star -\rho_u}, \qquad \pi_2^\star = g_2^r \cdot C_2^{\star -\rho_u}$$

Observe that, although $(\hat{\boldsymbol{C}}_\theta^\star, \pi_1^\star, \pi_2^\star)$ are generated without using the witness $\theta^\star = \log_{g_1}(C_1^\star) = \log_{g_2}(C_2^\star)$, the NIZK property of GS proofs ensures that their distribution remains exactly as in Game 6: indeed, if we define $\tilde{r} = r - \rho_u \cdot \theta^\star$, we have

$$\hat{\boldsymbol{C}}_\theta^\star = \hat{\boldsymbol{u}}_{\mathsf{com}^\star}^{\theta^\star} \cdot \hat{\boldsymbol{u}}_1^{\tilde{r}}, \qquad \pi_1^\star = g_1^{\tilde{r}}, \qquad \pi_2^\star = g_2^{\tilde{r}},$$

which implies $\Pr[W_7] = \Pr[W_6]$.

Game 8**:** We modify the generation of the challenge ciphertext, which is generated using the private key $\mathsf{SK} = (x_1, x_2)$ instead of the public key: Namely, the challenger computes

$$C_1^\star = g_1^{\theta^\star}, \qquad C_2^\star = g_2^{\theta^\star}, \qquad C_0^\star = M_\beta \cdot C_1^{\star x_1} \cdot C_2^{\star x_2},$$

while $(\hat{\boldsymbol{C}}_\theta^\star, \pi_1^\star, \pi_2^\star)$ are computed using the NIZK simulation trapdoor $\rho_u \in \mathbb{Z}_p$ as in Game 7. This change does not affect the adversary's view since the ciphertext retains the same distribution. We have $\Pr[W_8] = \Pr[W_7]$.

Game 9**:** We modify again the distribution of the challenge ciphertext which is obtained as

$$C_1^\star = g_1^{\theta_1^\star}, \qquad C_2^\star = g_2^{\theta_2^\star}, \qquad C_0^\star = M_\beta \cdot C_1^{\star x_1} \cdot C_2^{\star x_2},$$

for random and independent $\theta_1^\star, \theta_2^\star \xleftarrow{R} \mathbb{Z}_p$, while the NIZK proof $(\hat{\boldsymbol{C}}_\theta^\star, \pi_1^\star, \pi_2^\star)$ is simulated using $\rho_u \in \mathbb{Z}_p$ as in Game 8. Since the witness $\theta^\star \in \mathbb{Z}_p$ was not used anymore in Game 8, a straightforward reduction shows that any noticeable change in $\mathcal{A}$'s output distribution implies a DDH distinguisher in $\mathbb{G}$. We have $|\Pr[W_9] - \Pr[W_8]| \leq \mathbf{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathbb{G}}(\lambda)$.

In the final game, it is easy to see that $\Pr[W_9] = 1/2$ since the challenge ciphertext does not carry any information about $\beta \in \{0,1\}$. Indeed, we have

$$C_1^\star = g_1^{\theta_1^\star}, \qquad C_2^\star = g_2^{\theta_1^\star + \theta_1'}, \qquad C_0^\star = M_\beta \cdot X^{\theta_1^\star} \cdot {g_2}^{\theta_1' \cdot x_2},$$

for some random $\theta_1' \in_R \mathbb{Z}_p$, which implies that the term ${g_2}^{\theta_1' \cdot x_2}$ perfectly hides M_β in the expression of $C_0^\star$. This follows from the fact that $x_2 \in \mathbb{Z}_p$ is perfectly independent of the adversary's view. Indeed, the public key leaves $x_2 \in \mathbb{Z}_p$ completely undetermined as it only reveals $X = g_1^{x_1} g_2^{x_2}$. During the game, decryption queries are guaranteed not to reveal anything about x_2 since all NIZK proofs

$(\hat{\boldsymbol{C}}_\theta, \pi_1, \pi_2)$ take place on Groth-Sahai CRSes $(\hat{\boldsymbol{u}}_{\hat{\mathsf{com}}}, \hat{\boldsymbol{u}}_1)$ which are perfectly sound (as they span the entire vector space $\hat{\mathbb{G}}^2$) whenever $\hat{\mathsf{com}} \neq \hat{\mathsf{com}}^\star$. This implies that, although the adversary can see a simulated NIZK proof $(\hat{\boldsymbol{C}}^\star_\theta, \pi^\star_1, \pi^\star_2)$ for a false statement in the challenge phase, it remains unable to trick the decryption oracle into accepting a ciphertext $\boldsymbol{C} = (\mathsf{SVK}, \hat{\mathsf{com}}, \mathsf{open}, C_0, C_1, C_2, \hat{\boldsymbol{C}}_\theta, \boldsymbol{\pi}, \boldsymbol{\sigma})$ such that $\log_{g_1}(C_1) \neq \log_{g_2}(C_2)$. As a consequence, the adversary does not learn anything about x_2 from responses of the decryption oracle. □

Lemma 1. *In Game 3, there exists an ECM-TCR adversary with advantage $\epsilon \geq \Pr[F_3]$ against the trapdoor commitment scheme of Section 2.5 and which runs in about the same time as $\mathcal{A}$.*

Proof. Let $\mathcal{A}$ be an adversary against the SP-CCA encryption scheme as in the proof of Theorem 2 and let the event F_3 be defined as in Game 3. Then, we build an adversary $\mathcal{B}_3$ against the ECM-CTR security of the structure-preserving trapdoor commitment defined in Section 2.5 which efficiently runs $\mathcal{A}$.

The challenger $\mathcal{B}_3$ is given the public parameter PP_{TC} and a commitment key ck generated as in the trapdoor commitment scheme as well as an access to a commit-open oracle $\mathcal{O}_{\mathsf{ck}}$ as defined in Definition 8. Then, $\mathcal{B}_3$ runs step 3 to step 6 of the key generation algorithm of the encryption scheme to get PK and $\mathsf{SK} = (x_1, x_2)$ as specified in Game 2 and Game 3.

The adversary $\mathcal{A}$ is given PK and $\mathcal{B}_3$ is easily able to answer to $\mathcal{A}$'s decryption queries as described in Game 2 and Game 3 thanks to SK. In order to compute the challenge ciphertext given $\{m_0, m_1\}$, $\mathcal{B}_3$ picks $\beta \xleftarrow{R} \{0,1\}$, runs all the steps of the encryption algorithm with m_β except for step 3 for which $\mathcal{B}_3$ queries $\mathcal{O}_{\mathsf{ck}}$ on $\mathsf{SVK}^\star$ to get $(\hat{\mathsf{com}}^\star, \mathsf{open}^\star)$. The computed ciphertext $\boldsymbol{C}^\star$ is then given to $\mathcal{A}$.

Assuming that F_3 occurs, which means that $\mathcal{A}$ makes a post-challenge decryption query for a valid ciphertext $\boldsymbol{C} = (\mathsf{SVK}, \hat{\mathsf{com}}, \mathsf{open}, C_0, C_1, C_2, \hat{\boldsymbol{C}}_\theta, \boldsymbol{\pi}, \boldsymbol{\sigma})$ such that $\hat{\mathsf{com}} = \hat{\mathsf{com}}^\star$ but $(\mathsf{SVK}, \mathsf{open}) \neq (\mathsf{SVK}^\star, \mathsf{open}^\star)$, the challenger simply outputs $(\hat{\mathsf{com}}^\star, \mathsf{SVK}, \mathsf{open})$.

Obviously, we have $TC.\mathsf{Verify}(\mathsf{ck}, \hat{\mathsf{com}}^\star, \mathsf{SVK}, \mathsf{open}) = 1$ since $\boldsymbol{C}$ is valid. However, during the ECM-TR experiment $\mathcal{B}_3$ only chose a single message $\mathsf{SVK}^\star$ so that there is only one target in $Q = \{(\hat{\mathsf{com}}^\star, \mathsf{SVK}^\star, \mathsf{open}^\star)\}$. Moreover, since we also have $(\hat{\mathsf{com}}^\star, \mathsf{SVK}, \mathsf{open}) \notin Q$, we find $\Pr[F_3] = \mathbf{Adv}^{ECM\text{-}TCR}_{TC,\mathcal{B}_3}(\lambda)$. □

While we do not explicit provide a threshold decryption mechanism in the paper, this can be easily achieved in the same way as in the SXDH-based threshold cryptosystem described in [46]. As a result, we readily obtain a robust and non-interactive structure-preserving threshold cryptosystem with CCA2-security in the adaptive corruption setting.

It would be interesting to improve the efficiency of the scheme using quasi-adaptive NIZK arguments [38] in the same way as in [43]. Unfortunately, we did not manage to obtain the required simulation-soundness property while keeping the QA-NIZK arguments structure-preserving.

4 A Randomizable RCCA-Secure Construction

Given a message M over $\mathbb{G}$, the encryption algorithm computes an ElGamal-like encryption of the form $(c_0, c_1, c_2) = (f^\theta, g^\theta, M \cdot h^\theta)$. In order to have an alternative decryption in the reduction as well as publicly verifiable ciphertexts, the algorithm then derives an LHSP signature (Section 2.4) on the vector $\boldsymbol{v} = (c_0^b, c_1^b, g^{1-b}, c_1^{1-b}, c_2^{1-b})$, where $b = 1$ is a hidden bit. This is made possible by giving an LHSP signature on $\boldsymbol{v}_1 = (f, g, 1, 1, 1)$ and $\boldsymbol{v}_2 = (1, 1, 1, g, h)$ in the public key since $\boldsymbol{v} = \boldsymbol{v}_1^\theta$. Note that, if $b = 0$, the encryption algorithm cannot derive a signature on $\boldsymbol{v}$ since $(1, 1, g, c_1, c_2)$ is outside the linear span of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. The goal of the security reduction is to compute the challenge ciphertext with $b = 0$ (using the signing key) and force the adversary to keep this $b = 0$ in any re-randomization of the challenge. This allows detecting when the adversary attempts to obtain the decryption of a *replayed* ciphertext.

In order to make freshly generated ciphertexts indistinguishable from (re-randomizations of) the challenge ciphertext, we use Groth-Sahai commitments and NIWI proofs to hide b. The encryption algorithm computes a commitment to g^b and $\boldsymbol{v}$ and proves that $b \in \{0, 1\}$ and that $\boldsymbol{v}$ is well-formed with respect to (c_0, c_1, c_2). Then, it proves that the LHSP signature on $\boldsymbol{v}$ is valid.

This proof can be seen as a quasi-adaptive NIZK proof [38] (defined in Appendix B.2) that either (c_0, c_1, c_2) is well-formed or that I know a one-time signature on (c_1, c_2) (of Section 2.2) which corresponds to an LHSP signature on (g, c_1, c_2), where g is the fixed element of the verification-key.

In order to statistically re-randomize ciphertext, the OR-proof should be efficiently and publicly adaptable and at the same time it should not support any other kind of malleability. Even though in the NIWI setting the Groth-Sahai proofs are perfectly re-randomizable the constants of the proofs are modified when we compute $(c_0', c_1', c_2') = (c_0, c_1, c_2) \cdot (f, g, h)^{\theta'}$ as well as the variables $\boldsymbol{v}' = \boldsymbol{v} \cdot (\boldsymbol{v}_1^b \cdot \boldsymbol{v}_2^{1-b})^{\theta'}$. Since proving that $\boldsymbol{v}'$ has the correct form requires the same random coins as those used in the commitment of g^b, the encryption algorithm simply adds in the ciphertext a commitment to $\boldsymbol{v}_1^b \cdot \boldsymbol{v}_2^{1-b}$, a proof of well-formedness and a Groth-Sahai NIWI proof of an LHSP signature that can be derived from the public key.

At a first glance, ciphertexts may appear not to prevent malleability of the encrypted message M since nothing seems to "freeze" c_2 in the ciphertext when $c_2^{1-b} = 1$ in honest execution. However, the ciphertext actually binds c_2 in the proof elements which depend on the random coins of the commitments.

Keygen(λ): Choose bilinear groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with generators $f, g \xleftarrow{R} \mathbb{G}$, $\hat{g}, \hat{h} \xleftarrow{R} \hat{\mathbb{G}}$ and do the following.

1. Choose a random exponent $\alpha \xleftarrow{R} \mathbb{Z}_p$ and set $h = g^\alpha$.
2. Choose random $\boldsymbol{u}_1, \boldsymbol{u}_2 \xleftarrow{R} \mathbb{G}^2$ and $\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2 \xleftarrow{R} \hat{\mathbb{G}}^2$.
3. Define $\boldsymbol{v}_1 = (f, g, 1, 1, 1)$ and $\boldsymbol{v}_2 = (1, 1, 1, g, h)$, then generate a crs for a QA-NIZK proof system for the language of vectors in $\mathsf{span}\langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle$: pick $\mathsf{tk} = (\chi_j, \gamma_j)_{j=1}^5 \xleftarrow{R} \mathbb{Z}_p^{2\times 5}$ and compute $\hat{g}_j = \hat{g}^{\chi_j} \hat{h}^{\gamma_j}$, for each $1 \le j \le 5$,

as well as the language dependent parameters $(z_1, r_1) = (f^{\chi_1} g^{\chi_2}, f^{\gamma_1} g^{\gamma_2})$ and $(z_2, r_2) = (g^{\chi_4} h^{\chi_5}, g^{\gamma_4} h^{\gamma_5})$. Then, we have

$$e(z_1, \hat{g}) \cdot e(r_1, \hat{h}) = (f, \hat{g}_1) \cdot (g, \hat{g}_2),$$
$$e(z_2, \hat{g}) \cdot e(r_2, \hat{h}) = (g, \hat{g}_4) \cdot (h, \hat{g}_5).$$

4. Define the private key as $\mathsf{SK} = \alpha \in \mathbb{Z}_p$ and erase tk. The public key $\mathsf{PK} \in \mathbb{G}^{11} \times \hat{\mathbb{G}}^{16}$ is defined to be

$$\mathsf{PK} = \big(f,\ g,\ h,\ \boldsymbol{u}_1,\ \boldsymbol{u}_2,\ z_1,\ r_1,\ r_2,\ z_2,\ \hat{g},\ \hat{h},\ \hat{\boldsymbol{u}}_1,\ \hat{\boldsymbol{u}}_2,\ \{\hat{g}_j\}_{j=1}^{5}\big).$$

Encrypt(PK, M): To encrypt $M \in \mathbb{G}$, conduct the following steps:

1. Pick $\theta \xleftarrow{R} \mathbb{Z}_p$ and compute $(c_0, c_1, c_2) = (f^\theta, g^\theta, M \cdot h^\theta)$.
2. Define the bit $b = 1$ and set $G = g^b \in \mathbb{G}$ and $\hat{g}^b \in \hat{\mathbb{G}}$. Prove that

$$e(\boxed{G}, \hat{g}) = e(g, \boxed{\hat{g}^b}) \qquad e(\boxed{G}, \hat{g}/\boxed{\hat{g}^b}) = 1_{\mathbb{G}_T}. \tag{11}$$

Namely, compute commitments to $G = g^b$ (resp. $\hat{g}^b$), which are obtained as $C_G = (1, G) \cdot \boldsymbol{u}_1^{r_g} \cdot \boldsymbol{u}_2^{s_g}$ (resp. $\hat{C}_b = (1, \hat{g}^b) \cdot \hat{\boldsymbol{u}}_1^{r_b} \cdot \hat{\boldsymbol{u}}_2^{s_b}$), for random $r_g, s_g, r_b, s_b \xleftarrow{R} \mathbb{Z}_p$. Let $\pi_G \in \mathbb{G}^2 \times \hat{\mathbb{G}}^2$ and $\pi_{bit} \in \mathbb{G}^4 \times \hat{\mathbb{G}}^4$ be the proof elements for relations (11).

3. Define $(\Theta_0, \Theta_1, \Theta_2) = (c_0^b, c_1^b, c_2^b)$ and prove that[7]

$$e(\boxed{\Theta_1}, \hat{g}) = e(c_1, \boxed{\hat{g}^b}) \qquad e(\boxed{\Theta_2}, \hat{g}) = e(c_2, \boxed{\hat{g}^b}). \tag{12}$$

More precisely, compute commitments to Θ_i as $C_i = (1, \Theta_i) \cdot \boldsymbol{u}_1^{\bar{r}_i} \cdot \boldsymbol{u}_2^{\bar{s}_i}$, for each $i \in \{0, 1, 2\}$, and for random $\bar{r}_i, \bar{s}_i \xleftarrow{R} \mathbb{Z}_p$. The corresponding proof elements π_1, π_2 both live in $\mathbb{G}^2 \times \hat{\mathbb{G}}^2$.

4. Derive a QA-NIZK proof $(z, r) = (z_1^\theta, r_1^\theta)$ that $\boldsymbol{v} := \boldsymbol{v}_1^\theta \in \mathbb{G}^5$ belongs to $\mathsf{span}\langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle$. Since $b = 1$, we have

$$\boldsymbol{v} = (\boldsymbol{v}_1^\theta)^b \cdot (\boldsymbol{v}_2^\theta)^{1-b} = (c_0^b, c_1^b, 1, 1, 1) = (c_0^b, c_1^b, g^{1-b}, c_1^{1-b}, c_2^{1-b}),$$

which allows generating a NIWI proof $\pi_{enc} \in \hat{\mathbb{G}}^2$ that $(z, r, \Theta_0, \Theta_1, \Theta_2, g^b)$ satisfy

$$\begin{aligned} e(\boxed{z}, \hat{g}) \cdot e(\boxed{r}, \hat{h}) = e(\boxed{\Theta_0}, \hat{g}_1) \cdot e(\boxed{\Theta_1}, \hat{g}_2) \cdot e(g/\boxed{g^b}, \hat{g}_3) \\ \cdot e(c_1/\boxed{\Theta_1}, \hat{g}_4) \cdot e(c_2/\boxed{\Theta_2}, \hat{g}_5). \end{aligned} \tag{13}$$

together with the Groth-Sahai commitments $C_z, C_r \in \mathbb{G}^2$ of $z, r \in \mathbb{G}$.

[7] Note that we intentionally omit to prove the validity of Θ_0 as the unforgeability of the LHSP signature is sufficient for this purpose. As a consequence, c_0 does not have to be in the ciphertext.

5. To enable re-randomization, define $H = h^b$ and $F = f^b$ and compute Groth-Sahai commitments to H and F as $C_H = (1, h^b) \cdot \boldsymbol{u_1}^{r_h} \cdot \boldsymbol{u_2}^{s_h} \in \mathbb{G}^2$ and $C_F = (1, f^b) \cdot \boldsymbol{u_1}^{r_f} \cdot \boldsymbol{u_2}^{s_f} \in \mathbb{G}^2$ for random $r_h, r_f, s_h, s_f \xleftarrow{R} \mathbb{Z}_p$. Then, generate a NIWI proof $\pi_H \in \mathbb{G}^2 \times \hat{\mathbb{G}}^2$ that

$$e(\boxed{H}, \hat{g}) = e(h, \boxed{\hat{g}^b}).$$

6. Derive a QA-NIZK argument $(z_{rand}, r_{rand}) = (z_1^b \cdot z_2^{1-b}, r_1^b \cdot r_2^{1-b})$ that $\boldsymbol{w} := v_1^b \cdot v_2^{1-b}$ belongs to $\mathsf{span}\langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle$. Since $\boldsymbol{w} = (f^b, g^b, 1, g^{1-b}, h^{1-b})$, generate a proof $\pi_{rand} \in \hat{\mathbb{G}}^2$ that

$$\begin{aligned} &e(\boxed{z_{rand}}, \hat{g}) \cdot e(\boxed{r_{rand}}, \hat{h}) \\ &\quad = e(\boxed{F}, \hat{g}_1) \cdot e(\boxed{G}, \hat{g}_2) \cdot e(g/\boxed{G}, \hat{g}_4) \cdot e(h/\boxed{H}, \hat{g}_5), \end{aligned}$$

together with the commitments $C_{z_{rand}}, C_{r_{rand}} \in \mathbb{G}^2$.

Return the ciphertext $\boldsymbol{c} = (c_1, c_2, \pi_{\mathsf{Enc}}, \pi_{\mathsf{Rand}})$ of $\mathbb{G}^{34} \times \hat{\mathbb{G}}^{18}$ where,

$$\begin{aligned} \pi_{\mathsf{Enc}} &= (C_G, \hat{C}_b, \pi_G, \pi_{bit}, C_0, C_1, C_2, \pi_1, \pi_2, C_z, C_r, \pi_{enc}), \\ \pi_{\mathsf{Rand}} &= (C_H, \pi_H, C_F, C_{z_{rand}}, C_{r_{rand}}, \pi_{rand}). \end{aligned}$$

ReRand(PK, $\boldsymbol{c}$): Parse $\boldsymbol{c} = (c_1, c_2, \pi_{\mathsf{Enc}}, \pi_{\mathsf{Rand}})$ as above and do the following:

1. Pick $\theta' \xleftarrow{R} \mathbb{Z}_p$ and compute $(c_1', c_2') = (c_1 \cdot g^{\theta'}, c_2 \cdot h^{\theta'})$.
2. Update[8] the commitments C_0, C_1, C_2 and the proofs π_1, π_2 of relations (12) according to the update of the constants c_1, c_2 into c_1', c_2'. Namely, compute $(C_0', C_1', C_2') = (C_0 \cdot C_F^{\theta'}, C_1 \cdot C_G^{\theta'}, C_2 \cdot C_H^{\theta'})$ as well as $\pi_1' = \pi_1 \cdot \pi_G^{\theta'}$ and $\pi_2' = \pi_2 \cdot \pi_H^{\theta'}$.
3. Update[9] C_z, C_r and the NIWI proof π_{enc} for relation (13). Namely, compute $C_z' = C_z \cdot C_{z_{rand}}^{\theta'}$ and $C_r' = C_r \cdot C_{r_{rand}}^{\theta'}$ as well as $\pi_{enc}' = \pi_{enc} \cdot \pi_{rand}^{\theta'}$. We should have

$$\Theta_0' = f^{b \cdot (\theta + \theta')}, \qquad \Theta_1' = g^{b \cdot (\theta + \theta')}, \qquad \Theta_2' = M^b \cdot h^{b \cdot (\theta + \theta')},$$

while C_z' and C_r' are now commitments to

$$\begin{aligned} z' &= z \cdot z_{rand}^{\theta'} = (z_1^b \cdot z_2^{1-b})^{\theta + \theta'} \\ r' &= r \cdot r_{rand}^{\theta'} = (r_1^b \cdot r_2^{1-b})^{\theta + \theta'}. \end{aligned}$$

4. Re-randomize $C_G, \hat{C}_b, C_0', C_1', C_2', C_z', C_r', C_H, C_F, C_{z_{rand}}, C_{r_{rand}}$ and the proofs $\pi_G, \pi_{bit}, \pi_1', \pi_2', \pi_{enc}', \pi_H, \pi_{rand}$ so as to get $C_G'', \hat{C}_b'', C_0'', C_1'', C_2'', C_z'', C_r'', C_H'', C_F'', C_{z_{rand}}'', C_{r_{rand}}''$ and $\pi_G'', \pi_{bit}'', \pi_1'', \pi_2'', \pi_{enc}'', \pi_H'', \pi_{rand}''$.

[8] This is can be done efficiently because $\boldsymbol{c}$ contains the commitments and the proofs $C_G, \pi_G \in \pi_{\mathsf{Enc}}$ and $C_H, \pi_H, C_F \in \pi_{\mathsf{Rand}}$ for which π_G, π_H should not only be associated to the bit b but should also contain the same random coins of $\hat{C}_b$ used in π_1, π_2.

[9] At this point, $\{C_i'\}_{i=0,1,2}$ are no longer commitments to $\{\Theta_i\}_{i=0,1,2}$ since the variables have changed into $\Theta_0' = \Theta_0 \cdot F^{\theta'}$, $\Theta_1' = \Theta_1 \cdot G^{\theta'}$ and $\Theta_2' = \Theta_2 \cdot H^{\theta'}$.

Return the ciphertext $\boldsymbol{c'} = (c'_1, c'_2, \pi'_{\mathsf{Enc}}, \pi'_{\mathsf{Rand}})$ where,

$$\pi'_{\mathsf{Enc}} = (C''_G, \hat{C}''_b, \pi''_G, \pi''_{bit}, C''_0, C''_1, C''_2, \pi''_1, \pi''_2, C''_z, C''_r, \pi''_{enc}),$$
$$\pi'_{\mathsf{Rand}} = (C''_H, \pi''_H, C''_F, C''_{z_{rand}}, C''_{r_{rand}}, \pi''_{rand}).$$

Decrypt($\mathsf{SK}, \boldsymbol{c}$): Parse $\boldsymbol{c} = (c_1, c_2, \pi_{\mathsf{Enc}}, \pi_{\mathsf{Rand}})$ as above and check whether all the proofs are valid. If not, output $\perp$, and otherwise return $M = c_1/c_2^{\alpha}$.

As far as efficiency goes, ciphertexts consist of 34 elements of $\mathbb{G}$ and 18 elements of $\hat{\mathbb{G}}$. Correctness follows from the correctness of the Groth-Sahai proofs and the correctness of the underlying LHSP signatures.

We show that the above re-randomizable encryption scheme, denoted by $\mathcal{E}$, is statistically re-randomizable even for adversarially chosen ciphertext as defined in Definition 14 under the name of statistical unlikability [47].

Theorem 3. *The above scheme $\mathcal{E}$ provides statistical unlinkability.*

Proof. We only consider valid adversarially-generated ciphertext $\boldsymbol{c}$ since the validity of ciphertext is efficiently recognizable. Given $\boldsymbol{c} \leftarrow \mathcal{A}(\mathsf{PK})$, we define two distributions on ciphertexts as in the definition of unlikability. The first distribution generates $\boldsymbol{c'} \leftarrow \mathsf{Encrypt}(\mathsf{PK}, \mathsf{Decrypt}(\mathsf{SK}, \boldsymbol{c}))$ while the second distribution generates $\boldsymbol{c'} \leftarrow \mathsf{ReRand}(\mathsf{PK}, \boldsymbol{c})$. Clearly if we write $\boldsymbol{c'} = (c'_1, c'_2, \pi'_{\mathsf{Enc}}, \pi'_{\mathsf{Rand}})$, the first distribution generates (c_1, c_2) as a fresh ElGamal ciphertext and the perfectly NIWI proofs $(\pi'_{\mathsf{Enc}}, \pi'_{\mathsf{Rand}})$ are completely random subject to the verification of all the pairing product equations detailed in the encryption algorithm of $\mathcal{E}$. Indeed, the key generation algorithm sets the CRSes $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ as random elements as in the perfect NIWI setting of the Groth-Sahai proof system [36]. For the same reason, ReRand transforms $\boldsymbol{c}$ into a perfectly re-randomized ciphertext $\boldsymbol{c'}$. Indeed, step 1 leads to a perfectly re-randomized ElGamal ciphertext $(c'_1, c'_2) = (c_1, c_2) \cdot (g, h)^{\theta'}$. Steps 2 and 3 adapt the Groth-Sahai commitments and proofs with respect to the constant (c'_1, c'_2) to keep the validity of the ciphertext. Finally, step 4 completely re-randomizes these commitments and proofs and the NIWI setting ensures that the resulting $(\pi'_{\mathsf{Enc}}, \pi'_{\mathsf{Rand}})$ are uniformly re-distributed among all the valid proofs satisfying the same pairing product equations with the constant (c'_1, c'_2). Consequently, $\boldsymbol{c'}$ is distributed as a fresh ciphertext of $\mathsf{Decrypt}(\mathsf{SK}, \boldsymbol{c})$ even if the adversary tried to put some subliminal information in $\boldsymbol{c}$. □

Next, we show that $\mathcal{E}$ is secure against a Replayable Chosen-Ciphertext Attack (RCCA) in the sense of Definition 13.

Theorem 4. *The above scheme $\mathcal{E}$ provides RCCA security under the SXDH assumption. More precisely, we have* $\mathbf{Adv}^{\mathrm{RCCA}}_{\mathcal{A},\mathcal{E}}(\lambda) \leq 4 \cdot \mathbf{Adv}^{\mathrm{SXDH}}(\lambda) + q_d \cdot 2^{-\lambda}$.

Proof. The proof uses a sequence of games starting with the real game and ending with a game where even an unbounded adversary has no advantage. For each i, S_i is the event that the challenger outputs 1 in Game i meaning that the adversary rightly guesses which message is encrypted in the challenge ciphertext. We assume that security parameter λ is given in each game.

Game 1: This is the real attack game where the adversary chooses M_0 and M_1 and obtains a challenge ciphertext $\boldsymbol{c}^\star$ as a real encryption of M_β, for some $\beta \xleftarrow{R} \{0,1\}$ chosen by the challenger, in the challenge phase. We recall that the adversary may query the decryption of any ciphertext. In the post-challenge phase, when the challenger uses SK to faithfully reply to the decryption queries it runs the decryption algorithm and returns $\perp$ if the (public) verification fails. If the decryption returns M, the challenger sends back M except if $M \in \{M_0, M_1\}$, in which case "*replay*" is returned. We denote by S_1 the event that the adversary outputs $\beta' = \beta$, which causes the challenger to output 1.

Game 2: This game is like Game 1 except that, in the challenge phase, the challenge ciphertext $\boldsymbol{c}^\star = (c_1^\star, c_2^\star, \pi_{\mathsf{Enc}}^\star, \pi_{\mathsf{Rand}}^\star)$, the proofs

$$\pi_{\mathsf{Enc}}^\star = (C_G^\star, \hat{C}_b^\star, \pi_G^\star, \pi_{bit}^\star, C_0^\star, C_1^\star, C_2^\star, \pi_1^\star, \pi_2^\star, C_z^\star, C_r^\star, \pi_{enc}^\star),$$
$$\pi_{\mathsf{Rand}}^\star = (C_H^\star, \pi_H^\star, C_F^\star, C_{z_{rand}}^\star, C_{r_{rand}}^\star, \pi_{rand}^\star).$$

are obtained by computing $\pi_{\mathsf{Enc}}^\star, \pi_{\mathsf{Rand}}^\star$ as simulated proofs using the trapdoor $\mathsf{tk} = \{(\chi_i, \gamma_i)\}_{i=1}^5$. This is achieved by computing $(\tilde{z}, \tilde{r}) \in \mathbb{G}^2$ as a linearly homomoprhic signature on the vector $\boldsymbol{v}^\star = (1_\mathbb{G}, 1_\mathbb{G}, g, c_1^\star, c_2^\star)$. In step 2 of the encryption algorithm, the challenger thus sets $b = 0$, and conducts the remaining steps of the encryption algorithm except for $(\tilde{z}, \tilde{r})$ at step 4. Thanks to the perfect witness indistinguishability of Groth-Sahai proofs (recall that $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ form CRSes for the perfect NIWI setting in the real game), the NIWI proofs $\pi_{\mathsf{Enc}}^\star, \pi_{\mathsf{Rand}}^\star$ have exactly the same distribution as in Game 1 and $\mathcal{A}$'s view remains unchanged. We have $\Pr[S_2] = \Pr[S_1]$. Note that tk is also used to generate the LHSP signatures on the vectors $\boldsymbol{v}_1, \boldsymbol{v}_2$ of the public key.

Game 3: In this game, we modify the distribution of the public key. In step 2 of the key generation algorithm, we choose $\boldsymbol{u}_2 = \boldsymbol{u}_1^\xi$ and $\hat{\boldsymbol{u}}_2 = \hat{\boldsymbol{u}}_1^\zeta$, with $\xi, \zeta \xleftarrow{R} \mathbb{Z}_p$, instead of choosing $\boldsymbol{u}_2 \xleftarrow{R} \mathbb{G}^2$ and $\hat{\boldsymbol{u}}_2 \xleftarrow{R} \mathbb{G}^2$ uniformly. Under the SXDH assumption, this change should not significantly affect $\mathcal{A}$'s behavior and we have $|\Pr[S_3] - \Pr[S_2]| \leq 2 \times \mathbf{Adv}^{\mathrm{SXDH}}(\lambda)$. Note that $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ now form perfectly sound CRSes.

Game 4: We modify the decryption oracle. When the adversary $\mathcal{A}$ queries the decryption of $\boldsymbol{c} = (c_1, c_2, \pi_{\mathsf{Enc}}, \pi_{\mathsf{Rand}})$, the challenger parses the proofs as

$$\pi_{\mathsf{Enc}} = (C_G, \hat{C}_b, \pi_G, \pi_{bit}, C_0, C_1, C_2, \pi_1, \pi_2, C_z, C_r, \pi_{enc}),$$
$$\pi_{\mathsf{Rand}} = (C_H, \pi_H, C_F, C_{z_{rand}}, C_{r_{rand}}, \pi_{rand})$$

and rejects $\boldsymbol{c}$ if the proofs do not properly verify. Otherwise, instead of merely using the private key $\mathsf{SK} = \alpha$ to compute $M = c_1/c_2^\alpha$ as in the real decryption algorithm, the challenger $\mathcal{B}$ uses the extraction trapdoor $\beta = \log_{u_{1,1}}(u_{1,2})$ of the Groth-Sahai CRS $(\boldsymbol{u}_1, \boldsymbol{u}_2)$, where $\boldsymbol{u}_1 = (u_{1,1}, u_{1,2})$, to extract the witnesses g^b, (z, r) and $\boldsymbol{v} = (\Theta_0, \Theta_1, g/g^b, c_1/\Theta_1, c_2/\Theta_2)$ from their commitments C_G, C_z, C_r and $\{C_i\}_{i=0}^2$ which are contained in π_{Enc}.

Then, the challenger uses $\mathsf{tk} = \{(\chi_i, \gamma_i)\}_{i=1}^5$ to compute an LHSP signatures on $\boldsymbol{v} = (v_1, v_2, v_3, v_4, v_5)$

$$z^\dagger = \prod_{i=1}^5 v_i^{\chi_i}, \qquad\qquad r^\dagger = \prod_{i=1}^5 v_i^{\gamma_i},$$

and rejects the ciphertext in the event that $z^\dagger \neq z$. If $\boldsymbol{c}$ is not rejected, $\mathcal{B}$ computes $M = c_1/c_2^\alpha$. If $M \in \{M_0, M_1\}$ in the post-challenge phase, $\mathcal{B}$ returns "*replay*" as in the actual RCCA game. Otherwise, it returns M to $\mathcal{A}$. It is easy to see that, if $\mathcal{B}$ rejects a ciphertext that would not have been rejected in Game 3, then $\mathcal{B}$ is able to solve the DP problem. This is because $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ are perfectly sound Groth-Sahai CRSes and the validity of the proof π_{enc} implies that (z, r) would be another valid homomorphic signature on $\boldsymbol{v} \in \mathbb{G}^5$ than the one that $\mathcal{B}$ can compute. Therefore, this would provide $\mathcal{B}$ with two distinct linearly homomorphic signatures on the same vector and allow $\mathcal{B}$ to solve an instance of the DP problem as done in the proof of [42, Theorem 1]. We thus have $|\Pr[S_4] - \Pr[S_3]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathrm{DP}}(\lambda)$.

Game 5: We modify the decryption oracle in all pre-challenge and post-challenge decryption queries $\boldsymbol{c} = (c_1, c_2, \pi_{\mathsf{Enc}}, \pi_{\mathsf{Rand}})$ to avoid the use of the secret key $\mathsf{SK} = \alpha = \log_g h$. This change allows modifying the generation of the public element $h = g^x f^y$ with uniformly sampled $x, y \xleftarrow{R} \mathbb{Z}_p$.
In the case of pre-challenge queries, if the commitment C_G contained in π_{Enc} opens to $g^b = 1$ (meaning that $b = 0$), $\mathcal{B}$ rejects the ciphertext. In the case of post-challenge queries $\boldsymbol{c}$, if $g^b = 1$ (i.e., $b = 0$) and the ciphertext is not rejected by the rules of Game 4, the challenger $\mathcal{B}$ returns "replay" without extracting the encrypted message. Additionally, in all decryption queries, if $g^b = g$ (namely, $b = 1$), $\mathcal{B}$ computes $M := c_2 \cdot c_1^{-x} \cdot \Theta_0^{-y}$. Before the challenge phase, it always outputs M. In the case of post-challenge queries, $\mathcal{B}$ returns "replay" if $M \in \{M_0, M_1\}$ and M otherwise. We now analyze the adversary's view in this game under the light of the unforgeability of LHSP signatures:

Before the challenge: It is easy to see that the probability to reject a ciphertext that would not have been rejected in Game 4 is statistically negligible. This follows from the fact that, from the public key, $\mathcal{A}$ has only obtained linearly homomorphic signatures on $(\boldsymbol{v}_1, \boldsymbol{v}_2)$, the span of which clearly does not contain $\boldsymbol{v} = (\Theta_0, \Theta_1, g/g^b, C_1/\Theta_1, C_2/\Theta_2)$ when $g/g^b \neq 1_{\mathbb{G}}$. Therefore, pre-challenge decryption queries for which $g^b = 1$ are rejected in Game 4 except in the event that $z^\dagger = z$. This event only occurs with probability at most $1/p$ at each such query since $z^\dagger$ (as computed from $\boldsymbol{v}$ using tk) is completely unpredictable from the public key. This follows from the fact that honestly-generated LHSP signatures are deterministic functions of tk while there exist exponentially many valid signatures on each vector of messages. The signing key tk retains sufficient entropy to make it statistically impossible to predict the honestly-generated signature on a vector outside the span of $(\boldsymbol{v}_1, \boldsymbol{v}_2)$, which are given in PK.

After the challenge: First, the perfect soundness of the Groth-Sahai proofs $\{\boldsymbol{\pi}_i\}_{i=1}^2$ for relation (12) allows extracting witnesses that satisfy $\Theta_i = c_i^b$, for

each $i \in \{1,2\}$, and then $\boldsymbol{v} = (\Theta_0, c_1^b, g^{1-b}, c_1^{1-b}, c_2^{1-b})$. The difference with pre-challenge queries is that the adversary is also given information on the signature on $\boldsymbol{v}^\star$ from the challenge ciphertext $\boldsymbol{c}^\star$. Hence, in post-challenge queries, LHSP signatures must be in $\mathsf{span}\langle \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}^\star \rangle$. Secondly, we consider the two cases $b \in \{0,1\}$:

- If $g^b = 1_{\mathbb{G}}$ (i.e., $b = 0$), we have $\boldsymbol{v} = (\Theta_0, 1_{\mathbb{G}}, g, c_1, c_2)$. Since $\boldsymbol{c}$ was not rejected, the vector $\boldsymbol{v}$ must be in $\mathsf{span}\langle \boldsymbol{v}_2, (1_{\mathbb{G}}, 1_{\mathbb{G}}, g, c_1^\star, c_2^\star) \rangle$ (i.e. without $\boldsymbol{v}_1$ because the second component of $\boldsymbol{v}$ is $1_{\mathbb{G}}$) except with probability $1/p$. Indeed, otherwise, the same argument as in Game 4 shows that $\boldsymbol{c}$ can only avoid rejection if it contains a commitment C_z to $z^\dagger = z$ and we argued that it is statistically independent of $\mathcal{A}$'s view for vectors outside $\mathsf{span}\langle \boldsymbol{v}_1, \boldsymbol{v}_2, (1_{\mathbb{G}}, 1_{\mathbb{G}}, g, c_1^\star, c_2^\star) \rangle$. This means that $\boldsymbol{v} = \boldsymbol{v}_0 \cdot \boldsymbol{v}_2^\theta$, for some θ, where $\boldsymbol{v}_0 := (1_{\mathbb{G}}, 1_{\mathbb{G}}, g, 1_{\mathbb{G}}, M_\beta)$. Said otherwise, the queried ciphertext is a randomization of the challenge ciphertext, so that $\mathcal{B}$ can rightfully return "*replay*" without changing the view of $\mathcal{A}$.
- If $g^b = g$ (i.e., $b = 1$), we have $\boldsymbol{v} = (\Theta_0, c_1, 1_{\mathbb{G}}, 1_{\mathbb{G}}, 1_{\mathbb{G}})$. Since $\boldsymbol{c}$ was not rejected, $\boldsymbol{v}$ must be in the span of $\boldsymbol{v}_1$ except with probability $1/p$ (via the same argument on the event $z^\dagger = z$ as above). With overwhelming probability $(p-1)/p$, we thus have $\Theta_0 = f^{\log_g c_1}$, which implies that

$$c_1^x \cdot \Theta_0^y = h^{\log_g c_1} = c_1^\alpha$$

  if $\mathsf{SK} := \alpha = x + \log_g(f) \cdot y$ is the secret key that underlies $h = g^x f^y$. It follows that $\mathcal{A}$ obtains the same response as in Game 4.

At each decryption query, $\mathcal{B}$'s response deviates from its response in Game 4 with probability at most $1/p$. A union bound over all decryption queries leads to $|\Pr[S_5] - \Pr[S_4]| \leq q_d/p$ if q_d is the number of decryption queries.

Game 6: We modify the distribution of the challenge ciphertext. Namely, we choose $(c_0^\star, c_1^\star, c_2^\star)$ as a completely random triple $(c_0^\star, c_1^\star, c_2^\star) \xleftarrow{R} \mathbb{G}^3$ instead of a well-formed tuple $(1_{\mathbb{G}}, 1_{\mathbb{G}}, M_\beta) \cdot (f, g, h)^{\theta^\star}$, for a random $\theta^\star \xleftarrow{R} \mathbb{Z}_p$. Under the SXDH assumption, this modification has no noticeable impact on $\mathcal{A}$'s output distribution since, given a DDH_1 instance (g, f, g^a, f^{a+c}) (where either $c = 0$ or $c \in_R \mathbb{Z}_p$), it is sufficient to define $h = g^x \cdot f^y$, as previously, and set $c_0^\star = f^{a+c}$, $c_1^\star = g^a$ and $c_2^\star = M_\beta \cdot (g^a)^x \cdot (f^{a+c})^y$ during the challenge phase. At this point, $(g^a)^x \cdot (f^{a+c})^y = h^a \cdot f^{cy}$ and we obtain the inequality $|\Pr[S_6] - \Pr[S_5]| \leq \mathbf{Adv}^{\mathrm{SXDH}}(\lambda)$.

In Game 6, no information about $\beta \in \{0,1\}$ is leaked anywhere, so that we get $\Pr[S_6] = 1/2$. Since the SXDH assumption implies the DP assumption, we thus find the following advantage

$$|\Pr[S_1] - 1/2| \leq 4 \times \mathbf{Adv}^{\mathrm{SXDH}}(\lambda) + q_d \times 2^{-\lambda},$$

which concludes the proof. □

Acknowledgements

The first author was supported in part by the "Programme Avenir Lyon Saint-Etienne de l'Université de Lyon" in the framework of the programme "Investissements d'Avenir" (ANR-11-IDEX-0007) and in part by the French ANR ALAMBIC project (ANR-16-CE39-0006). The second author is supported by the F.R.S-FNRS as a postdoctoral researcher.

References

1. M. Abe, , J. Groth, K. Haralambiev, and M. Ohkubo. Optimal structure-preserving signatures in asymmetric bilinear groups. *Crypto 2011*. Springer, 2011.
2. M. Abe, J. Camenisch, R. Dowsley, and M. Dubovitskaya. On the impossibility of structure-preserving deterministic primitives. *TCC 2014*. Springer, 2014.
3. M. Abe, J. Camenisch, M. Dubovitskaya, and R. Nishimaki. Universally composable adaptive oblivious transfer (with access control) from standard assumptions. *Digital Identity Management 2013*. ACM Press, 2013.
4. M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. X. Wang and K. Sako, editors, *Advances in Cryptology - ASIACRYPT 2012*. Springer, 2012.
5. M. Abe, B. David, M. Kohlweiss, R. Nishimaki, and M. Ohkubo. Tagged one-time signatures: Tight security and optimal tag size. K. Kurosawa and G. Hanaoka, editors, *Public-Key Cryptography - PKC 2013*. Springer, 2013.
6. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. T. Rabin, editor, *Advances in Cryptology - CRYPTO 2010*. Springer, 2010.
7. M. Abe, J. Groth, and M. Ohkubo. Separating short structure-preserving signatures from non-interactive assumptions. *Asiacrypt 2011*. Springer, 2011.
8. M. Abe, K. Haralambiev, and M. Ohkubo. Signing on elements in bilinear groups for modular protocol design. *IACR Cryptology ePrint Archive*, 2010:133, 2010.
9. M. Abe, K. Haralambiev, and M. Ohkubo. Group to group commitments do not shrink. *Eurocrypt 2012*. Springer, 2012.
10. M. Abe, M. Kohlweiss, M. Ohkubo, and M. Tibouchi. Fully structure-preserving signatures and shrinking commitments. E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*. Springer, 2015.
11. R. Barbulescu, P. Gaudry, A. Joux, and E. Thome. A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. *Eurocrypt 2014*. Springer, 2014.
12. M. Bellare and S. Shoup. Two-tier signatures, strongly unforgeable signatures, and fiat-shamir without random oracles. T. Okamoto and X. Wang, editors, *Public Key Cryptography - PKC 2007*. Springer, 2007.
13. D. Boneh, X. Boyen, and S. Halevi. Chosen ciphertext secure public key threshold encryption without random oracle. *In CT-RSA 2006*. Springer, 2006.
14. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. M. K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004*. Springer, 2004.
15. D. Boneh, G. Segev, and B. Waters. Targeted malleability: homomorphic encryption for restricted computations. *Innovations in Theoretical Computer Science (ITCS) 2012*, 2012.

16. J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. A. Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*. Springer, 2009.
17. J. Camenisch, M. Dubovitskaya, R. Enderlein, and G. Neven. Oblivious transfer with hidden access control from attribute-based encryption. *SCN 2012*, 2012.
18. J. Camenisch, M. Dubovitskaya, and K. Haralambiev. Efficient structure-preserving signature scheme from standard assumptions. *SCN 2012*. Springer, 2012.
19. J. Camenisch, M. Dubovitskaya, K. Haralambiev, and M. Kohlweiss. Composable and modular anonymous credentials: Definitions and practical constructions. *Asiacrypt*, 2015.
20. J. Camenisch, T. Gross, and T. Heydt-Benjamin. Rethinking accountable privacy supporting services. *Digital Identity Management 2008*. ACM Press, 2008.
21. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *Eurocrypt 2004*. Springer, 2004.
22. R. Canetti, H. Krawczyk, and J. B. Nielsen. Relaxing chosen-ciphertext security. D. Boneh, editor, *CRYPTO 2003*. Springer, 2003.
23. J. Cathalo, B. Libert, and M. Yung. Group encryption: Non-interactive realization in the standard model. M. Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009*. Springer, 2009.
24. M. Chase and M. Kohlweiss. A new hash-and-sign approach and structure-preserving signatures from dlin. *SCN 2012*. Springer, 2012.
25. M. Chase, M. Kohlweiss, A. Lysyanskaya, and S. Meiklejohn. Malleable proof systems and applications. D. Pointcheval and T. Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*. Springer, 2012.
26. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. *Crypto 1998*. Springer, 1998.
27. B. M. David, R. Nishimaki, S. Ranellucci, and A. Tapp. Generalizing efficient multiparty computation. A. Lehmann and S. Wolf, editors, *Information Theoretic Security - 8th International Conference, ICITS 2015, Lugano, Switzerland, May 2-5, 2015. Proceedings*. Springer, 2015.
28. Y. Dodis, K. Haralambiev, A. Lopez-Alt, and D. Wichs. Efficient public-key cryptography in the presence of key leakage. *Asiacrypt 2010*. Springer, 2010.
29. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *STOC 1991*. ACM Press, 1991.
30. K. Emura, G. Hanaoka, G. Ohtake, T. Matsuda, and S. Yamada. Chosen ciphertext secure keyed-homomorphic public-key encryption. *PKC 2013*. Springer, 2013.
31. R. Granger, T. Kleinjung, and J. Zumbrägel. Breaking '128-bit secure' supersingular binary curves (or how to solve discrete logarithms in $\mathfrak{O}_{2^{4\cdot 1223}}$ and $\mathfrak{O}_{2^{12\cdot 367}}$). *Crypto 2014*. Springer, 2014.
32. M. Green and S. Hohenberger. Universally composable adaptive oblivious transfer. *Asiacrypt 2008*, 2008.
33. J. Groth. Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. M. Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*. Springer, 2004.
34. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. X. Lai and K. Chen, editors, *Advances in Cryptology - ASIACRYPT 2006*. Springer, 2006.
35. J. Groth. Homomorphic trapdoor commitments to group elements. Cryptology ePrint Archive: Report 2009/007, 2009.

36. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(053), 2007.
37. D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. R. Safavi-Naini and R. Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*. Springer, 2012.
38. C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. K. Sako and P. Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*. Springer, 2013.
39. E. Kiltz. Chosen-ciphertext security from tag-based encryption. *TCC 2006*. Springer, 2006.
40. E. Kiltz, J. Pan, and H. Wee. Structure-preserving signatures from standard assumptions, revisited. R. Gennaro and M. Robshaw, editors, *Advances in Cryptology - CRYPTO 2015*. Springer, 2015.
41. B. Libert and M. Joye. Group signatures with message-dependent opening in the standard model. *CT-RSA 2014*. Springer, 2014.
42. B. Libert, T. Peters, M. Joye, and M. Yung. Linearly homomorphic structure-preserving signatures and their applications. R. Canetti and J. A. Garay, editors, *Advances in Cryptology - CRYPTO 2013*. Springer, 2013.
43. B. Libert, T. Peters, M. Joye, and M. Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive nizk proofs and cca2-secure encryption from homomorphic signatures. *Eurocrypt 2014*. Springer, 2014.
44. B. Libert, T. Peters, and M. Yung. Group signatures with almost-for-free revocation. *Crypto 2012*. Springer, 2015.
45. B. Libert, T. Peters, and M. Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. *Crypto 2015*. Springer, 2015.
46. B. Libert and M. Yung. Non-interactive cca-secure threshold cryptosystems with adaptive security: New framework and constructions. R. Cramer, editor, *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*. Springer, 2012.
47. M. Prabhakaran and M. Rosulek. Rerandomizable RCCA encryption. A. Menezes, editor, *Advances in Cryptology - CRYPTO 2007*. Springer, 2007.
48. M. Prabhakaran and M. Rosulek. Homomorphic encryption with cca security. *ICALP (2) 2008*. Springer, 2008.
49. C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. J. Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91*. Springer, 1991.
50. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*. IEEE Computer Society, 1999.
51. Y. Sakai, N. Attrapadung, and G. Hanaoka. Attribute-based signatures for circuits from bilinear map. *PKC 2016*. Springer, 2016.
52. Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai, T. Matsuda, and K. Omote. Group signatures with message-dependent opening. *Pairing 2012*. Springer, 2012.
53. M. Scott. Authenticated id-based key exchange and remote log-in with simple token and pin number. Cryptology ePrint Archive: Report 2002/164, 2002.
54. V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. *In Euocrypt'98*. Springer, 1998.

A Groth-Sahai Proofs

Our constructions use Groth-Sahai proofs for pairing product equations (PPE) of the form:

$$\prod_{j=1}^{n} e(\mathcal{A}_j, \mathcal{Y}_j) \cdot \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{B}_i) \cdot \prod_{i=1}^{m}\prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{Y}_j)^{\gamma_{i,j}} = t_T,$$

where $\mathcal{X}_i, \mathcal{Y}_j$ are variables in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and $\mathcal{A}_j \in \mathbb{G}, \mathcal{B}_i \in \hat{\mathbb{G}}$ and $t_T \in \mathbb{G}_T$ are constants for $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$.

A non-interactive witness indistinguishable (NIWI) proof system is a tuple of four algorithms ($\mathsf{Setup}, \mathsf{Prove}, \mathsf{VerifyProof}$). Setup outputs a common reference string (CRS) crs, Prove first generates commitments of variables and constructs proofs that these variables satisfy the statement, and $\mathsf{VerifyProof}$ verifies the proof. Such a proof system should satisfy correctness, soundness and witness-indistinguishability. *Correctness* requires that honestly generated proofs for true statements be always accepted by the verifier. *Soundness* guarantees that cheating provers can only prove true statements with all but negligible probability. *Witness-indistinguishability* requires the existence of an efficient simulator $\mathsf{GSSimSetup}$ that produces a common reference string (CRS) crs' which is computationally indistinguishable from a normal crs. When commitments are computed using crs', they are perfectly hiding and the corresponding proofs are witness indistinguishable: i.e., so long as a statement as several witnesses, the proof leaks no information on which specific witness is used to generate it. *Zero-knowledge* additionally requires the existence of an algorithm $\mathsf{GSSimProve}$ that, given a simulated CRS crs' and some trapdoor information τ, generates a simulated proof of the statement without using the witnesses and in such a way that the proof is indistinguishable from a real proof.

In the perfect soundness setting, a CRS $(\boldsymbol{u_1}, \boldsymbol{u_2}, \hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ consists of vectors $\boldsymbol{u_1} = (u_{11}, u_{12})$, $\boldsymbol{u_2} = (u_{21}, u_{22}) \in \mathbb{G}^2$ and $\hat{\boldsymbol{u}}_1 = (\hat{u}_{11}, \hat{u}_{12})$, $\hat{\boldsymbol{u}}_2 = (\hat{u}_{21}, \hat{u}_{22}) \in \hat{\mathbb{G}}^2$ that are linearly dependent. Namely, there exist $\zeta, \hat{\zeta} \in \mathbb{Z}_p$ for which $\boldsymbol{u_2} = \boldsymbol{u_1}^{\zeta}$ and $\hat{\boldsymbol{u}}_2 = \hat{\boldsymbol{u}}_1^{\hat{\zeta}}$. Moreover, NIWI proofs for pairing product equations are perfectly sound (meaning that proofs for false statements do not exist) and the pair $(x, y) = (\log_{u_{11}}(u_{12}), \log_{\hat{u}_{11}}(\hat{u}_{12})) \in \mathbb{Z}_p^2$ can serve as an extraction trapdoor to extract committed group elements $X \in \mathbb{G}$ and $\hat{X} \in \hat{\mathbb{G}}$ from their commitments $\boldsymbol{C}_X = (1, X) \cdot \boldsymbol{u_1}^{\theta_1} \cdot \boldsymbol{u_2}^{\theta_2}$, $\hat{\boldsymbol{C}}_X = (1, \hat{X}) \cdot \hat{\boldsymbol{u}}_1^{\theta_3} \cdot \hat{\boldsymbol{u}}_2^{\theta_4}$. In the perfect witness indistinguishability setting, $(\boldsymbol{u_1}, \boldsymbol{u_2})$ are linearly independent vectors, just like $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$. In this case, $\boldsymbol{C}_X = (1, X) \cdot \boldsymbol{u_1}^{\theta_1} \cdot \boldsymbol{u_2}^{\theta_2}$ and $\hat{\boldsymbol{C}}_X = (1, \hat{X}) \cdot \hat{\boldsymbol{u}}_1^{\theta_3} \cdot \hat{\boldsymbol{u}}_2^{\theta_4}$ are perfectly hiding commitments to X and $\hat{X}$, respectively, and non-interactive proofs for pairing product equations are perfectly witness indistinguishable. Under the SXDH assumption, no PPT adversary can distinguish a perfectly sound CRS from a perfectly hiding CRS.

Regardless of which kind of CRS is used, linear pairing product equations (i.e., where $\gamma_{ij} = 0$ for all i, j) have proofs in $\mathbb{G}^2 \times \hat{\mathbb{G}}^2$ when they involve witnesses in both $\mathbb{G}$ and $\hat{\mathbb{G}}$. When all witnesses are in $\mathbb{G}$, proofs live in $\hat{\mathbb{G}}^2$. For quadratic statement (i.e., where $\gamma_{ij} \neq 0$ for some i, j) the proof are in $\mathbb{G}^4 \times \hat{\mathbb{G}}^4$.

B Definitions for Involved Primitives

B.1 Definitions for Linearly Homomorphic Structure-Preserving Signatures

Let $(\mathbb{G}, \mathbb{G}_T)$ be groups of prime order p such that a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ can be efficiently computed.

A signature scheme is *structure-preserving* [6, 8] if messages, signatures and public keys all live in the group $\mathbb{G}$. In linearly homomorphic structure-preserving signatures, 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. Depending on the application, one may want the tags to be group elements or not. In this paper, they can be arbitrary strings.

Definition 9. *A* linearly homomorphic structure-preserving signature scheme over $(\mathbb{G}, \mathbb{G}_T)$ *is a tuple of efficient algorithms* $\Sigma = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{SignDerive}, \mathsf{Verify})$ *for which the message space consists of* $\mathcal{M} := \mathcal{T} \times \mathbb{G}^n$*, for some integer* $n \in \mathsf{poly}(\lambda)$ *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* $(\mathsf{pk}, \mathsf{sk})$*, where* pk *includes the description of a tag space* $\mathcal{T}$*, where each tag serves as a file identifier.*

$\mathsf{Sign}(\mathsf{sk}, \tau, \boldsymbol{M})$ *takes as input a private key* sk*, a file identifier* $\tau \in \mathcal{T}$ *and a vector of group elements* $\boldsymbol{M} = (M_1, \ldots, M_n) \in \mathbb{G}^n$*. 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 homomorphic signature derivation algorithm. It inputs a public key* pk*, a file identifier* τ *as well as* ℓ *pairs* $(\omega_i, \sigma^{(i)})$*, each of which consists of a coefficient* $\omega_i \in \mathbb{Z}_p$ *and a signature* $\sigma^{(i)} \in \mathbb{G}^{n_s}$*. It outputs a signature* $\sigma \in \mathbb{G}^{n_s}$ *on the vector* $\boldsymbol{M} = \prod_{i=1}^{\ell} \boldsymbol{M}_i{}^{\omega_i}$*, where* $\sigma^{(i)}$ *is a signature on* $\boldsymbol{M}_i$*.*

$\mathsf{Verify}(\mathsf{pk}, \tau, \sigma, \boldsymbol{M})$ *is a verification algorithm that takes as input a public key* pk*, a file identifier* $\tau \in \mathcal{T}$*, a signature* σ *and a vector* $\boldsymbol{M} = (M_1, \ldots, M_n)$*. It outputs* 0 *or* 1*.*

In a *one-time* linearly homomorphic SPS, the tag τ can be omitted in the specification as a given key pair $(\mathsf{pk}, \mathsf{sk})$ only allows signing one linear subspace.

As in all linearly homomorphic signatures, the desired security notion mandates the adversary's inability to come up with a valid triple $(\tau^\star, \boldsymbol{M}^\star, \sigma^\star)$ for a new file identifier $\tau^\star$ or, if $\tau^\star$ appeared in signatures generated by the signing oracle, for a vector $\boldsymbol{M}^\star$ outside the linear span of the vectors that have been legitimately signed for the tag $\tau^\star$.

B.2 Quasi-Adaptive NIZK Arguments

Quasi-Adaptive NIZK (QA-NIZK) proofs [38] are NIZK proofs where the CRS is allowed to depend on the specific language for which proofs have to be generated. The CRS is divided into a fixed part Γ, produced by an algorithm K_0, and a

language-dependent part ψ. However, there should be a single simulator for the entire class of languages.

Let λ be a security parameter. For public parameters Γ produced by K_0, let $\mathcal{D}_\Gamma$ be a probability distribution over a collection of relations $\mathcal{R} = \{R_\rho\}$ parametrized by a string ρ with an associated language

$$\mathcal{L}_\rho = \{x \mid \exists w : R_\rho(x, w) = 1\}.$$

A tuple of algorithms $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ is a QA-NIZK proof system for $\mathcal{R}$ if there exists a PPT simulator $(\mathsf{S}_1, \mathsf{S}_2)$ such that, for any PPT adversaries $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$, we have the properties hereunder.

We assume that the CRS ψ contains an encoding of ρ, which is thus available to V. The definition of Quasi-Adaptive Zero-Knowledge requires a single simulator for the entire family of relations $\mathcal{R}$.

Quasi-Adaptive Completeness:

$$\begin{aligned}&\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow \mathcal{D}_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho);\\ &(x, w) \leftarrow \mathcal{A}_1(\Gamma, \psi);\ \pi \leftarrow \mathsf{P}(\psi, x, w) : \mathsf{V}(\psi, x, \pi) = 1 \text{ if } R_\rho(x, w) = 1] = 1\ .\end{aligned}$$

Quasi-Adaptive Soundness:

$$\begin{aligned}\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow \mathcal{D}_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho);\ (x, \pi) \leftarrow \mathcal{A}_2(\Gamma, \psi) :\\ \mathsf{V}(\psi, x, \pi) = 1\ \wedge\ \neg(\exists w : R_\rho(x, w) = 1)] \in \mathsf{negl}(\lambda)\ .\end{aligned}$$

Quasi-Adaptive Zero-Knowledge:

$$\begin{aligned}&\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow \mathcal{D}_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho)\ :\ \mathcal{A}_3^{\mathsf{P}(\psi, ., .)}(\Gamma, \psi) = 1]\\ &\approx \Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow \mathcal{D}_\Gamma;\ (\psi, \tau_{sim}) \leftarrow \mathsf{S}_1(\Gamma, \rho)\ :\ \mathcal{A}_3^{\mathsf{S}(\psi, \tau_{sim}, ., .)}(\Gamma, \psi) = 1]\,,\end{aligned}$$

where

- $\mathsf{P}(\psi, ., .)$ emulates the actual prover. It takes as input a pair (x, w) and outputs a proof π if $(x, w) \in R_\rho$. Otherwise, it outputs $\perp$.
- $\mathsf{S}(\psi, \tau_{sim}, ., .)$ is an oracle that takes as input (x, w). It outputs a simulated proof $\mathsf{S}_2(\psi, \tau_{sim}, x)$ if $(x, w) \in R_\rho$ and $\perp$ if $(x, w) \notin R_\rho$.

B.3 Definitions of Re-Randomizable Encryption and RCCA Security

A public-key encryption (PKE) scheme is defined as follows.

Definition 10. *(PKE) A public-key encryption (PKE) scheme is tuple of algorithms* $\mathcal{E} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *that:*

$\mathsf{Setup}(\lambda)$: *This is a setup algorithm that takes the security parameter* λ *and generates the public parameter* PP.

$\mathsf{KeyGen}(\mathsf{PP})$: *is key generation algorithm which takes in* PP *and outputs the public key* PK *as well as a secret key* SK.

$\mathsf{Encrypt}(\mathsf{PP}, \mathsf{PK}, m)$: *is a randomized algorithm that takes as input a pair* (PK, m) *made of a public key and a plaintext. It outputs a ciphertext* c.

$\mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}, c)$: *takes in a secret key* SK *and a ciphertext* C. *It outputs either a plaintext* $m \in \mathcal{M}$ *or* $\perp$.

We assume that valid ciphertexts (i.e., which are in the range of the encryption algorithm) are publicly recognizable.

The correctness and public verifiability are defined as follows:

1. *(Correctness) For any ciphertext* c *computed by* $c \leftarrow \mathsf{Encrypt}(\mathsf{SK}, m)$, *we have always* $m = \mathsf{Decrypt}(\mathsf{PK}, c)$.
2. *(Public Verifiability) There exists a PPT algorithm* $\mathsf{Verify}(\mathsf{PP}, \mathsf{PK}, c) \to \{0, 1\}$ *which returns false if and only if* $\mathsf{Decrypt}(\mathsf{PK}, c)$ *outputs* $\perp$.

Definition 11. *A PKE scheme* $\mathcal{E} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is secure against* adaptive chosen-ciphertext attacks (IND-CCA2) *if no PPT adversary* $\mathcal{A}$ *has non-negligible advantage in the experiment below:*

$$\mathbf{Adv}^{IND\text{-}CCA2}_{\mathcal{A},\mathcal{E}}(\lambda) := \Pr\left[\beta = \beta' \;\middle|\; \begin{array}{l} \mathsf{PP} \leftarrow \mathsf{Setup}(\lambda) \\ (\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{KeyGen}(\mathsf{PP}) \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{Decrypt}^1(\mathsf{SK},\cdot)}(\mathsf{PP}, \mathsf{PK}) \\ \beta \xleftarrow{R} \{0,1\} \\ \textit{if } |m_0| \neq |m_1| \\ \quad \textit{then output } (\beta, \beta') \textit{ with } \beta' \xleftarrow{R} \{0,1\} \\ c^\star \leftarrow \mathsf{Encrypt}(\mathsf{PK}, m_\beta) \\ \beta' \leftarrow \mathcal{A}^{\mathsf{Decrypt}^2_{c^\star}(\mathsf{SK},\cdot)}(\mathsf{PP}, \mathsf{PK}, c^\star) \end{array}\right] - \frac{1}{2}$$

In the above experiment, $\mathsf{Decrypt}^1(\mathsf{SK}, \cdot)$ *is an oracle which decrypts any arbitrary ciphertext and* $\mathsf{Decrypt}^2_{c^\star}(\mathsf{SK}, \cdot)$ *is a restricted oracle which allows the adversary to decrypt any ciphertext except* $c^\star$.

We also recall the notion of Replayable Chosen-Ciphertext Security (RCCA) [22], which is defined via a similar security game except that, at each decryption query occurring after the challenge phase, if the ciphertext decrypts to one the messages $\{m_0, m_1\}$, the oracle returns *Replay*.

Definition 12. *A* re-randomizable encryption *scheme is tuple of efficient algorithms* $\mathcal{E} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{ReRand})$ *that:*

$\mathsf{Setup}(\lambda)$: *The setup algorithm takes the security parameter* λ *and generates the public parameter* PP.

$\mathsf{KeyGen}(\mathsf{PP})$: *The key generation algorithm takes* PP, *outputs the public key* PK *and the secret key* SK.

$\mathsf{Encrypt}(\mathsf{PP}, \mathsf{PK}, m)$: *is a randomized encryption algorithm that inputs* $(\mathsf{PP}, \mathsf{PK}, m)$. *It generates a ciphertext* c.

$\mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}, c)$: *The decryption algorithm takes* (PK, C)*, it tries to decrypt the ciphertext if it can not then outputs* $\perp$*, otherwise it outputs the decryption result* m *which is in the message space* $\mathcal{M}$*.*

$\mathsf{ReRand}(\mathsf{PP}, \mathsf{PK}, c)$: *is a probabilistic re-randomization algorithm. It takes as input* (PK, c) *and outputs a new ciphertext* c'*.*

Correctness is generalized as follows. For all $\mathsf{PP} \leftarrow \mathsf{Setup}(\lambda)$*,* $(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{KeyGen}(\mathsf{PP})$*, the following two conditions must hold:*

(a) For any message m*,* $m = \mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}, \mathsf{Encrypt}(\mathsf{PP}, \mathsf{PK}, m))$*.*
(b) For any m*,* $\mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}, \mathsf{ReRand}(\mathsf{PP}, \mathsf{PK}, c)) = \mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}, c)$*.*

Definition 13. *We say that a* re-randomizable encryption *PKE scheme* $\mathcal{E} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{ReRand})$ *is secure against* replayable chosen-ciphertext attacks *(RCCA) if no PPT adversary has noticeable advantage in the experiment below:*

$$\mathbf{Adv}^{RCCA}_{\mathcal{A},\mathcal{E}}(\lambda) := \Pr\left[\beta = \beta' \;\middle|\; \begin{array}{l} \mathsf{PP} \leftarrow \mathsf{Setup}(\lambda) \\ (\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{KeyGen}(\mathsf{PP}) \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{Decrypt}^1(\mathsf{SK},\cdot)}(\mathsf{PP}, \mathsf{PK}) \\ \beta \xleftarrow{R} \{0,1\} \\ \textit{if } |m_0| \neq |m_1| \\ \qquad \textit{then output } (\beta, \beta') \textit{ with } \beta' \xleftarrow{R} \{0,1\} \\ c^\star \leftarrow \mathsf{Encrypt}(\mathsf{PK}, m_b) \\ \beta' \leftarrow \mathcal{A}^{\mathsf{Decrypt}^2_{m_0,m_1}(\mathsf{SK},\cdot)}(\mathsf{PP}, \mathsf{PK}, c^\star) \end{array}\right] - \frac{1}{2}$$

In the experiment, $\mathsf{Decrypt}^1(\mathsf{SK}, \cdot)$ *is an oracle which can decrypt any ciphertext and* $\mathsf{Decrypt}^2_{m_0,m_1}(\mathsf{SK}, \cdot)$ *is a restricted oracle which allows the adversary to decrypt any ciphertext* c*, except when* $\mathsf{Decrypt}(\mathsf{SK}, c) = m_0$ *or* $\mathsf{Decrypt}(\mathsf{SK}, c) = m_1$*. In these two cases, the oracle returns "replay".*

We also define the unlinkability of the re-randomizable encryption scheme which was first proposed by Prabhakaran and Rosulek [47] and re-used by Chase *et al.* [25]. It intuitively captures that, for any ciphertext c in the support of the encryption algorithm, a re-randomization of c is identically distributed to a fresh encryption of $\mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}, c)$.

Definition 14. *We say that a* re-randomizable encryption *PKE scheme* $\mathcal{E} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{ReRand})$ *is* unlinkable *if no PPT adversary* $\mathcal{A}$

has noticeable advantage in the experiment below.

$$
\begin{aligned}
&\mathbf{Adv}^{\mathrm{unlink}}_{\mathcal{A},\mathcal{E}}(\lambda) \\
&:= \Pr\left[\beta = \beta' \;\middle|\; \begin{array}{l}
\mathsf{PP} \leftarrow \mathsf{Setup}(\lambda) \\
(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{KeyGen}(\mathsf{PP}) \\
c \leftarrow \mathcal{A}(\mathsf{PP}, \mathsf{PK}) \\
\beta \xleftarrow{R} \{0,1\} \\
\mathit{if}\ \mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}, c) = \perp \\
\qquad \mathit{then\ output}\ (\beta, \beta')\ \mathit{with}\ \beta' \xleftarrow{R} \{0,1\} \\
\mathit{if}\ \beta = 1 \\
\qquad \mathit{then}\ c^\star \leftarrow \mathsf{Encrypt}(\mathsf{PP}, \mathsf{PK}, \mathsf{Decrypt}(\mathsf{PP}, \mathsf{SK}, c)) \\
\mathit{else}\ c^\star \leftarrow \mathsf{ReRand}(\mathsf{PP}, \mathsf{PK}, c) \\
\beta' \leftarrow \mathcal{A}(\mathsf{PP}, \mathsf{PK}, c^\star) \\
\mathit{output}\ (\beta, \beta')
\end{array} \right] - \frac{1}{2}
\end{aligned}
$$

In the above definition, if the adversary chooses an invalid ciphertext c, we replace its output β' by a random bit so as to annihilate its advantage: namely, re-randomized ciphertexts are only required to be satistically indistinguishable from fresh ciphertexts when the re-randomization is applied to valid ciphertexts.

Like Prabhakaran and Rosulek [47] and Chase *et al.* [25], we will consider statistical unlinkability for valid original ciphertexts.