Algebraic partitioning: Fully compact and (almost) tightly secure cryptography

Dennis Hofheinz*

May 26, 2015

Abstract

We describe a new technique for conducting "partitioning arguments". Partitioning arguments are a popular way to prove the security of a cryptographic scheme. For instance, to prove the security of a signature scheme, a partitioning argument could divide the set of messages into "signable" messages for which a signature can be simulated during the proof, and "unsignable" ones for which any signature would allow to solve a computational problem. During the security proof, we would then hope that an adversary only requests signatures for signable messages, and later forges a signature for an unsignable one.

In this work, we develop a new class of partitioning arguments from simple assumptions. Unlike previous partitioning strategies, ours is based upon an algebraic property of the partitioned elements (e.g., the signed messages), and not on their bit structure. This allows to perform the partitioning efficiently in a "hidden" way, such that already a single "slot" for a partitioning operation in the scheme can be used to implement many different partitionings sequentially, one after the other. As a consequence, we can construct complex partitionings out of simple basic (but algebraic) partitionings in a very space-efficient way.

As a demonstration of our technique, we provide the first signature and public-key encryption schemes that achieve the following properties simultaneously: they are (almost) tightly secure under a simple assumption, and they are fully compact (in the sense that parameters, keys, and signatures, resp. ciphertexts only comprise a constant number of group elements).

Keywords. Partitioning arguments, tight security proofs, digital signatures, public-key encryption.

1 Introduction

Partitioning arguments. Many security reductions rely on a *partitioning argument*. Informally, a partitioning argument divides the parts of a large system into those parts that are under the control of the simulation, and those parts into which a computational challenge can be embedded. For instance, a partitioning argument for a signature scheme could divide the set of message into "signable messages" (for which a signature can be generated by the security reduction), and "unsignable messages" (for which any signature would solve an underlying problem). During the security reduction, we hope that an adversary only asks for the signatures of signable messages, but forges a signature for an unsignable one. Partitioning arguments are a popular means for proving the security of signature schemes (e.g., [33, 16, 35, 28]), identity-based encyption schemes (e.g., [9, 8, 35, 13]), or tightly secure cryptosystems (e.g., [14, 5, 30]).

The complexity of bit-based partitioning. All of the above works (except for [16, 9], which use a programmable random oracle to implement a partitioning) partition messages or identities according to their bit representation. For instance, in the signature scheme from [28], messages

*Karlsruhe Institute of Technology, Dennis.Hofheinz@kit.edu

are signable precisely if they do not start with a particular bit prefix. This non-algebraic approach requires a certain preparation in the scheme itself: already the scheme must establish certain distinctions of messages based on their bit representation. For instance, the signature scheme of [35] uses a hash function of the form $H(M) = h_0 \prod_j h_{j,M_j}$, where M_j are the bits of the signed message M, and h_0 and the $h_{j,b}$ are public group elements. This leads to comparatively large public parameters or keys, in particular because all potential distinctions (based on the values of the M_j) are already present in the scheme.

Our contribution. In this work, we develop an entirely different partitioning approach: instead of partitioning based on the bit representation, we partition according to a simple algebraic predicate. Namely, we view a message M as above as a $\mathbb{Z}_p$-element, and consider various Legendre symbols $L_j = \left(\frac{f_j(M)}{p}\right)$ for different affine functions f_j. Taken together, sufficiently many L_j uniquely determine M, but the computation of each L_j can be encoded as a series of $\mathbb{Z}_p$-operations.[1] Intuitively, this algebraic property allows to "internalize" and hide the computations of the L_j, e.g., by hiding the f_j inside a homomorphic commitment. As a consequence, only one "universal" partitioning (according to a single L_j) needs to be performed in the scheme itself; in the analysis, several simple partitionings can then be implemented sequentially, by varying the f_j.

Comparison with previous partitioning techniques. Compared to previous, bit-based partitioning approaches, our new strategy has the advantage that it simultaneously leads to compact schemes and to a tight security reduction. Previous partitioning strategies were either based on more complex partitionings (such as [33, 8, 35, 28]) that lead to a non-tight security reduction, or on a sequence of simple bit-based partitionings (such as [14, 5, 30]) that lead to large public parameters or keys. In contrast, we support many simple algebraic partitionings (and thus a tight security reduction), but we occupy only one "partitioning slot" in the public parameters. This leads to tightly secure and very compact applications, as we will detail next.

Applications. Specifically, we demonstrate the usefulness of our partitioning technique by describing the first (almost) tightly secure signature and PKE schemes that are fully compact, in the sense that parameters, keys, and signatures (resp. ciphertexts) only contain a constant number of group elements. Our security reduction loses only a factor of $\mathbf{O}(k)$, where k is the security parameter. In particular, our security reduction does not degrade in the number of users or signatures, resp. ciphertexts. The security of our schemes is based upon the Decisional Diffie-Hellman (DDH) assumption in both preimage groups of a pairing. (This assumption is also called "Symmetric External Diffie-Hellman" or SXDH.) Tables 1 and 2 give a more detailed comparison with existing schemes.

In the following, we give more details on our techniques and results. To do so, we start with a little background concerning our applications.

Tight security reductions. To argue for the security of a given cryptographic scheme S, we usually employ a security reduction. That is, we try to argue that every hypothetical adversary $\mathcal{A}_S$ on S can be converted into an adversary $\mathcal{A}_P$ on an allegedly hard computational problem P. In that sense, the only way to break S is to solve P. Of course, we are most interested in reductions to well-investigated problems P. Furthermore, there are reasons to consider the *tightness* of the reduction: a tight reduction guarantees that $\mathcal{A}_P$'s success ε_P in solving P (in a reasonable metric) is about the same as $\mathcal{A}_S$'s success ε_S in attacking S.

To explain the impact of a (non-)tight reduction in more detail, consider a public-key encryption (PKE) scheme S that is deployed in a many-user environment. In this setting, an adversary $\mathcal{A}_S$ on S may observe, say, n_C ciphertexts generated for each of the, say, n_U users. Most known security reductions in this setting are non-tight, in the sense that $\varepsilon_P \leq \frac{\varepsilon_S}{n_U \cdot n_C}$. As a consequence, keylength recommendations should also take n_U and n_C into account; no

[1] Technically, we will not even need to explicitly compute L_j, but only prove that $L_j = 1$. This is possible using a quadratic equation over $\mathbb{Z}_p$.

"universal" keylength recommendation can be given for such a scheme. This is particularly problematic in settings that grow significantly beyond initial expectations.

Tightly secure encryption and signature schemes. The construction of tightly secure cryptographic schemes appears to be a nontrivial task. For instance, although already explicitly considered in 2000 [3], tightly secure PKE schemes have only been constructed very recently [27, 2, 14, 5, 30].[2,3] Moreover, the schemes from [27, 2] have rather large ciphertexts, and the schemes induced by [14, 5] and from [30] require large parameters (but offer small keys and ciphertexts).

The situation for tightly secure signature schemes is somewhat brighter, but results are still limited. There are efficient signature schemes that are tightly secure under "q-type" [7, 15, 34] or interactive [20] assumptions, or in the random oracle model [23, 4, 29]. There are also more recent and somewhat less efficient schemes tightly secure under simple[4] assumptions [11, 27, 14, 5, 30] (see also [1, 2]). Some of these latter schemes can even be converted into tightly secure PKE schemes; however, all of the schemes [11, 27, 2, 14, 5, 30] suffer from asymptotically large parameters, keys, or signatures (resp. ciphertexts).

Scheme	parameters	verification key	signature	reduction loss	assumption
BMS03 [11]	0	$k+3$	$k+1$	$\mathbf{O}(k)$	CDH
HJ12 [27]	2	28	$8k+22$	$\mathbf{O}(1)$	DLIN
CW13 [14]	$2d^2(2n+1)$	d	$4d$	$\mathbf{O}(k)$	d-LIN
BKP14 [5]	d	$d^2(2n+1)$	$2d+1$	$\mathbf{O}(k)$	$\mathcal{D}_d$-MDDH
LJYP14 [30]	0	$\mathbf{O}(d^2 n)$	$2d+1$	$\mathbf{O}(k)$	d-LIN
This work	14	6	25	$\mathbf{O}(k)$	DDH

Table 1: Comparison of different (at least almost) tightly EUF-CMA secure signature schemes from simple[4] assumptions in pairing-friendly groups. The **parameters**, **verification key**, and **signature** columns denote space complexity, measured in group elements. The **reduction loss** column denotes the (multiplicative) loss of the security reduction to the respective **assumption**. For the schemes from [14, 5], we assume the signature scheme induced by the presented IBE scheme. Furthermore, $n = \Theta(k)$ denotes the bitlength of the signed message (if the signed message is a bitstring and not a group element or an exponent). We note that [30] mention that their scheme can be generalized to the d-LIN assumption (including 1-LIN=DDH). However, since they only give explicit complexities for the arising signatures (identical to the ones from [5]), we restrict to their DLIN-based scheme. Finally, we remark that all of these schemes (except for [11]) imply tightly secure PKE schemes (cf. Table 2).

The scheme of Chen and Wee. Our technical ideas are best presented with our signature scheme. At a very high level, we follow the strategy of Chen and Wee [14] (see also [5]), where we interpret their IBE scheme as a signature scheme using Naor's trick [10]. In their scheme, signatures are of the form

$$\sigma = \Big(h_0, \ \mathit{sigk} \cdot \prod_{i=1}^{n} h_{i,M_i} \Big), \tag{1}$$

where sigk is the secret key, $M = (M_i)_{i=1}^n \in \{0,1\}^n$ is the bit representation of the signed message, and $h_0, (h_{i,0}, h_{i,1})_{i=1}^n$ are group elements chosen from a joint public distribution.[5]

During their proof of existential unforgeability (EUF-CMA security), Chen and Wee gradually modify signatures generated by the security experiment for an adversary $\mathcal{A}$. This is done via a small hybrid argument over the bit indices of messages, and thus yields a security proof

[2] Actually, [14, 5] construct tightly secure identity-based encryption (IBE) schemes. However, those IBE schemes can be viewed as tightly secure signature schemes (using Naor's trick [10]), and then converted into tightly secure PKE schemes using the transformation from [27]. In fact, the PKE scheme of [30] can be viewed as a (modified and highly optimized) conversion of the IBE scheme from [14].

[3] We note that earlier PKE schemes achieve at least a certain form of tight security under "q-type" assumptions [21, 22, 26], or in the random oracle model [19, 12, 6].

[4] With a "simple" assumption, we mean one in which the adversary gets a challenge whose size only depends on the security parameter, and is then supposed to output a unique solution without further interaction. Examples of simple assumptions are DLOG, DDH, d-LIN, or RSA, but not, say, Strong Diffie-Hellman [7] or q-ABDHE [21].

[5] We note that although their scheme can be viewed as a generalization of Waters signatures [35], their analysis is entirely different. Also, we omit here certain subtleties regarding the used distributions of group elements.

Scheme	parameters	public key	ciphertext	reduction loss	assumption
HJ12 [27]	$\mathbf{O}(1)$	$\mathbf{O}(1)$	$\mathbf{O}(k)$	$\mathbf{O}(1)$	DLIN
AKDNO13 [2]	$\mathbf{O}(1)$	$\mathbf{O}(1)$	$\mathbf{O}(k)$	$\mathbf{O}(1)$	DLIN
CW13 [14]	$\mathbf{O}(d^2k)$	$\mathbf{O}(d)$	$\mathbf{O}(d)$	$\mathbf{O}(k)$	d-LIN
BKP14 [5]	$\mathbf{O}(d)$	$\mathbf{O}(d^2k)$	$\mathbf{O}(d)$	$\mathbf{O}(k)$	$\mathcal{D}_d$-MDDH
LJYP14 [30]	$\mathbf{O}(1)$	$\mathbf{O}(d^2k)$	$\mathbf{O}(d)$	$\mathbf{O}(k)$	d-LIN
LJYP14 [30]	3	$24k+30$	69	$\mathbf{O}(k)$	DLIN
This work	15	2	60	$\mathbf{O}(k)$	DDH

Table 2: Comparison of different (at least almost) tightly IND-CCA secure PKE schemes from simple[4] assumptions. As in Table 1, the **parameters**, **public key**, and **ciphertext** columns denote space complexity, measured in group elements, and the **reduction loss** column denotes the (multiplicative) loss of the security reduction to the respective **assumption**. For the schemes from [14, 5], we assume the PKE scheme induced by the respective signature scheme when going through the construction of [27]. We note that [30] only describe a symmetric-pairing version of their scheme, so their DDH-based scheme is not explicit. However, we expect that their DDH-based scheme has slightly more compact ciphertexts than ours.

that loses a factor of $\mathbf{O}(n)$. Concretely, in the i-th hybrid, generated signatures are of the form $\sigma = (h_0, sigk_{M_1,\ldots,M_i} \cdot \prod_{j=1}^{n} h_{j,M_j})$, where $sigk_{M_1,\ldots,M_i} = \mathcal{R}(M_1, \ldots, M_i)$ for a truly random function $\mathcal{R}$. Similarly, a forged message-signature pair (M^*, σ^*) from $\mathcal{A}$ is only considered valid if it is consistent with $sigk_{M_1^*,\ldots,M_i^*}$ (instead of $sigk$). In other words, in the i-th hybrid, the secret key used in signatures depends on the first i bits of the signed message.

Thus, the difference between the $(i-1)$-th and the i-th hybrid is an additional dependency of used secret keys on the i-th message bit M_i. To progress from hybrid $i-1$ to hybrid i, Chen and Wee first partition the message space in two halves (according to M_i). Then, using an elaborate argument, they consistently modify the secret keys used for messages from one half, and thus essentially decouple those keys from the keys used for messages from the other half. This creates an additional dependency on M_i. After $n = |M|$ such steps, each signature uses a different secret key (up to multiple signatures of the same message). In particular, $\mathcal{A}$ gets no information about the secret key $sigk_{M_1^*,\ldots,M_n^*}$ used to verify its own forgery, and existential unforgeability follows.

We would like to highlight the partitioning character of their analysis: in their proof, Chen and Wee introduce more and more dependencies of signatures on the corresponding messages, and each such dependency is based upon a different partitioning of the message space.[6] Now observe that already regular signatures (as in (1)) feature distinctions based on all bits of M. These distinctions provide the technical tool to introduce dependencies in the security proof. However, as a consequence, rather complex joint distributions need to be sampled during signature generation, which results in public parameters of $\mathbf{O}(n)$ group elements.

Algebraic partitioning. In a nutshell, our main technical tool is a new way to partition the message space of a signature scheme. We call this tool "algebraic partitioning." Concretely, a signature for a message $M \in \mathbb{Z}_p$ in our scheme consists essentially of an encryption of the secret key X, along with a consistency proof:

$$\sigma \;=\; (\; C = \mathrm{Enc}(pk, X),\; \pi\;). \tag{2}$$

The corresponding encryption key pk is part of the verification key vk, and the consistency proof π proves the following statement:

"**Either** C encrypts the secret key X, **or** $f(M) \in \mathbb{Z}_p$ is a quadratic residue (or both)."

Here, p is the order of the underlying group, and $f : \mathbb{Z}_p \to \mathbb{Z}_p$ is an affine function fixed (but hidden) in the verification key. Implicitly, this provides a *single* partitioning of messages into

[6] We note that a similar technique has also been used in the context of pseudorandom functions [24, 31].

those for which $f(M)$ is a quadratic residue, and those for which $f(M)$ is not. However, since f is hidden, many partitionings can be induced (one after the other) by varying f during a proof.

In fact, during the security proof, this partitioning will fulfill the same role as the bit-based partitioning in the analysis of Chen and Wee. In particular, it will help to introduce additional dependencies of the signature on the message. More specifically, in the i-th hybrid of the security proof, C will not encrypt X, but a value X_M that depends on the i Legendre symbols $\left(\frac{f_j(M)}{p}\right)$ for randomly chosen (but fixed) affine functions $f_1, \ldots, f_i$. Each new such dependency is introduced by first refreshing the affine function f hidden in vk, and then modifying all values encrypted in signatures whenever possible (i.e., whenever $f(M)$ is a quadratic residue).[7] Observe that the single explicit partitioning in regular signatures is used several times (for different f_j) to introduce many dependencies of signatures on messages in the proof. The remaining strategy can then be implemented as in [14].

Our different strategy to partition the message space results in a very compact scheme. Namely, since only one explicit partitioning step is performed in the scheme, parameters, keys, and signatures comprise only a constant number of group elements. Specifically, parameters, keys, and signatures contain 14, 6, and 25 group elements, respectively. Besides, our scheme is compatible with Groth-Sahai proofs [25]. Hence, when used in the construction of [27], we immediately get the first compact (in the above sense) PKE scheme that is tightly IND-CCA secure under a simple assumption.[8]

Roadmap. After recalling some basic definitions, we present our signature scheme in Section 3. In Appendix A, we give a direct construction of a PKE scheme derived from our signature scheme. Finally, in Appendix B, we give more details on the exact Groth-Sahai equations arising from the consistency proofs of signatures and ciphertexts.

Acknowledgements. The author would like to thank Eike Kiltz, Julia Hesse, Willi Geiselmann, and the anonymous reviewers for helpful feedback.

2 Preliminaries

Notation. Throughout the paper, $k \in \mathbb{N}$ denotes the security parameter. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. For a finite set S, we denote with $s \leftarrow S$ the process of sampling s uniformly from S. For a probabilistic algorithm A, we denote with $y \leftarrow A(x; R)$ the process of running A on input x and with randomness R, and assigning y the result. We write $y \leftarrow A(x)$ for $y \leftarrow A(x; R)$ with uniformly chosen R, and we write $A(x) = y$ for the event that $A(x; R)$ (for uniform R) outputs y. If A's running time is polynomial in k, then A is called probabilistic polynomial-time (PPT). A function $f : \mathbb{N} \to \mathbb{R}$ is negligible if it vanishes faster than the inverse of any polynomial (i.e., if $\forall c \exists k_0 \forall k \geq k_0 : |f(x)| \leq 1/k^c$).

Collision-resistant hashing. A hash function generator is a PPT algorithm $\mathcal{H}$ that, on input 1^k, outputs (the description of) an efficiently computable function $H : \{0,1\}^* \to \{0,1\}^k$.

Definition 2.1 (Collision-resistance)**.** *We say that a hash function generator $\mathcal{H}$ outputs collision-resistant functions* H *(or, when the reference to $\mathcal{H}$ is clear, that such an* H *is collision-resistant), if*

$$\mathrm{Adv}^{\mathrm{cr}}_{\mathcal{H},\mathcal{A}}(k) \;=\; \Pr\left[x \neq x' \;\wedge\; H(x) = H(x') \;\middle|\; H \leftarrow \mathcal{H}(1^k),\ (x, x') \leftarrow \mathcal{A}(1^k, H)\right]$$

is negligible for every PPT adversary $\mathcal{A}$.

[7] This neglects a number of details. For instance, in the somewhat simplified scheme above, π always ties the ciphertexts in signatures for quadratic non-residues $f(M)$ to a single value X. In our actual proof, we will thus simulate a part of π, such that the encrypted values can be decoupled from the original secret key X.

[8] Actually, plugging our scheme directly into the construction of [27] yields an asymptotically compact, but not very efficient scheme. Thus, we provide a more direct and efficient explicit PKE construction with parameters, public keys, and ciphertexts comprised of 15, 2, and 60 group elements, respectively.

Signature schemes. A signature scheme SIG consists of four PPT algorithms SPars, SGen, Sig, Ver. Parameter generation $\mathrm{SPars}(1^k)$ outputs public parameters *spp* that are shared among all users. Key generation $\mathrm{SGen}(spp)$ takes public parameters *spp*, and outputs a verification key *vk* and a signing key *sigk*. The signature algorithm $\mathrm{Sig}(spp, sigk, M)$ takes public parameters *spp*, a signing key *sigk*, and a message M, and outputs a signature σ. Verification $\mathrm{Ver}(spp, vk, M, \sigma)$ takes public parameters *spp*, a verification key *vk*, a message M, and a potential signature σ, and outputs a verdict $b \in \{0,1\}$. For correctness, we require that $1 \leftarrow \mathrm{Ver}(spp, vk, M, \sigma) = 1$ always and for all M, all $(vk, sigk) \leftarrow \mathrm{SGen}(1^k)$, and all $\sigma \leftarrow \mathrm{Sig}(spp, sigk, M)$. For the sake of readability, we will omit the public parameters *spp* from invocations of Sig and Ver when the reference is clear.

Definition 2.2 (Multi-user (one-time) existential unforgeability)**.** *Let* SIG *be a signature scheme as above, and consider the following experiment for an adversary $\mathcal{A}$:*

1. *$\mathcal{A}$ specifies (in unary) the number $n_U \in \mathbb{N}$ of desired scheme instances.*
2. *The experiment then samples parameters $spp \leftarrow \mathrm{SPars}(1^k)$ as well as n_U keypairs $(vk^{(\ell)}, sigk^{(\ell)}) \leftarrow \mathrm{SGen}(spp)$.*
3. *$\mathcal{A}$ is invoked on input $(1^k, spp, (vk^{(\ell)})_{\ell=1}^{n_U})$, and gets access to signing oracles $\mathrm{Sig}(sigk^{(\ell)}, \cdot)$ for all $\ell \in [n_U]$. Finally, $\mathcal{A}$ outputs an index $\ell^* \in [n_U]$ and a potential forgery (M^*, σ^*).*
4. *$\mathcal{A}$ wins iff $\mathrm{Ver}(vk^{(\ell^*)}, M^*, \sigma^*) = 1$ and M^* has not been queried to $\mathrm{Sig}(sigk^{(\ell^*)}, \cdot)$.*

Let $\mathrm{Adv}^{\text{euf-mcma}}_{\mathrm{SIG},\mathcal{A}}(k)$ denote the probability that $\mathcal{A}$ wins in the above experiment. We say that SIG *is existentially unforgeable under chosen-message attacks in the multi-user setting (EUF-mCMA secure) iff $\mathrm{Adv}^{\text{euf-mcma}}_{\mathrm{SIG},\mathcal{A}}(k)$ is negligible for every PPT $\mathcal{A}$. Let $\mathrm{Adv}^{\text{ot-euf-mcma}}_{\mathrm{SIG},\mathcal{A}}(k)$ be the probability that $\mathcal{A}$ wins in the slightly modified experiment in which only one* Sig*-query to each scheme instance ℓ is allowed. We say that* SIG *is existentially unforgeable under one-time chosen-message attacks in the multi-user setting (OT-EUF-mCMA secure) iff $\mathrm{Adv}^{\text{ot-euf-mcma}}_{\mathrm{SIG},\mathcal{A}}(k)$ is negligible for every PPT $\mathcal{A}$.*

Public-key encryption schemes. A public-key encryption (PKE) scheme PKE consists of four PPT algorithms (EPars, EGen, Enc, Dec). The parameter generation algorithm $\mathrm{EPars}(1^k)$ outputs public parameters *epp*. Key generation $\mathrm{EGen}(epp)$ outputs a public key *pk* and a secret key *sk*. Encryption $\mathrm{Enc}(epp, pk, M)$ takes parameters *epp*, a public key *pk*, and a message M, and outputs a ciphertext C. Decryption $\mathrm{Dec}(epp, sk, C)$ takes public parameters *epp*, a secret key *sk*, and a ciphertext C, and outputs a message M. For correctness, we require $\mathrm{Dec}(epp, sk, C) = M$ always and for all M, all $epp \leftarrow \mathrm{EPars}(1^k)$, all $(pk, sk) \leftarrow \mathrm{EGen}(epp)$, and all $C \leftarrow \mathrm{Enc}(epp, pk, M)$. As with signatures, we usually omit the public parameters *epp* from invocations of Enc and Dec.

Definition 2.3 (Multi-user, multi-challenge indistinguishability of ciphertexts)**.** *For a public-key encryption scheme* PKE *and an adversary $\mathcal{A}$, consider the following security experiment $\mathrm{Exp}^{\text{ind-mcca}}_{\mathrm{PKE},\mathcal{A}}(k)$:*

1. *$\mathcal{A}$ specifies (in unary) the number $n_U \in \mathbb{N}$ of desired scheme instances.*
2. *The experiment samples parameters $epp \leftarrow \mathrm{EPars}(1^k)$, and n_U keypairs through $(pk^{(\ell)}, sk^{(\ell)}) \leftarrow \mathrm{EGen}(epp)$, and uniformly chooses a bit $b \leftarrow \{0,1\}$.*
3. *$\mathcal{A}$ is invoked on input $(1^k, epp, (pk^{(\ell)})_{\ell=1}^{n_U})$, and gets access to challenge oracles $\mathcal{O}^{(\ell)}$ and decryption oracles $\mathrm{Dec}(sk^{(\ell)}, \cdot)$ for all $\ell \in [n_U]$. Here, challenge oracle $\mathcal{O}^{(\ell)}$, on input two messages M_0, M_1, outputs an encryption $C \leftarrow \mathrm{Enc}(pk^{(\ell)}, M_b)$ of M_b.*
4. *Finally, $\mathcal{A}$ outputs a bit b', and the experiment outputs 1 iff $b = b'$.*

A PPT adversary $\mathcal{A}$ is valid if every pair (M_0, M_1) of messages submitted to an $\mathcal{O}^{(\ell)}$ by $\mathcal{A}$ satisfies $|M_0| = |M_1|$, and if $\mathcal{A}$ never submits any challenge ciphertext (previously received from an $\mathcal{O}^{(\ell)}$) to the corresponding decryption oracle $\mathrm{Dec}(sk^{(\ell)}, \cdot)$. Let

$$\mathrm{Adv}^{\text{ind-mcca}}_{\mathrm{PKE},\mathcal{A}}(k) \;=\; \Pr\left[\mathrm{Exp}^{\text{ind-mcca}}_{\mathrm{PKE},\mathcal{A}}(k) = 1\right] - 1/2.$$

We say that PKE *has indistinguishable ciphertexts under chosen-ciphertext attacks in the multi-user, multi-challenge setting (short: is IND-mCCA secure) iff $\mathrm{Adv}^{\text{ind-mcca}}_{\mathrm{PKE},\mathcal{A}}(k)$ is negligible for all valid $\mathcal{A}$.*

Let $\mathrm{Adv}^{\text{ind-mcpa}}_{\mathrm{PKE},\mathcal{A}}$ *be defined similarly, except that* $\mathcal{A}$ *has no access to any* Dec *oracles.* PKE *has indistinguishable ciphertexts under chosen-plaintext attacks in the multi-user, multi-challenge setting (short: is IND-mCPA secure) iff* $\mathrm{Adv}^{\text{ind-mcpa}}_{\mathrm{PKE},\mathcal{A}}(k)$ *is negligible for all valid* $\mathcal{A}$*.*

Quadratic residues and Legendre symbols. Let p be a prime. Then, $\mathrm{QR}_p \subseteq \mathbb{Z}_p^*$ is the set of quadratic residues modulo p, i.e., the set of all $x \in \mathbb{Z}_p^*$ for which an $r \in \mathbb{Z}_p^*$ with $r^2 = x \bmod p$ exists. Given p and an $x \in \mathrm{QR}_p$, such an r can be computed efficiently. For $x \in \mathbb{Z}_p$, we let $\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}} \bmod p$ denote the Legendre of x modulo p. We have $\left(\frac{x}{p}\right) \in \{-1, 0, 1\}$, and in particular $\left(\frac{x}{p}\right) = 1 \Leftrightarrow x \in \mathrm{QR}_p$, as well as $\left(\frac{x}{p}\right) = 0 \Leftrightarrow x = 0$, and $\left(\frac{x}{p}\right) = -1 \Leftrightarrow x \in \mathbb{Z}_p^* \setminus \mathrm{QR}_p$.

Group and pairing generators. A group generator $\mathcal{G}$ is a PPT algorithm that, on input 1^k, outputs the description of a group $\mathbb{G}$, along with its (prime) order p, and a generator g of $\mathbb{G}$. A pairing generator $\mathcal{P}$ is a PPT algorithm that, on input 1^k, outputs descriptions of:

- three groups $\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T$ of the same prime order p, along with p, and generators $g, \hat{g}$ of $\mathbb{G}, \hat{\mathbb{G}}$,
- a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ that is non-degenerate in the sense of $e(g, \hat{g}) \neq 1 \in \mathbb{G}_T$.

Occasionally, it will also be useful to consider a pairing generator $\mathcal{P}$ as a group generator (that only outputs $(\mathbb{G}, p, g)$ or $(\hat{\mathbb{G}}, p, \hat{g})$).

Assumption 2.4 (Decisional Diffie-Hellman)**.** *For a group generator* $\mathcal{G}$ *and an adversary* $\mathcal{A}$*, let* $\mathrm{Adv}^{\mathrm{ddh}}_{\mathcal{G},\mathcal{A}}(k)$ *be the following difference:*

$$\Pr\left[\mathcal{A}(1^k, \mathbb{G}, p, g, g^x, g^y, g^{xy}) = 1\right] - \Pr\left[\mathcal{A}(1^k, \mathbb{G}, p, g, g^x, g^y, g^z) = 1\right].$$

Here, the probability is over $(\mathbb{G}, p, g) \leftarrow \mathcal{G}(1^k)$ *and uniformly chosen* $x, y, z \in \mathbb{Z}_p$*. We say that the Decisional Diffie-Hellman (DDH) assumption holds with respect to* $\mathcal{G}$ *iff* $\mathrm{Adv}^{\mathrm{ddh}}_{\mathcal{G},\mathcal{A}}$ *is negligible for every PPT* $\mathcal{A}$*. When the reference to* $\mathcal{G}$ *is clear, we also say that the DDH assumption holds in* $\mathbb{G}$ *(and write* $\mathrm{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{A}}$*). On occasion, we might also say that the DDH assumption holds in groups* $\mathbb{G}$ *or* $\hat{\mathbb{G}}$ *sampled by a pairing generator, with the obvious meaning.*

ElGamal encryption. The ElGamal encryption scheme $\mathrm{PKE}_{\mathrm{eg}}$ is defined as follows, where we assume a suitable group generator $\mathcal{G}$.

- $\mathrm{EPars}_{\mathrm{eg}}(1^k)$ runs $(\mathbb{G}, p, g) \leftarrow \mathcal{G}(1^k)$ and outputs $epp = (\mathbb{G}, p, g)$.
- $\mathrm{EGen}_{\mathrm{eg}}(epp)$ picks a uniform $sk \leftarrow \mathbb{Z}_p$, sets $pk = g^{sk}$, and outputs (pk, sk).
- $\mathrm{Enc}(pk, M)$, for $M \in \mathbb{G}$, picks an $R \leftarrow \mathbb{Z}_p$, and outputs $C = (g^R, pk^R \cdot M)$.
- $\mathrm{Dec}(sk, C)$, for $C = (C_1, C_2) \in \mathbb{G}^2$, outputs $M = C_2 / C_1^{sk}$.

The ElGamal scheme is tightly IND-mCPA secure under the DDH assumption in $\mathbb{G}$. Concretely, for every valid IND-mCPA adversary $\mathcal{A}$, there is a DDH adversary $\mathcal{B}$ (of roughly the same complexity as the IND-mCPA experiment with $\mathcal{A}$) with $\mathrm{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{B}}(k) = \mathrm{Adv}^{\text{ind-mcpa}}_{\mathrm{PKE}_{\mathrm{eg}},\mathcal{A}}(k)$.

Groth-Sahai proofs. In a setting with a pairing generator, Groth-Sahai proofs [25] provide a very versatile and efficient way to prove the satisfiability of very general classes of equations over $\mathbb{G}$ and $\hat{\mathbb{G}}$. We will not need them in full generality, and the next definition only captures a number of abstract properties of Groth-Sahai proofs we will use. In particular, we will not formalize the exact classes of languages amenable to Groth-Sahai proofs. (For the exact languages used in our application, however, we give more details in Appendix B.1.) Like [18, 17], we formalize Groth-Sahai proofs as *commit-and-prove* systems:

Definition 2.5 (GS proofs [25])**.** *The Groth-Sahai proof system for a given pairing generator* $\mathcal{P}$ *consists of the following PPT algorithms, where gpp denotes group parameters sampled by* $\mathcal{P}$*.*

Common reference strings. $\mathrm{HGen}(gpp)$ *and* $\mathrm{BGen}(gpp)$ *sample hiding, resp. binding common reference strings (CRSs)* CRS*.*

Commitments. *For a (hiding or binding) CRS* CRS *and a* $\mathbb{G}$*-,* $\hat{\mathbb{G}}$*-, or* $\mathbb{Z}_p$*-element* v*, the commitment algorithm* $\mathrm{Com}(gpp, \mathrm{CRS}, v; R)$ *outputs a commitment* C*, where* R *denotes the used random coins.*

Proofs. *Let* CRS *be a CRS, and let* $\mathcal{X}$ *be a system of equations. Each equation may be over* $\mathbb{G}$, $\hat{\mathbb{G}}$, *or* $\mathbb{Z}_p$, *and involve variables and constants. Let* $(\nu_i)_i$ *be a variable assignment that satisfies* $\mathcal{X}$, *and let* $(R_i)_i$ *be a vector of random coins for* Com. *Then* $\text{Prove}(gpp, \text{CRS}, \mathcal{X}, (\nu_i, R_i)_i)$ *outputs a proof* π.

Verification. *For a CRS* CRS, *a system* $\mathcal{X}$ *of equations, a commitment vector* $(C_i)_i$ *to an assignment of the variables in* $\mathcal{X}$, *and a proof* π, $\text{Verify}(gpp, \text{CRS}, \mathcal{X}, (C_i)_i, \pi)$ *outputs a verdict* $b \in \{0,1\}$.

Simulation. *For a hiding CRS generated as* $\text{CRS} \leftarrow \text{HGen}(gpp; R_{\text{CRS}})$, *a system* $\mathcal{X}$ *of equations, and a vector* $(R_i)_i$ *of commitment random coins, we have that* $\text{Sim}(gpp, R_{\text{CRS}}, \mathcal{X}, (R_i)_i)$ *outputs a simulated proof* π.

As with signatures and encryption, we usually omit the group parameters *gpp* on invocations of Com, Prove, Verify, Sim when the reference is clear.

Theorem 2.6 (Properties of GS proofs [25])**.** *The algorithms from Definition 2.5 satisfy the following for all choices group parameters* $gpp \leftarrow \mathcal{P}(1^k)$ *(unless noted otherwise):*

Homomorphic commitments. *For any (hiding or binding) CRS* CRS, *any two given commitments* $\text{Com}(\text{CRS}, \nu; R)$ *and* $\text{Com}(\text{CRS}, \nu'; R')$ *to* $\mathbb{G}$*-elements* ν, ν' *allow to efficiently compute a commitment* $\text{Com}(\text{CRS}, \nu \cdot \nu'; R \cdot R')$ *to* $\nu \cdot \nu'$. *(Note that the corresponding random coins* $R \cdot R'$ *can be efficiently computed from* R *and* R'*.) The same holds for two commitments to* $\hat{\mathbb{G}}$*-elements, and two commitments to* $\mathbb{Z}_p$*-elements (where the homomorphic operation on* $\mathbb{Z}_p$*-elements is addition).*

Dual-mode commitments. *Consider a commitment* $C \leftarrow \text{Com}(\text{CRS}, \nu; R)$. *If* CRS *is binding, then* C *uniquely determines* ν, *and if* CRS *is hiding, then the distribution of* C *does not depend on* ν.

CRS indistinguishability. *For every PPT adversary* $\mathcal{A}$, *there are PPT adversaries* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *with*

$$\left|\Pr\left[\mathcal{A}(1^k, \text{HGen}(gpp)) = 1\right] - \Pr\left[\mathcal{A}(1^k, \text{BGen}(gpp)) = 1\right]\right| \leq \left|\text{Adv}^{\text{ddh}}_{\mathbb{G},\mathcal{A}_1}(k)\right| + \left|\text{Adv}^{\text{ddh}}_{\hat{\mathbb{G}},\mathcal{A}_2}(k)\right|,$$

where the probability is over $gpp \leftarrow \mathcal{P}(1^k)$, *and the random coins of* HGen, BGen, *and* $\mathcal{A}$.

Perfect completeness. *For every (hiding or binding) CRS* CRS, *every system* $\mathcal{X}$ *of equations, every satisfying assignment* $(\nu_i)_i$ *of* $\mathcal{X}$, *and every possible vector* $(C_i)_i$ *of commitments generated through* $C_i \leftarrow \text{Com}(\text{CRS}, \nu_i; R_i)$, *we always have* $\text{Verify}(\text{CRS}, \mathcal{X}, (C_i)_i, \text{Prove}(\text{CRS}, \mathcal{X}, (\nu_i, R_i)_i)) = 1$.

Perfect soundness. *For every binding CRS* CRS, *every system* $\mathcal{X}$ *of equations that is not satisfiable, and every* $(C_i)_i$ *and* π, $\text{Verify}(\text{CRS}, \mathcal{X}, (C_i)_i, \pi) = 0$ *always.*

Perfect simulation. *For every hiding CRS* $\text{CRS} \leftarrow \text{HGen}(gpp; R_{\text{CRS}})$, *and every system* $\mathcal{X}$ *of equations that is satisfied by a variable assignment* $(\nu_i)_i$, *the following two distributions are identical:*

$$\big((C_i)_i,\ \text{Prove}(\text{CRS}, \mathcal{X}, (\nu_i, R_i)_i)\big) \quad \text{for } C_i \leftarrow \text{Com}(\text{CRS}, \nu_i; R_i) \text{ and fresh } R_i,$$
$$\big((C_i)_i,\ \text{Sim}(R_{\text{CRS}}, \mathcal{X}, (R_i)_i)\big) \quad \text{for } C_i \leftarrow \text{Com}(\text{CRS}, 1; R_i) \text{ and fresh } R_i.$$

(The probability space consists of the R_i *and the coins of* Prove *and* Sim.*)*

Since simulation is perfect (in the sense above), it also holds for reused commitments (i.e., when multiple adaptively chosen statements $\mathcal{X}$ that involve the same variables and commitments are proven, see also [17]). Besides, perfect simulation directly implies perfect witness-indistinguishability (under a hiding CRS): for any two vectors $(\nu_i)_i$ and $(\nu'_i)_i$ of satisfying assignments of a given system $\mathcal{X}$ of equations, the corresponding commitments and proofs $((C_i)_i, \pi)$ and $((C'_i)_i, \pi')$ are identically distributed. Again, this holds even if the same commitments are used in several proofs for adaptively generated statements $\mathcal{X}$.

3 The signature scheme

3.1 Scheme description

Setting and ingredients. We assume the following ingredients:

- A pairing generator $\mathcal{P}$ that outputs groups $\mathbb{G} = \langle g \rangle$ and $\hat{\mathbb{G}} = \langle \hat{g} \rangle$ of prime order $p > 2^k$ and an asymmetric pairing $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$. We make the DDH assumption in both $\mathbb{G}$ and $\hat{\mathbb{G}}$.
- The ElGamal encryption scheme (given by algorithms $\mathrm{EGen}_{\mathrm{eg}}, \mathrm{Enc}_{\mathrm{eg}}, \mathrm{Dec}_{\mathrm{eg}}$) over $\mathbb{G}$. (That is, we will use $\mathcal{P}$ in place of $\mathrm{EPars}_{\mathrm{eg}}$ to generate the group $\mathbb{G}$ for ElGamal.)
- A Groth-Sahai proof system for $\mathcal{P}$ (see Definition 2.5), given by algorithms HGen, BGen, Com, Prove, Verify, Sim.

Public parameters. $\mathrm{SPars}(1^k)$ samples group parameters

$$gpp = (\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, p, g, \hat{g}, e) \leftarrow \mathcal{P}(1^k)$$

and sets $epp_{\mathrm{eg}} = (\mathbb{G}, p, g)$. Then, SPars generates two binding Groth-Sahai CRSs and two ElGamal keypairs:

$$\mathrm{CRS}_1 \leftarrow \mathrm{BGen}(gpp) \qquad (pk_0, sk_0) \leftarrow \mathrm{EGen}_{\mathrm{eg}}(epp_{\mathrm{eg}})$$
$$\mathrm{CRS}_2 \leftarrow \mathrm{BGen}(gpp) \qquad (pk_1, sk_1) \leftarrow \mathrm{EGen}_{\mathrm{eg}}(epp_{\mathrm{eg}}).$$

The public parameters are then defined as

$$spp = (gpp, \mathrm{CRS}_1, \mathrm{CRS}_2, pk_0, pk_1).$$

Key generation. $\mathrm{SGen}(spp)$ first sets up the exponents

$$Z = X \leftarrow \mathbb{Z}_p^* \qquad \text{and} \qquad \alpha = \beta = 0,$$

and commits to them using fresh random coins R_Z, R_α, R_β:

$$C_\alpha \leftarrow \mathrm{Com}(\mathrm{CRS}_1, \alpha; R_\alpha),\ C_\beta \leftarrow \mathrm{Com}(\mathrm{CRS}_1, \beta; R_\beta),\ C_Z \leftarrow \mathrm{Com}(\mathrm{CRS}_2, Z; R_Z).$$

We will use that α, β define an affine function $f : \mathbb{Z}_p \to \mathbb{Z}_p$ through $f(x) = \alpha \cdot x + \beta \bmod p$.

Verification and signing key are given by

$$vk = (C_Z, C_\alpha, C_\beta) \qquad sigk = (X, R_Z, R_\alpha, R_\beta).$$

Signature generation. $\mathrm{Sig}(sigk, M)$, for $M \in \mathbb{Z}_p$, picks fresh random coins R and encrypts

$$C_0 = \mathrm{Enc}_{\mathrm{eg}}(pk_0, g^{Z_0}; R) \qquad C_1 = \mathrm{Enc}_{\mathrm{eg}}(pk_1, g^{Z_1}; R)$$

for $Z_0 = Z_1 = X \in \mathbb{Z}_p$, using the same coins R in both encryptions for efficiency. Then, Sig generates proofs π_1 and π_2 for the respective statements

$$\Big(\underbrace{Z_0 = Z_1}_{S1} \quad \vee \quad \underbrace{f(M) \in \mathrm{QR}_p \cup \{0\}}_{S2}\Big) \qquad \text{and} \qquad \underbrace{Z_0 = Z}_{S3}. \tag{3}$$

Here, Z_0, Z_1, Z, f refer to the values encrypted (resp. committed to) in $C_0, C_1, C_Z, (C_\alpha, C_\beta)$. Concretely, Sig generates a proof π_1 for $S1 \vee S2$ under CRS_1, using as witness $Z_0 = Z_1 = X$ and the encryption coins R. Also, Sig computes a proof π_2 for S3 under CRS_2, using as witness X and R_Z, R. We stress that π_1 and π_2 are independently generated, with different (fresh) Groth-Sahai commitments to the respective witnesses. We describe the exact Groth-Sahai equations for these proofs in Appendix B.1, and give some intuition on the meaning of the statements S1-S3 in Section 3.2 below.

The signature is then defined as

$$\sigma = (C_0, C_1, \pi_1, \pi_2).$$

Verification. $\mathrm{Ver}(spp, vk, M, \sigma)$ outputs 1 if and only if both proofs π_1 and π_2 in σ are valid with respect to $M, C_0, C_1, C_Z, C_\alpha, C_\beta$.

Correctness. The completeness of Groth-Sahai proofs implies the correctness of SIG.

Efficiency. SIG has the following efficiency characteristics (cf. Appendix B.1):

- The public parameters consist of 8 $\mathbb{G}$- and 6 $\hat{\mathbb{G}}$-elements, plus the group parameters *gpp*.
- Each verification key contains 2 $\mathbb{G}$- and 4 $\hat{\mathbb{G}}$-elements.
- Each signing key contains 7 $\mathbb{Z}_p$-exponents.
- Each signature contains 11 $\mathbb{G}$- and 14 $\hat{\mathbb{G}}$-elements.

3.2 Security proof

More details on the role of π_1 and π_2 in signatures. Before we proceed to the proof, we give some intuition on the proofs π_1 and π_2 published in signatures (and the statements S1-S3):

- π_1 proves that *either* C_0 and C_1 encrypt the same value *or* that the signed message satisfies a special property S2 (or both). In the scheme, all messages are special in this sense (because $f(M) = 0$ for all M). However, in the proof, we can adjust f and, e.g., partition the set of messages into special and non-special ones in a random and roughly balanced way. Intuitively, this provides a means to make the double encryption (C_0, C_1) inconsistent (and subsequently change the encrypted values) in signatures for special messages. At the same time, any valid adversarial forgery on a *non-special* message (that does not satisfy S2) must carry a consistent double encryption (C_0, C_1).
- In the scheme, π_2 ties the plaintext encrypted in C_0 to the master secret Z. In the simulation, we will remove that connection by simulating π_2. Specifically, recall that π_1 and π_2 are independently generated, using independently generated Groth-Sahai commitments to the respective witnesses. Thus, in the proof, we can simulate π_2 without witness (by choosing a hiding CRS_2 and using Sim), while preserving the soundness of π_1 (assuming CRS_1 is binding). This simulation of π_2 will be instrumental in changing the message encrypted in C_0 (when the signed message is special in the above sense).

Theorem 3.1 (Security of SIG)**.** *Under the DDH assumptions in $\mathbb{G}$ and $\hat{\mathbb{G}}$, the signature scheme* SIG *from Section 3.1 is EUF-mCMA secure. Concretely, for every EUF-mCMA adversary $\mathcal{A}$ on* SIG*, there exist DDH adversaries $\mathcal{B}$ and $\mathcal{B}'$ (of roughly the same complexity as the EUF-mCMA experiment with $\mathcal{A}$ and* SIG*) with*

$$\mathrm{Adv}^{\text{euf-mcma}}_{\text{SIG},\mathcal{A}}(k) \;\leq\; (8n+1)\cdot\left|\mathrm{Adv}^{\text{ddh}}_{\mathbb{G},\mathcal{B}}(k)\right| + (4n+1)\cdot\left|\mathrm{Adv}^{\text{ddh}}_{\hat{\mathbb{G}},\mathcal{B}'}(k)\right| + \mathbf{O}(n/2^k) \tag{4}$$

for $n = 2\lceil \log_2(p)\rceil + k$, where p denotes the order of $\mathbb{G}$ and $\hat{\mathbb{G}}$, and k is the security parameter.

Proof outline. The proof starts with a number of preparations for the core argument. Our main goal during this phase will be to implement an additional and explicit check of $\mathcal{A}$'s forgery $\sigma^* = (C_0^*, C_1^*, \pi_1^*, \pi_2^*)$ for $\mathrm{Dec}_{\text{eg}}(sk_0, C_0^*) = g^{X^*}$. (Note that in the default key setup, this explicit check is redundant, since valid signatures *must* fulfill statement S3 from (3).)

In the core argument (from Game 4 to Game 5, detailed in Lemma 3.2), we replace the value X used in generated signatures and the additional forgery check with a value $\mathcal{H}(M)$ that depends on the signed message. We start with a constant function $\mathcal{H}(M) = X$ (which corresponds to Game 4), and then introduce more and more dependencies of $\mathcal{H}(M)$ on the Legendre symbols $\left(\frac{f_j(M)}{p}\right)$ for independently and randomly selected (invertible) affine functions f_j.

Each such dependency is introduced as follows. We start by committing to (the coefficients of) a new random function f^* in C_α, C_β. This change allows us to modify the messages Z_0, Z_1 encrypted in generated signatures for all M with $f^*(M) \in QR_p \cup \{0\}$ (and only for those M), by proving S2 (and not S1) in signatures. We will also abort if $\mathcal{A}$'s forgery satisfies $f^*(M^*) \in QR_p \cup \{0\}$, and we will keep enforcing our forgery check on C_0^*. Hence, from $\mathcal{A}$'s point of view, an additional dependency on $\left(\frac{f^*(M)}{p}\right)$ is consistently introduced on *all* signatures. More importantly, this dependency is also enforced during the additional forgery check.

After sufficiently many such dependencies are introduced (for several different f^*), all signatures are consistently generated with (or checked for) $Z_0 = Z_1 = \mathcal{R}(M)$ for a truly random

function $\mathcal{R}$. At this point, $\mathcal{A}$ has to predict a truly random function $\mathcal{R}$ on a fresh input M^* in order to produce a valid forgery. Hence, $\mathcal{A}$'s forgery success must be negligible.

Figs. 1 and 2 (on page 21 and page 22) give a more technical summary of the game transitions of the proof (also taking into account the notation for the multi-user case).

Proof. We proceed in games. Let out_i denote the output of Game i.

Game 1 is the original EUF-mCMA game with $\mathcal{A}$ and SIG. Of course,

$$\Pr[out_1 = 1] \;=\; \mathrm{Adv}^{\text{euf-mcma}}_{\mathrm{SIG},\mathcal{A}}(k). \tag{5}$$

In the following, we apply a superscript to variables to denote to which SIG instance they belong. For instance, we denote with $X^{(\ell)}$ and $sk_0^{(\ell)}, sk_1^{(\ell)}$ the respective values from the ℓ-th used SIG instance. Furthermore, we write X^* for $X^{(\ell^*)}$ for the challenge instance ℓ^* selected by $\mathcal{A}$ for his forgery, and similarly for sk_0^* and sk_1^*.

Thus, in **Game** 2, we implement an additional "forgery check". Concretely, we only consider a forgery $\sigma^* = (C_0^*, C_1^*, \pi_1^*, \pi_2^*)$ from $\mathcal{A}$ as valid if π_1^* and π_2^* are valid *and* if $\mathrm{Dec}_{\mathrm{eg}}(sk_0^*, C_0^*) = g^{X^*}$. (Otherwise, the game outputs 0.) This change is purely conceptual: indeed, since CRS_2 is binding, we can use the soundness of Groth-Sahai proofs. Thus, any valid proof π_2^* guarantees that S3 (from (3)) holds, and so $\mathrm{Dec}_{\mathrm{eg}}(sk_0^*, C_0^*) = g^{X^*}$. We obtain

$$\Pr[out_2 = 1] \;=\; \Pr[out_1 = 1]\,. \tag{6}$$

In **Game** 3, we generate both CRS_1 and CRS_2 as hiding CRSs, using HGen. The CRS indistinguishability of Groth-Sahai proofs yields

$$\Pr[out_3 = 1] - \Pr[out_2 = 1] \;=\; \mathrm{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{B}_3}(k) + \mathrm{Adv}^{\mathrm{ddh}}_{\hat{\mathbb{G}},\mathcal{B}_3'}(k) \tag{7}$$

for suitable DDH adversaries $\mathcal{B}_3$ and $\mathcal{B}_3'$. (Here, we use the re-randomizability of DDH tuples. This enables a reduction that loses only a factor of 1 instead of 2.)

In **Game** 4, we simulate all proofs π_2 in signatures generated for $\mathcal{A}$, using the Groth-Sahai simulator Sim (on input the random coins R_{CRS} used to prepare CRS). We also generate the corresponding commitments C_Z in all verification keys as $C_Z \leftarrow \mathrm{Com}(\mathrm{CRS}_2, 1)$. We stress that all $X^{(\ell)}$ are still chosen randomly, and all signatures are generated with encryptions C_0, C_1 of $X^{(\ell)}$. By the simulation property of Groth-Sahai proofs (see Theorem 2.6 and the following comment concerning the reuse of commitments), these changes do not affect $\mathcal{A}$'s view:

$$\Pr[out_4 = 1] \;=\; \Pr[out_3 = 1]\,. \tag{8}$$

In **Game** 5, we change the generation of signatures *and* the forgery check from Game 2 as follows. To describe these changes, let $\mathcal{R}^{(\ell)} : \mathbb{Z}_p \to \mathbb{Z}_p^*$ (for all scheme instances $\ell \in [n_U]$) be truly random functions. Our changes in Game 5 are then as follows:

- All signatures generated for $\mathcal{A}$ contain encryptions C_0, C_1 of exponents $Z_0 = Z_1 = \mathcal{R}^{(\ell)}(M)$ (encoded as g^{Z_0}, g^{Z_1}) instead of $Z_0 = Z_1 = X^{(\ell)}$, where M is the signed message. As in Game 4, the corresponding proof π is generated using witnesses for S1 and S3 from (3).
- Any forgery $\sigma^* = (C_0^*, C_1^*, \pi_1^*, \pi_2^*)$ for a (fresh) message M^* from $\mathcal{A}$ is considered valid only if π_1^* and π_2^* are valid *and* $\mathrm{Dec}_{\mathrm{eg}}(sk_0^*, C_0^*) = \mathcal{R}^*(M^*)$ holds. Otherwise, the game outputs 0. (Again, we use the shorthand notation $\mathcal{R}^* = \mathcal{R}^{(\ell^*)}$ for the challenge instance ℓ^*.)

In particular, the second change implies that

$$\Pr[out_5 = 1] \;\le\; 1/(p-1) \;\le\; 1/2^k, \tag{9}$$

since $\mathcal{R}^*(M^*)$ is information-theoretically hidden from $\mathcal{A}$.

Hence, it remains to relate Game 4 and Game 5:

Lemma 3.2. *For* $n = 2\lceil \log_2(p) \rceil + k$ *and suitable DDH adversaries* $\mathcal{B}_5$ *and* $\mathcal{B}_5'$*, we have*

$$\left| \Pr[out_5 = 1] - \Pr[out_4 = 1] \right| \leq 8n \cdot \left| \mathrm{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{B}_5}(k) \right| + 4n \cdot \left| \mathrm{Adv}^{\mathrm{ddh}}_{\hat{\mathbb{G}},\mathcal{B}_5'}(k) \right| + \mathbf{O}(n/2^k). \tag{10}$$

Before we prove Lemma 3.2, we remark that putting together (5-10), we obtain (4), which is sufficient to show Theorem 3.1. □

Proof of Lemma 3.2. We will consider a series of hybrid games between Game 4 and Game 5. Concretely, Game 4.i (for $i \geq 0$) is defined like Game 4, except for the following changes:

- We initially uniformly and independently choose i invertible affine functions $f_j : \mathbb{Z}_p \to \mathbb{Z}_p$ (for $j \in [i]$). The f_j define a "partial fingerprint" function $\mathcal{L}_i : \mathbb{Z}_p \to \{-1, 0, 1\}^i$ through

$$\mathcal{L}_i(M) = \left(\left(\frac{f_1(M)}{p} \right), \ldots, \left(\frac{f_i(M)}{p} \right) \right). \tag{11}$$

  For every scheme instance $\ell \in [n_U]$, let $\mathcal{H}_i^{(\ell)} : \mathbb{Z}_p \to \mathbb{Z}_p^*$ be the composition of $\mathcal{L}_i$ with a truly random function $\mathcal{R}_i^{(\ell)} : \{-1, 0, 1\}^i \to \mathbb{Z}_p^*$ (so that $\mathcal{H}_i^{(\ell)}(M) = \mathcal{R}_i^{(\ell)}(\mathcal{L}_i(M))$).
- Signatures for $\mathcal{A}$ contain encryptions C_0, C_1 of exponents $Z_0 = Z_1 = \mathcal{H}_i^{(\ell)}(M)$.
- Any forgery $\sigma^* = (C_0^*, C_1^*, \pi_1^*, \pi_2^*)$ for a (fresh) message M^* from $\mathcal{A}$ is considered valid only if π_1^* and π_2^* are valid *and* $\mathrm{Dec}_{\mathrm{eg}}(sk_0^*, C_0^*) = \mathcal{H}_i^{(\ell)}(M^*)$.

Note that every $\mathcal{H}_0^{(\ell)}$ is a constant function that maps every input M to the same random value. Hence, Game 4.0 is identical to Game 4:

$$\Pr[out_{4.0} = 1] = \Pr[out_4 = 1]. \tag{12}$$

Conversely, for large enough i and with high probability, the "fingerprint function" $\mathcal{L}_i$ becomes injective, so that all $\mathcal{H}_i^{(\ell)}$ become independent truly random functions from $\mathbb{Z}_p$ to $\mathbb{Z}_p^*$:

Lemma 3.3. *For* $n = 2\lceil \log_2(p) \rceil + k$*, the function* $\mathcal{L}_n$ *from (11) is injective, except with probability* $1/2^k$ *(over the choice of the invertible affine functions* $f_j : \mathbb{Z}_p \to \mathbb{Z}_p$*).*

We postpone a proof of Lemma 3.3 for now.

Hence, the functions $\mathcal{H}_n^{(\ell)} = \mathcal{R}_n^{(\ell)} \circ \mathcal{L}_n$ used in Game 4.n (for $n = 2\lceil \log_2(p) \rceil + k$) are statistically close to truly random functions $\mathcal{R}^{(\ell)}$ (as used in Game 5):

$$\left| \Pr[out_{4.n} = 1] - \Pr[out_5 = 1] \right| \leq 1/2^k. \tag{13}$$

Thus, we only need to show that there is no detectable difference between Game 4.i and Game 4.$(i+1)$ for any i. We do so using a hybrid argument (i.e., a sequence of games) that interpolates between Game 4.i and Game 4.$(i+1)$. (See Fig. 2 for an overview.)

Concretely, **Game** 4.i.0 is identical to Game 4.i. Thus,

$$\Pr[out_{4.i.0} = 1] = \Pr[out_{4.i} = 1]. \tag{14}$$

In **Game** 4.i.1, we initially choose an invertible affine function $f^* : \mathbb{Z}_p \to \mathbb{Z}_p$ uniformly, and we abort (with output 0) if the message M^* for which $\mathcal{A}$ finally prepares a forgery satisfies $f^*(M^*) \in \mathrm{QR}_p \cup \{0\}$. We stress that f^* is not (yet) committed to in any C_α, C_β, and thus completely hidden from $\mathcal{A}$. Hence, an abort occurs with probability $\frac{p+1}{2p} = \frac{1}{2} + \frac{1}{2p}$, independently of $\mathcal{A}$'s view, so

$$\Pr[out_{4.i.1} = 1] = \left(\frac{1}{2} - \frac{1}{2p} \right) \cdot \Pr[out_{4.i.0} = 1] \geq \frac{1}{2} \cdot \Pr[out_{4.i.0} = 1] - \frac{1}{2p}. \tag{15}$$

In **Game** 4.i.2, we commit to the coefficients f_0^*, f_1^* of the function f^* from Game 4.i.1 in C_α, C_β for all verification keys (instead of the coefficients $\alpha = \beta = 0$). Accordingly, we generate

all signatures for $\mathcal{A}$ by proving statement S2 (and not S1) from (3) whenever possible (i.e., upon all signature queries with $f^*(M) \in QR_p \cup \{0\}$). Since CRS_1 is hiding, we can use the witness-indistinguishability of Groth-Sahai proofs to obtain

$$\Pr[out_{4.i.2} = 1] \;=\; \Pr[out_{4.i.1} = 1]\,. \tag{16}$$

To describe our change in **Game** 4.i.3, recall that in Game 4.i.2, functions $\mathcal{H}_i^{(\ell)}$ is used to determine both the values $Z_0 = Z_1 = \mathcal{H}_i^{(\ell)}(M)$ encrypted in C_0, C_1 upon signature queries, and to implement the forgery check. In Game 4.i.3, we use *three* such functions $\mathcal{H}_i^{(\ell)}, \mathcal{Z}_i^{(\ell)}, \mathcal{Q}_i^{(\ell)} : \mathbb{Z}_p \to \mathbb{Z}_p^*$. Each of these functions is defined like $\mathcal{H}_i^{(\ell)}$, for the same fingerprint function $\mathcal{L}_i$, but with different (i.e., independently chosen) random functions $\mathcal{R}_i^{(\ell)}$. (In other words, we can write $\mathcal{H}_i^{(\ell)} = F \circ \mathcal{L}_i$, and $\mathcal{Z}_i^{(\ell)} = F' \circ \mathcal{L}_i$, and $\mathcal{Q}_i^{(\ell)} = F'' \circ \mathcal{L}_i$ for independently random functions $F, F', F'' : \{-1, 0, 1\}^i \to \mathbb{Z}_p^*$. Intuitively, thus, $\mathcal{Z}_i^{(\ell)}$ and $\mathcal{Q}_i^{(\ell)}$ are "decoupled copies" of $\mathcal{H}_i^{(\ell)}$.)

Our goal will be to use the functions $\mathcal{H}_i^{(\ell)}, \mathcal{Z}_i^{(\ell)}, \mathcal{Q}_i^{(\ell)}$ for messages M satisfying $f^*(M) \notin QR_p$, $f^*(M) = 0$, and $f^*(M) \in QR_p$, respectively. This will be conceptually identical to using a single function $\mathcal{H}_{i+1}^{(\ell)}$ for all messages of a given scheme instance ℓ. At this point, however, we can only partially implement this strategy, since we can only replace the messages encrypted in C_1, but not those from C_0. (Indeed, sk_0^* is still required to implement the additional forgery check in Game 4.i.3.)

Thus, in Game 4.i.3, for every scheme instance $\ell \in [n_U]$, we use the respective function $\mathcal{H}_i^{(\ell)}$ to generate all ciphertexts C_0, C_1 in signatures (as in Game 4.i.2), with the following exceptions:

- For signature queries with $f^*(M) = 0$, we encrypt $Z_1 = \mathcal{Z}_i^{(\ell)}(M)$ (instead of $Z_1 = \mathcal{H}_i^{(\ell)}(M)$) in the ciphertext C_1 of the generated signature.
- For signature queries with $f^*(M) \in QR_p$, we encrypt $Z_1 = \mathcal{Q}_i^{(\ell)}(M)$ in C_1.

Note that for signatures with $f^*(M) \in QR_p \cup \{0\}$, the random coins used to generate C_1 (or C_0) are not used as a witness in the process of constructing π. Furthermore, no secret key $sk_1^{(\ell)}$ has to be known to the game. A reduction to the (tight) IND-mCPA security of ElGamal yields

$$\sum_{i=0}^{n-1} \Pr[out_{4.i.3} = 1] - \Pr[out_{4.i.2} = 1] \;=\; n \cdot \mathrm{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{B}_{4.i.3}}(k) \tag{17}$$

for a suitable DDH adversary $\mathcal{B}_{4.i.3}$. (We note that even though the random coins R of C_1 are not known explicitly to $\mathcal{B}_{4.i.3}$, a C_0 with reused R can be constructed from $sk_0^{(\ell)}$ and a given g^R.)

Our next step will be to replace the values encrypted in C_0 in a similar way. To do so, however, we need some preparations, since Game 4.i.3 still knows the secret keys $sk_0^{(\ell)}$ (to finally implement the forgery check). Fortunately, however, we can alternatively use the $sk_1^{(\ell)}$ to implement this check. (To see why this yields the same functionality, recall that by our abort rule from Game 1, we may restrict to forgeries with $f^*(M^*) \notin QR_p \cup \{0\}$. However, by (3), a valid forgery for such a message must contain C_0^* and C_1^* that encrypt the same message.)

As a first step, in **Game** 4.i.4, we initially generate a binding CRS CRS_1 (using $CRS_1 \leftarrow \mathrm{BGen}(gpp)$). The CRS indistinguishability of Groth-Sahai proofs ensures that

$$\sum_{i=0}^{n-1} \Pr[out_{4.i.4} = 1] - \Pr[out_{4.i.3} = 1] \;=\; n \cdot \left(\mathrm{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{B}_{4.i.4}}(k) + \mathrm{Adv}^{\mathrm{ddh}}_{\hat{\mathbb{G}},\mathcal{B}'_{4.i.4}}(k)\right) \tag{18}$$

for suitable DDH adversaries $\mathcal{B}_{4.i.4}$ and $\mathcal{B}'_{4.i.4}$.

Next, in **Game** 4.i.5, we implement the forgery check rule from Game 2 using sk_1^* (and not sk_0^*). That is, when $\mathcal{A}$ submits a forgery $\sigma^* = (C_0^*, C_1^*, \pi_1^*, \pi_2^*)$, we check if $\mathrm{Dec}_{eg}(sk_1^*, C_1^*) = \mathcal{H}_i^*(M^*)$ holds (and reject the forgery if not). We may assume that $M^* \notin QR_p \cup \{0\}$ (since

otherwise, we trivially abort anyway). But for such M^*, a valid forgery *must* fulfill S1 from (3), since at this point, CRS_1 is binding. In other words, we have $\mathrm{Dec}_{eg}(sk_1^*, C_1^*) = \mathcal{H}_i^*(M^*)$ if and only if $\mathrm{Dec}_{eg}(sk_0^*, C_0^*) = \mathcal{H}_i^*(M^*)$. Hence, the change in Game 4.i.5 is purely conceptual, and we get:

$$\Pr[out_{4.i.5} = 1] \;=\; \Pr[out_{4.i.4} = 1]\,. \tag{19}$$

Since we no longer use sk_0^* (or the random coins from any C_1 generated upon a signature query), we can continue with our strategy. Specifically, in **Game** 4.i.6, we generate all ciphertexts C_0, C_1 in signatures as follows:

- For queries with $f^*(M) \notin \mathrm{QR}_p$, we encrypt $Z_0 = Z_1 = \mathcal{H}_i^{(\ell)}(M)$ in C_0 and C_1.
- For queries with $f^*(M) = 0$, we encrypt $Z_0 = Z_1 = \mathcal{Z}_i^{(\ell)}(M)$ in C_0 and C_1.
- For queries with $f^*(M) \in \mathrm{QR}_p$, we encrypt $Z_0 = Z_1 = \mathcal{Q}_i^{(\ell)}(M)$ in C_0 and C_1.

Observe that the only difference to Game 4.i.5 is that the messages Z_0 encrypted in ciphertexts C_0 in signatures with $f^*(M) \in \mathrm{QR}_p \cup \{0\}$ are changed. For such encryptions, neither secret key nor random coins are used by the game. Hence, a reduction to the (tight) IND-mCPA security of ElGamal yields

$$\sum_{i=0}^{n-1} \Pr[out_{4.i.6} = 1] - \Pr[out_{4.i.5} = 1] \;=\; n \cdot \mathrm{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{B}_{4.i.6}}(k) \tag{20}$$

for a suitable DDH adversary $\mathcal{B}_{4.i.6}$. (Again, a reuse of random coins between C_0 and C_1 is possible since the secret key sk_1 is known to $\mathcal{B}_{4.i.6}$ during the reduction.)

Now in Game 4.i.6, we handle both signature queries and $\mathcal{A}$'s forgery with either $\mathcal{H}_i^{(\ell)}$, $\mathcal{Z}_i^{(\ell)}$, or $\mathcal{Q}_i^{(\ell)}$, depending on the Legendre symbol $\left(\frac{M}{p}\right)$ of M. This is equivalent to handling all messages with a single function $\mathcal{H}_{i+1}^{(\ell)}$ by the definition of $\mathcal{H}_i^{(\ell)}$ (see also (11)). Hence, we already "almost" implement the rules of Game $4.(i+1)$, and we only need to clean up things a little.

Namely, in **Game** 4.i.7, we again implement the forgery check from Game 2 using sk_0^* (and not sk_1^*). With the same reasoning as in Game 5, we get:

$$\Pr[out_{4.i.7} = 1] \;=\; \Pr[out_{4.i.6} = 1]\,. \tag{21}$$

Next, in **Game** 4.i.8, we again set up CRS_1 as a hiding CRS (using HGen). Again, CRS indistinguishability guarantees

$$\sum_{i=0}^{n-1} \Pr[out_{4.i.8} = 1] - \Pr[out_{4.i.7} = 1] \;=\; n \cdot \left(\mathrm{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{B}_{4.i.8}}(k) + \mathrm{Adv}^{\mathrm{ddh}}_{\hat{\mathbb{G}},\mathcal{B}'_{4.i.8}}(k)\right) \tag{22}$$

for suitable DDH adversaries $\mathcal{B}_{4.i.8}$ and $\mathcal{B}'_{4.i.8}$.

In **Game** 4.i.9, we again set up the commitments C_α, C_β in all verification keys as commitments to $\alpha = \beta = 0$. Accordingly, we generate all signatures for $\mathcal{A}$ by proving statement S1 from (3). (Note that this is possible again since all generated pairs (C_0, C_1) do encrypt the same message.) By the witness-indistinguishability of Groth-Sahai proofs,

$$\Pr[out_{4.i.9} = 1] \;=\; \Pr[out_{4.i.8} = 1]\,. \tag{23}$$

Finally, in **Game** 4.i.10, we do not abort anymore. (That is, we take back the abort rule from Game 1.) To see how this change affects the game's output, we make a few observations. First, note that in both Game 4.i.9 and Game 4.i.10, $\mathcal{A}$'s view only depends on the way f^* partitions the set of messages depending on $\left(\frac{f^*(M)}{p}\right)$, but not on *which* messages M are mapped by f^* to squares, and which to non-squares. (Indeed, any partitioning of the M is invariant under multiplying f^* with an invertible non-square modulo p. However, multiplication with an invertible non-square inverts the Legendre symbol of $f^*(M)$.)

Thus, the probability for $\mathcal{A}$ to successfully forge a signature with $\left(\frac{f^*(M^*)}{p}\right) = 1$ is exactly the same as that to forge a signature with $\left(\frac{f^*(M^*)}{p}\right) = -1$. Hence, if we cease to abort upon $f^*(M^*) \in QR_p \cup \{0\}$, we *at least* double $\mathcal{A}$'s success probability:

$$\Pr\left[out_{4.i.10} = 1\right] \;\geq\; 2 \cdot \Pr\left[out_{4.i.9} = 1\right]. \tag{24}$$

At the same time, Game 4.i.10 is identical to Game $4.(i+1)$. (As argued, the use of three functions $\mathcal{H}_i^{(\ell)}, \mathcal{Z}_i^{(\ell)}, \mathcal{Q}_i^{(\ell)}$ for each scheme instance ℓ is equivalent to the use of a single function $\mathcal{H}_{i+1}^{(\ell)}$ in Game $4.(i+1)$. Furthermore, CRS_1 is hiding, the C_α, C_β are set up as commitments to $\alpha = \beta = 0$, and the signatures use proofs of statement S1.) Thus,

$$\Pr\left[out_{4.i.10} = 1\right] \;=\; \Pr\left[out_{4.(i+1)} = 1\right]. \tag{25}$$

Collecting all differences of probabilities from (14-25), we obtain

$$\begin{aligned}\left|\Pr\left[out_{4.0} = 1\right] - \Pr\left[out_{4.n} = 1\right]\right| \;&\leq\; \left|\sum_{i=0}^{n-1} \Pr\left[out_{4.i} = 1\right] - \Pr\left[out_{4.(i+1)} = 1\right]\right| \\ &\leq\; 8n \cdot \left|\mathrm{Adv}^{\mathrm{ddh}}_{\mathbb{G},\mathcal{B}_5}(k)\right| + 4n \cdot \left|\mathrm{Adv}^{\mathrm{ddh}}_{\hat{\mathbb{G}},\mathcal{B}_5'}(k)\right| + \mathbf{O}(n/2^k)\end{aligned}$$

for DDH adversaries $\mathcal{B}_5$ and $\mathcal{B}_5'$ that combine all adversaries from the collected differences. Together with (12) and (13), we obtain (10). □

It remains to prove Lemma 3.3:

Proof of Lemma 3.3. For any distinct $M_0, M_1 \in \mathbb{Z}_p$ and a uniformly chosen invertible affine function $f : \mathbb{Z}_p \to \mathbb{Z}_p$, we have $\Pr\left[\left(\frac{f(M_0)}{p}\right) = \left(\frac{f(M_1)}{p}\right)\right] \leq 1/2$, since f is pairwise independent. As all f_j from (11) are chosen independently, we get

$$\Pr\left[\mathcal{L}_n(M_0) = \mathcal{L}_n(M_1)\right] \;\leq\; 1/2^n$$

for any two distinct M_0, M_1. A union bound over all $\mathbf{O}(p^2)$ such pairs (M_0, M_1) shows the claim. □

References

[1] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. "Structure-Preserving Signatures and Commitments to Group Elements". In: *Proc. CRYPTO 2010*. Vol. 6223. Lecture Notes in Computer Science. Springer, 2010, pp. 209–236.

[2] Masayuki Abe, Bernardo David, Markulf Kohlweiss, Ryo Nishimaki, and Miyako Ohkubo. "Tagged One-Time Signatures: Tight Security and Optimal Tag Size". In: *Proc. Public Key Cryptography 2013*. Vol. 7778. Lecture Notes in Computer Science. Springer, 2013, pp. 312–331.

[3] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. "Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements". In: *Proc. EUROCRYPT 2000*. Vol. 1807. Lecture Notes in Computer Science. Springer, 2000, pp. 259–274.

[4] Daniel J. Bernstein. "Proving Tight Security for Rabin-Williams Signatures". In: *Proc. EUROCRYPT 2008*. Vol. 4965. Lecture Notes in Computer Science. Springer, 2008, pp. 70–87.

[5] Olivier Blazy, Eike Kiltz, and Jiaxin Pan. "(Hierarchical) Identity-Based Encryption from Affine Message Authentication". In: *Proc. CRYPTO (1) 2014*. Vol. 8616. Lecture Notes in Computer Science. Springer, 2014, pp. 408–425.

[6] Alexandra Boldyreva. "Strengthening Security of RSA-OAEP". in: *Proc. CT-RSA 2009*. Vol. 5473. Lecture Notes in Computer Science. Springer, 2009, pp. 399–413.

[7] Dan Boneh and Xavier Boyen. "Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles". In: *Proc. EUROCRYPT 2004*. Vol. 3027. Lecture Notes in Computer Science. Springer, 2004, pp. 223–238.

[8] Dan Boneh and Xavier Boyen. "Secure Identity Based Encryption Without Random Oracles". In: *Proc. CRYPTO 2004*. Vol. 3152. Lecture Notes in Computer Science. Springer, 2004, pp. 443–459.

[9] Dan Boneh and Matthew K. Franklin. "Identity-Based Encryption from the Weil Pairing". In: *Proc. CRYPTO 2001*. Vol. 2139. Lecture Notes in Computer Science. Springer, 2001, pp. 213–229.

[10] Dan Boneh and Matthew K. Franklin. "Identity-Based Encryption from the Weil Pairing". In: *SIAM J. Comput.* 32.3 (2003), pp. 586–615.

[11] Dan Boneh, Ilya Mironov, and Victor Shoup. "A Secure Signature Scheme from Bilinear Maps". In: *Proc. CT-RSA 2003*. Vol. 2612. Lecture Notes in Computer Science. Springer, 2003, pp. 98–110.

[12] David Cash, Eike Kiltz, and Victor Shoup. "The Twin Diffie-Hellman Problem and Applications". In: *Proc. EUROCRYPT 2008*. Vol. 4965. Lecture Notes in Computer Science. Springer, 2008, pp. 127–145.

[13] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. "Bonsai Trees, or How to Delegate a Lattice Basis". In: *Proc. EUROCRYPT 2010*. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 523–552.

[14] Jie Chen and Hoeteck Wee. "Fully, (Almost) Tightly Secure IBE and Dual System Groups". In: *Proc. CRYPTO (2) 2013*. Vol. 8043. Lecture Notes in Computer Science. Springer, 2013, pp. 435–460.

[15] Benoît Chevallier-Mames and Marc Joye. "A Practical and Tightly Secure Signature Scheme Without Hash Function". In: *Proc. CT-RSA 2007*. Vol. 4377. Lecture Notes in Computer Science. Springer, 2006, pp. 339–356.

[16] Jean-Sébastien Coron. "On the Exact Security of Full Domain Hash". In: *Proc. CRYPTO 2000*. Vol. 1880. Lecture Notes in Computer Science. Springer, 2000, pp. 229–235.

[17] Alex Escala and Jens Groth. "Fine-Tuning Groth-Sahai Proofs". In: *Proc. Public Key Cryptography 2014*. Vol. 8383. Lecture Notes in Computer Science. Springer, 2014, pp. 630–649.

[18] Georg Fuchsbauer. "Commuting Signatures and Verifiable Encryption". In: *Proc. EUROCRYPT 2011*. Vol. 6632. Lecture Notes in Computer Science. Springer, 2011, pp. 224–245.

[19] David Galindo, Sebastià Martín Molleví, Paz Morillo, and Jorge Luis Villar. "Easy Verifiable Primitives and Practical Public Key Cryptosystems". In: *Proc. ISC 2003*. Vol. 2851. Lecture Notes in Computer Science. Springer, 2003, pp. 69–83.

[20] Rosario Gennaro, Shai Halevi, and Tal Rabin. "Secure Hash-and-Sign Signatures Without the Random Oracle". In: *Proc. EUROCRYPT 1999*. Vol. 1592. Lecture Notes in Computer Science. Springer, 1999, pp. 123–139.

[21] Craig Gentry. "Practical Identity-Based Encryption Without Random Oracles". In: *Proc. EUROCRYPT 2006*. Vol. 4004. Lecture Notes in Computer Science. Springer, 2006, pp. 445–464.

[22] Craig Gentry and Shai Halevi. "Hierarchical Identity Based Encryption with Polynomially Many Levels". In: *Proc. TCC 2009*. Vol. 5444. Lecture Notes in Computer Science. Springer, 2009, pp. 437–456.

[23] Eu-Jin Goh, Stanislaw Jarecki, Jonathan Katz, and Nan Wang. "Efficient Signature Schemes with Tight Reductions to the Diffie-Hellman Problems". In: *J. Cryptology* 20.4 (2007), pp. 493–514.

[24] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "On the Cryptographic Applications of Random Functions". In: *Proc. CRYPTO 1984*. Vol. 196. Lecture Notes in Computer Science. Springer, 1985, pp. 276–288.

[25] Jens Groth and Amit Sahai. "Efficient Noninteractive Proof Systems for Bilinear Groups". In: *SIAM J. Comput.* 41.5 (2012), pp. 1193–1232.

[26] Dennis Hofheinz. "All-But-Many Lossy Trapdoor Functions". In: *Proc. EUROCRYPT 2012*. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 209–227.

[27] Dennis Hofheinz and Tibor Jager. "Tightly Secure Signatures and Public-Key Encryption". In: *Proc. CRYPTO 2012*. Vol. 7417. Lecture Notes in Computer Science. Springer, 2012, pp. 590–607.

[28] Susan Hohenberger and Brent Waters. "Short and Stateless Signatures from the RSA Assumption". In: *Proc. CRYPTO 2009*. Vol. 5677. Lecture Notes in Computer Science. Springer, 2009, pp. 654–670.

[29] Saqib A. Kakvi and Eike Kiltz. "Optimal Security Proofs for Full Domain Hash, Revisited". In: *Proc. EUROCRYPT 2012*. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 537–553.

[30] Benoît Libert, Marc Joye, Moti Yung, and Thomas Peters. "Concise Multi-challenge CCA-Secure Encryption and Signatures with Almost Tight Security". In: *Proc. ASIACRYPT (2) 2014*. Vol. 8874. Lecture Notes in Computer Science. Springer, 2014, pp. 1–21.

[31] Moni Naor and Omer Reingold. "Number-theoretic Constructions of Efficient Pseudo-random Functions". In: *Proc. FOCS 1997*. IEEE Computer Society, 1997, pp. 458–467.

[32] Moni Naor and Moti Yung. "Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks". In: *Proc. STOC 1990*. ACM, 1990, pp. 427–437.

[33] Moni Naor and Moti Yung. "Universal One-Way Hash Functions and their Cryptographic Applications". In: *Proc. STOC 1989*. ACM, 1989, pp. 33–43.

[34] Sven Schäge. "Tight Proofs for Signature Schemes without Random Oracles". In: *Proc. EUROCRYPT 2011*. Vol. 6632. Lecture Notes in Computer Science. Springer, 2011, pp. 189–206.

[35] Brent Waters. "Efficient Identity-Based Encryption Without Random Oracles". In: *Proc. EUROCRYPT 2005*. Vol. 3494. Lecture Notes in Computer Science. Springer, 2005, pp. 114–127.

A Compact and (almost) tightly secure public-key encryption

Our signature scheme SIG from Section 3 is "almost" automorphic (in the sense of [1]). Namely, while its verification can be expressed as a system of equations that is compatible with Groth-Sahai proofs, its messages are exponents (as opposed to group elements). However, our scheme can still be used in the generic construction of [27]. This yields an (almost) tightly secure public-key encryption scheme with compact parameters, keys and ciphertexts. (Here, "compact" means "comprised of only a constant number of group elements or exponents.")

But although compact in the above sense, the resulting encryption scheme would be rather inefficient (in particular since it would use nested Groth-Sahai proofs). Thus, here we describe an optimized and more compact (almost) tightly secure public-key encryption scheme PKE.

Setting and ingredients. The basis for our PKE construction is the signature scheme SIG from Section 3, and we assume similar ingredients. In particular, we assume groups $\mathbb{G}$ and $\hat{\mathbb{G}}$, along with the ElGamal encryption and Groth-Sahai proofs over $\mathbb{G}$. Additionally, we assume:

- An OT-EUF-mCMA secure signature scheme with message space $\mathbb{Z}_p$, given by algorithms OPars, OGen, OSig, OVer. For concreteness, in all of the following, we assume the one-time signature scheme TOTS from [27] in $\mathbb{G}$. Its OT-EUF-mCMA security can be tightly reduced to the discrete logarithm assumption in $\mathbb{G}$ (which is implied by the DDH assumption in $\mathbb{G}$).
- A generator $\mathcal{H}$ of collision-resistant hash functions $H : \{0,1\}^* \to \{0,1\}^k$. We will interpret H-outputs as $\mathbb{Z}_p$-elements in the natural way. (Recall that $p > 2^k$.)

All ingredients can be instantiated under the DDH assumptions in $\mathbb{G}$ and $\hat{\mathbb{G}}$.

Public parameters. $\mathrm{EPars}(1^k)$ first proceeds like the parameter generation of SIG, and samples group parameters *gpp*, a hiding Groth-Sahai CRS, and two ElGamal public keys pk_0, pk_1. Then, EPars sets up exponents Z, α, β and *ciphertexts*

$$C_\alpha \leftarrow \mathrm{Enc}_{\mathrm{eg}}(pk_0, g^\alpha; R_\alpha),\ C_\beta \leftarrow \mathrm{Enc}_{\mathrm{eg}}(pk_0, g^\beta; R_\beta),\ C_Z \leftarrow \mathrm{Enc}_{\mathrm{eg}}(pk_0, g^Z; R_Z).$$

Note that here, we encrypt (and do not commit to) Z, α, β in order to be able to produce slightly more compact proofs involving Z, α, β later on. However, we note that conceptually, we could have as well committed to Z, α, β as with SIG.

Finally, EPars chooses parameters $opp \leftarrow \mathrm{OPars}(1^k)$ and a hash function H, and outputs

$$epp = (gpp, \mathrm{CRS}, pk_0, pk_1, opp, H, C_\alpha, C_\beta, C_Z).$$

Key generation. $\mathrm{EGen}(epp)$ samples two ElGamal keypairs

$$(pk_0', sk_0') \leftarrow \mathrm{EGen}_{\mathrm{eg}}(\mathbb{G}, p, g) \qquad (pk_1', sk_1') \leftarrow \mathrm{EGen}_{\mathrm{eg}}(\mathbb{G}, p, g),$$

and outputs a public and a secret key as

$$pk = (pk_0', pk_1') \qquad sk = (d, sk_d')$$

for a uniformly chosen bit $d \leftarrow \{0,1\}$.

Encryption. Intuitively, encryption corresponds to a Naor-Yung style double encryption with consistency proof [32]. The consistency proof itself proceeds as in [27], and essentially proves that either the double encryption is consistent, or a signature to a fresh value is known. (A suitable fresh value will be hash of a freshly sampled verification key of the one-time signature scheme.) Concretely, $\mathrm{Enc}(pk, M)$, for $M \in \mathbb{G}$, chooses a one-time signature keypair

$$(ovk, osk) \leftarrow \mathrm{OGen}(opp),$$

and encrypts the values $Z_0' = Z_1' = M \in \mathbb{G}$ and $Z_0 = Z_1 = 0$ as

$$C_0' = \mathrm{Enc}_{\mathrm{eg}}(pk_0', Z_0'; R') \qquad C_0 = \mathrm{Enc}_{\mathrm{eg}}(pk_0, g^{Z_0}; R)$$
$$C_1' = \mathrm{Enc}_{\mathrm{eg}}(pk_1', Z_1'; R') \qquad C_1 = \mathrm{Enc}_{\mathrm{eg}}(pk_1, g^{Z_1}; R).$$

(Note that for efficiency and to simplify proofs involving these values, we reuse the encryption random coins R' and R.) Then, Enc generates a proof π (under CRS) of the statement

$$Z'_0 = Z'_1 \ \vee \ \Big(\big(Z_0 = Z_1 \ \vee \ f(H(\mathit{ovk})) \in QR_p \cup \{0\} \big) \ \wedge \ \big(Z_0 = Z \ \vee \ Z = 0 \big) \Big). \tag{26}$$

Enc will prove the left branch $S1'$ of the outer $\vee$ clause, using as witness the encryption randomness R'. Hence, π essentially proves consistency of C'_0, C'_1, *or* the same statement as for a SIG-signature for $H(\mathit{ovk})$. (There are some slight differences compared to a SIG-signature: first, we use only one CRS. Hence, we cannot simulate proofs for substatement $Z_0 = Z$ during the proof. Instead, however, we can set $Z = 0$ to be able to generate proofs for $S3'$ without knowledge of Z_0. Second, because the random coins used for C_α, C_β, C_Z are not known at encryption time, the proof of quadratic residuosity becomes somewhat less efficient than the one in SIG's signing algorithm. We refer to Appendix B.2 for more details on the exact proof equations.)

Finally, Enc signs

$$\sigma \leftarrow \mathrm{OSig}(\mathit{osk}, H(C'_0, C'_1, C_0, C_1, \pi))$$

and outputs the ciphertext

$$C = (C'_0, C'_1, C_0, C_1, \pi, \mathit{ovk}, \sigma).$$

Decryption. $\mathrm{Dec}(\mathit{sk}, C)$ checks the validity of σ and π. If both σ and π are valid, Dec outputs $M \leftarrow \mathrm{Dec}_{eg}(\mathit{sk}'_d, C'_d)$; otherwise, Dec outputs $\perp$.

Efficiency. PKE has the following efficiency characteristics (cf. Appendix B.2):

- The public parameters consist of 12 $\mathbb{G}$- and 3 $\hat{\mathbb{G}}$-elements, plus the group parameters *gpp*, and a description of the hash function H.
- Each public key contains 2 $\mathbb{G}$-elements.
- Each secret key contains one $\mathbb{Z}_p$-exponent and a bit.
- Each ciphertext contains 27 $\mathbb{G}$- and 30 $\hat{\mathbb{G}}$-elements, and 3 $\mathbb{Z}_p$-exponents.

Theorem A.1 (Security of PKE)**.** *Under the DDH assumptions in $\mathbb{G}$ and $\hat{\mathbb{G}}$, and assuming that* H *is collision-resistant, the PKE scheme* PKE *described above is IND-mCCA secure. The corresponding security reductions lose only a multiplicative factor of $\mathbf{O}(k)$, where k is the security parameter.*

Proof sketch. The proof combines the strategy from [27] with our concrete signature scheme. This strategy proceeds in games, and modifies an IND-mCCA attack with adversary $\mathcal{A}$ as follows:

- First, the consistency proofs in all ciphertexts are prepared with different witnesses. More specifically, instead of proving $Z'_0 = Z'_1$, we prove the right branch of (26). (Note that this right branch corresponds to the validity of a SIG-signature for message $H(\mathit{ovk})$.) Thanks to the witness-indistinguishability of Groth-Sahai proofs, this change is not detectable by $\mathcal{A}$.
- Next, all challenge ciphertexts generated for $\mathcal{A}$ are made inconsistent. (This is possible since the ciphertext consistency proofs are prepared from signature witnesses now.) Concretely, recall that so far we have encrypted the respective challenge message M^*_b (for the secret bit b chosen by the IND-mCCA experiment) in both C'_0 and C'_1 of all challenge ciphertexts. Now we encrypt M^*_b in C'_d and M^*_{1-b} in C'_{1-d}, where d is the bit chosen for the respective PKE instance i. Hence, we change the encrypted message for all ElGamal instances whose secret key is not used. Since only the secret keys sk'_d (but not the sk'_{1-d}) are used in the experiment, this game modification can be justified with the (tight) security of ElGamal.
- We now reject all inconsistent (in the sense $\mathrm{Dec}_{eg}(\mathit{sk}'_0, C'_0) \neq \mathrm{Dec}_{eg}(\mathit{sk}'_1, C'_1)$) decryption queries from $\mathcal{A}$. At this point in the proof, we know both sk'_0 and sk'_1 for all PKE-instances, and can thus recognize the first inconsistent (in the above sense) decryption query with a valid consistency proof. Note that any such query implies a valid SIG-signature for a message $H(\mathit{ovk})$. The security of the one-time signature scheme guarantees that this message is fresh, so that $\mathcal{A}$ has essentially forged a SIG-signature. Any such forgery can be excluded with the same strategy as in the proof of Theorem 3.1 (with the differences described above).

At this point, $\mathcal{A}$ gets no information about the IND-mCCA secret b anymore. Namely, each challenge ciphertext contains ElGamal encryptions of both M_0^* and M_1^*, in an order determined by $d \oplus b$, where d denotes which ElGamal secret key sk'_d the experiment uses to decrypt for this instance. Now since inconsistent ciphertexts are rejected, the game's answer to $\mathcal{A}$'s decryption queries does not depend on the any of the bits d. Moreover, unless (any) d is known, also b is hidden. Hence, $\mathcal{A}$'s view is now completely independent of b, and thus $\mathcal{A}$'s IND-mCCA success is zero. □

B Details on the exact Groth-Sahai equations in our schemes

B.1 The exact Groth-Sahai equations for the proofs in signatures

We now give details on the proofs π_1 and π_2 in signatures from SIG. Recall that π_1 and π_2 shall prove the respective statements

$$\Big(\underbrace{Z_0 = Z_1}_{S1} \quad \vee \quad \underbrace{f(M) \in QR_p \cup \{0\}}_{S2}\Big) \qquad \text{and} \qquad \underbrace{Z_0 = Z}_{S3}. \tag{27}$$

The statements S1-S3. We now discuss the three individual statements S1-S3 from (27) in more detail. To this end, let us write the ElGamal ciphertexts C_0, C_1 from a signature as

$$C_0 = (A, B_0) = (g^R, pk_0^R \cdot g^{Z_0}) \qquad\qquad C_1 = (A, B_1) = (g^R, pk_1^R \cdot g^{Z_1}).$$

(Of course, the reused value $A = g^R$ will only appear once in a signature.)

S1. The statement $Z_0 = Z_1$ holds if and only if $(g, pk_1/pk_0, A, B_1/B_0)$ is a Diffie-Hellman tuple. Thus, S1 is equivalent to the equations $A = g^R$ and $B_1/B_0 = (pk_1/pk_0)^R$, with witness R.

S2. The statement $f(M) \in QR_p \cup \{0\}$ is equivalent to the existence of an exponent $W \in \mathbb{Z}_p$ with $f(M) = W^2 \bmod p$. (Recall that a commitment to $f(M)$ can be homomorphically computed from M and the commitments C_α, C_β.) Hence, a witness to S2 is given by (α, β, W).

S3. We can express $Z_0 = Z$ as an equation $B_0 = pk_0^R \cdot g^Z$ with witness (R, Z).

All involved commitment random coins are additionally required to construct a valid proof. Besides, so far we have neglected that in a setting with an asymmetric pairing, not all combinations of, e.g., $\mathbb{Z}_p$-products can be directly expressed. (For instance, a square W^2 needs to be rephrased as $W \cdot \widehat{W}$, with an additional proof that $W = \widehat{W}$.) Hence, in the rest of this section, we will decorate variables that correspond to a $\hat{\mathbb{G}}$-commitment with a hat (e.g., $\widehat{W}$).

The equations for π_1. Equations for the disjunction S1 $\vee$ S2 can be derived using standard techniques. However, if we optimize a little, we obtain the following equations for S1 $\vee$ S2:

$$A^{\widehat{U}} = g^{\widehat{V}} \qquad (B_1/B_0)^{\widehat{U}} = (pk_1/pk_0)^{\widehat{V}} \qquad \widehat{f(M)} = W \cdot \widehat{W} \qquad W = \widehat{W} + \widehat{U}.$$

(For instance, if we want to prove S2, we can set $\widehat{U} = \widehat{V} = 0$ and $W = \widehat{W}$ such that $f(M) = W^2$.) The involved variables from the verification key are $\widehat{\alpha}$ and $\widehat{\beta}$ (used to homomorphically construct $\widehat{f(M)}$). The variables whose commitments are placed in the signature are $\widehat{U}, \widehat{V}, W, \widehat{W}$. All of these variables are committed to using CRS_1.

The equations for π_2. Similarly, we obtain the following equations for S3:

$$A = g^{\widehat{S}} \qquad\qquad B_0 = pk_0^{\widehat{S}} \cdot g^Z.$$

The variables are Z (committed to in vk) and $\widehat{S}$ (from σ), both committed to using CRS_2.

Remarks and efficiency summary. We emphasize that hence, the proofs π_1 and π_2 are independent (and in particular do not share commitments). Furthermore, thanks to the composability of Groth-Sahai proofs, the commitments C_α, C_β, C_Z to α, β, Z that are placed in the

verification key can be directly (re-)used in proofs. Each commitment occupies 2 group elements. In total, the equations above comprise 4 linear equations over $\mathbb{G}$, and 2 quadratic equations over $\mathbb{Z}_p$. Thus, π_1 contains $4 \cdot 2 + 2 \cdot 1 + 2 \cdot 4 = 18$ group elements (12 of them from $\hat{\mathbb{G}}$), and π_2 contains $1 \cdot 2 + 2 \cdot 1 = 4$ group elements (2 of them from $\hat{\mathbb{G}}$).

B.2 The exact Groth-Sahai equations for the proofs in ciphertexts

We now detail the proof π in ciphertexts from PKE. Recall that π shall prove the statement

$$\underbrace{Z_0' = Z_1'}_{S1'} \vee \Big(\big(\underbrace{Z_0 = Z_1}_{S2'} \vee \underbrace{f(H(ovk)) \in QR_p \cup \{0\}}_{S3'} \big) \wedge \big(\underbrace{Z_0 = Z}_{S4'} \vee \underbrace{Z = 0}_{S5'} \big) \Big). \tag{28}$$

The variables in (28) refer to the messages encrypted in $\mathrm{PKE}_{\mathrm{eg}}$-ciphertexts from the public parameters and the PKE-ciphertext at hand. We make these $\mathrm{PKE}_{\mathrm{eg}}$-ciphertexts explicit as

$$\begin{aligned} C_0 &= \mathrm{Enc}_{\mathrm{eg}}(pk_0, g^{Z_0}; R) = (A, B_0) & C_0' &= \mathrm{Enc}_{\mathrm{eg}}(pk_0', g^{Z_0'}; R') = (A', B_0') \\ C_1 &= \mathrm{Enc}_{\mathrm{eg}}(pk_1, g^{Z_1}; R) = (A, B_1) & C_1' &= \mathrm{Enc}_{\mathrm{eg}}(pk_1', g^{Z_1'}; R') = (A', B_1') \\ C_Z &= \mathrm{Enc}_{\mathrm{eg}}(pk_0, g^{Z}; R_Z) = (A_Z, B_Z). \end{aligned}$$

Besides, a $\mathrm{PKE}_{\mathrm{eg}}$-ciphertext $C_f = \mathrm{Enc}_{\mathrm{eg}}(pk_0, g^{f(H(ovk))}; R_f) = (A_f, B_f)$ that determines the variable $f(H(ovk))$ can be homomorphically computed from the ciphertexts C_α, C_β, and $H(ovk)$.

The statements S1′-S5′. Let us take a closer look at the individual statements S1′-S5′:

S1′, S2′. These statements can be formalized like statement S1 for SIG. For instance, S1′ holds if and only if $(g, pk_1'/pk_0', A', B_1'/B_0')$ is a Diffie-Hellman tuple; a suitable witness is R'.

S4′, S5′. Similarly, S4′ holds precisely if $(g, pk_0, A/A_Z, B_0/B_Z)$ is a Diffie-Hellman tuple; a witness is $R - R_Z$. (Statement S5′ can be formalized analogously, with a witness R_Z.)

S3′. As with SIG, S3′ holds if and only if there is a $W \in \mathbb{Z}_p$ with $f(H(ovk)) = W^2 \bmod p$. A suitable witness consists of W, and the encryption randomness R_f of C_f.

A reformulation. The composed statement from (28) is equivalent to

$$\big(S1' \vee S2' \vee S3'\big) \quad \wedge \quad \big(S1' \vee S4' \vee S5'\big).$$

By the above, the first sub-statement $S1' \vee S2' \vee S3'$ is implied by the equations

$$\begin{aligned} A^{\widehat{U}} &= g^{\widehat{V}} & A'^{\widehat{U'}} &= g^{\widehat{V'}} & A_f^{\widehat{U_f}} &= g^{\widehat{V_f}} \\ (B_1/B_0)^{\widehat{U}} &= (pk_1/pk_0)^{\widehat{V}} & (B_1'/B_0')^{\widehat{U'}} &= (pk_1'/pk_0')^{\widehat{V'}} & B_0^{\widehat{U_f}} &= pk_0^{\widehat{V_f}} \cdot g^{\widehat{F}} \\ \widehat{F} &= W \cdot \widehat{W} & W &= \widehat{W} & \widehat{U} + \widehat{U'} + \widehat{U_f} &= 1 \end{aligned} \tag{29}$$

for new variables $\widehat{U}, \widehat{V}, \widehat{U'}, \widehat{V'}, \widehat{U_f}, \widehat{V_f}, \widehat{F}, W, \widehat{W}$. (We adopt the notation from Appendix B.1 to decorate variables in $\hat{\mathbb{G}}$ with a hat.) Roughly, the last equation guarantees that one of $\widehat{U}, \widehat{U'}, \widehat{U_f}$ is nonzero, and in fact that $\widehat{U_f} = 1$ once $\widehat{U} = \widehat{U'} = 0$. Furthermore, we have $\widehat{U'} \neq 0 \Rightarrow S1'$, and $\widehat{U} \neq 0 \Rightarrow S2'$, and $\widehat{U_f} \neq 0 \Rightarrow S3'$. Finally, a witness for (29) can be produced from *either* a witness for S1′, or for S2′, or for S3′. (For instance, we can set $\widehat{U'} = \widehat{V'} = 0$ whenever a witness for S1′ is not available.)

Similarly, sub-statement $S1' \vee S4' \vee S5'$ yields additional equations

$$\begin{aligned} (A/A_Z)^{\widehat{U_0}} &= g^{\widehat{V_0}} & A_Z^{\widehat{U_Z}} &= g^{\widehat{V_Z}} & \widehat{U'} + \widehat{U_0} + \widehat{U_Z} &= 1 \\ (B_0/B_Z)^{\widehat{U_0}} &= pk_0^{\widehat{V_0}} & B_Z^{\widehat{U_Z}} &= pk_0^{\widehat{V_Z}} \end{aligned}$$

for new variables $\widehat{U_0}, \widehat{V_0}, \widehat{U_Z}, \widehat{V_Z}$.

Summary. Summing up, π contains commitments to 13 variables (12 of them from $\hat{\mathbb{G}}$), and proves 10 $\mathbb{G}$-linear, 2 $\mathbb{Z}_p$-linear, and 3 quadratic equations over $\mathbb{Z}_p$. This yields a proof of $13 \cdot 2 + 10 \cdot 1 + 3 \cdot 4 = 48$ group elements (30 of them from $\hat{\mathbb{G}}$) and $2 \cdot 1 = 2$ exponents from $\mathbb{Z}_p$.

#	CRS_2	Z	π_2	$Z_0 = Z_1$	**forgery check**	**remark**
1	binding	$X^{(\ell)}$	proof of S3	$X^{(\ell)}$	—	EUF-mCMA
2	binding	$X^{(\ell)}$	proof of S3	$X^{(\ell)}$	$\boxed{\mathrm{Dec}_{\mathrm{eg}}(sk_0^*, C_0) = X^*}$	GS soundness
3	$\boxed{\text{hiding}}$	$X^{(\ell)}$	proof of S3	$X^{(\ell)}$	$\mathrm{Dec}_{\mathrm{eg}}(sk_0^*, C_0) = X^*$	GS CRS indist.
4	hiding	$\boxed{1}$	$\boxed{\text{Sim-output}}$	$X^{(\ell)}$	$\mathrm{Dec}_{\mathrm{eg}}(sk_0^*, C_0) = X^*$	GS simulation
5	hiding	1	Sim-output	$\boxed{\mathcal{R}^{(\ell)}(M)}$	$\mathrm{Dec}_{\mathrm{eg}}(sk_0^*, C_0) = \boxed{\mathcal{R}^{(\ell)}(M^*)}$	see Fig. 2

Figure 1: Outline of the main proof, see Theorem 3.1. $\boxed{\text{Boxes}}$ denote changes compared to the previous game. The first column denotes the game number, CRS_2 denotes the setup of the Groth-Sahai common reference string CRS_2, and Z denotes the value committed to in C_Z in verification keys. Column π_2 describes how proofs are prepared in signatures. Z_0, Z are the messages encrypted in C_0, C_1 in signatures generated for $\mathcal{A}$. **forgery check** describes an additional check required for a forgery to pass as valid (beyond being valid in the sense of Ver). The core of the proof is the transition from Game 4 to Game 5 (with the previous transitions preparing the ground), see also Fig. 2.

#	CRS_1	f	if $\left(\frac{f^*(M)}{p}\right) = 1$: π_1	Z_0	Z_1	if $\left(\frac{f^*(M)}{p}\right) = -1$: π_1	$Z_0 = Z_1$	**forgery check**	**abort condition**	**remark**
4.i.0	hiding	0	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathcal{H}_i^{(\ell)}(M)$	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathrm{Dec}_{eg}(sk_0^*, C_0^*) = \mathcal{H}_i^{(\ell)}(M^*)$	—	same as 4.i
4.i.1	hiding	0	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathcal{H}_i^{(\ell)}(M)$	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathrm{Dec}_{eg}(sk_0^*, C_0^*) = \mathcal{H}_i^{(\ell)}(M^*)$	$\boxed{f^*(M^*) \in \mathrm{QR}_p \cup \{0\}}$	loses factor ≈ 2
4.i.2	hiding	$\boxed{f^*}$	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathcal{H}_i^{(\ell)}(M)$	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathrm{Dec}_{eg}(sk_0^*, C_0^*) = \mathcal{H}_i^{(\ell)}(M^*)$	$f^*(M^*) \in \mathrm{QR}_p \cup \{0\}$	GS witness-ind.
4.i.3	hiding	f^*	$\boxed{S2}$	$\mathcal{H}_i^{(\ell)}(M)$	$\boxed{\mathcal{Q}_i^{(\ell)}(M)}$	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathrm{Dec}_{eg}(sk_0^*, C_0^*) = \mathcal{H}_i^{(\ell)}(M^*)$	$f^*(M^*) \in \mathrm{QR}_p \cup \{0\}$	ElGamal
4.i.4	$\boxed{\text{binding}}$	f^*	S2	$\mathcal{H}_i^{(\ell)}(M)$	$\mathcal{Q}_i^{(\ell)}(M)$	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathrm{Dec}_{eg}(sk_0^*, C_0^*) = \mathcal{H}_i^{(\ell)}(M^*)$	$f^*(M^*) \in \mathrm{QR}_p \cup \{0\}$	GS CRS indist.
4.i.5	binding	f^*	S2	$\mathcal{H}_i^{(\ell)}(M)$	$\mathcal{Q}_i^{(\ell)}(M)$	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathrm{Dec}_{eg}(\boxed{sk_1^*, C_1^*}) = \mathcal{H}_i^{(\ell)}(M^*)$	$f^*(M^*) \in \mathrm{QR}_p \cup \{0\}$	GS soundness
4.i.6	binding	f^*	S2	$\boxed{\mathcal{Q}_i^{(\ell)}(M)}$	$\mathcal{Q}_i^{(\ell)}(M)$	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathrm{Dec}_{eg}(sk_1^*, C_1^*) = \mathcal{H}_i^{(\ell)}(M^*)$	$f^*(M^*) \in \mathrm{QR}_p \cup \{0\}$	ElGamal
4.i.7	binding	f^*	S2	$\mathcal{Q}_i^{(\ell)}(M)$	$\mathcal{Q}_i^{(\ell)}(M)$	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathrm{Dec}_{eg}(\boxed{sk_0^*, C_0^*}) = \mathcal{H}_i^{(\ell)}(M^*)$	$f^*(M^*) \in \mathrm{QR}_p \cup \{0\}$	GS soundness
4.i.8	$\boxed{\text{hiding}}$	f^*	S2	$\mathcal{Q}_i^{(\ell)}(M)$	$\mathcal{Q}_i^{(\ell)}(M)$	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathrm{Dec}_{eg}(sk_0^*, C_0^*) = \mathcal{H}_i^{(\ell)}(M^*)$	$f^*(M^*) \in \mathrm{QR}_p \cup \{0\}$	GS CRS indist.
4.i.9	hiding	$\boxed{0}$	$\boxed{S1}$	$\mathcal{Q}_i^{(\ell)}(M)$	$\mathcal{Q}_i^{(\ell)}(M)$	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathrm{Dec}_{eg}(sk_0^*, C_0^*) = \mathcal{H}_i^{(\ell)}(M^*)$	$f^*(M^*) \in \mathrm{QR}_p \cup \{0\}$	GS witness-ind.
4.i.10	hiding	0	S1	$\mathcal{Q}_i^{(\ell)}(M)$	$\mathcal{Q}_i^{(\ell)}(M)$	S1	$\mathcal{H}_i^{(\ell)}(M)$	$\mathrm{Dec}_{eg}(sk_0^*, C_0^*) = \mathcal{H}_i^{(\ell)}(M^*)$	$\boxed{—}$	gains factor ≈ 2 same as 4.$(i+1)$

Figure 2: Transitions between two hybrids Game 4.i and Game 4.$(i+1)$ that in turn interpolate between Game 4 and Game 5 of the main proof. Again, $\boxed{\text{boxes}}$ denote changes compared to the previous game. The notation follows Fig. 1: # denotes the game number, and CRS_1 and f denote the setup of these values in the public parameters. The column π_1 describes which sub-statement (i.e., S1 or S2) the proof π_1 actually proves, and the columns Z_0, Z_1 describe how the game prepares signatures for $\mathcal{A}$. In this, we distinguish the cases where the Legendre symbol $\left(\frac{f^*(M)}{p}\right)$ of the message to be signed is 1 and -1, respectively. (We neglect the unlikely case $\left(\frac{f^*(M)}{p}\right) = 0$ in this overview.) Also, as in Fig. 1, **forgery check** describes an additional check required for a forgery to pass as valid. Finally, the functions $\mathcal{H}_i^{(\ell)}$ are defined at the beginning of the proof of Lemma 3.2. (Intuitively, $\mathcal{H}_i^{(\ell)}$ is a random function that however does not depend on its full input M, but only on i values $\left(\frac{f_j(M)}{p}\right)$ for randomly chosen f_j.) We refer to Lemma 3.2 for a detailed proof and a justification for each game transition.