Short Group Signatures via Structure-Preserving Signatures: Standard Model Security from Simple Assumptions*

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

[1] Ecole Normale Supérieure de Lyon (France)
[2] Ecole Normale Supérieure (France)
[3] Google Inc. and Columbia University (USA)

Abstract. Group signatures are a central cryptographic primitive which allows users to sign messages while hiding their identity within a crowd of group members. In the standard model (without the random oracle idealization), the most efficient constructions rely on the Groth-Sahai proof systems (Eurocrypt'08). The structure-preserving signatures of Abe *et al.* (Asiacrypt'12) make it possible to design group signatures based on well-established, constant-size number theoretic assumptions (a.k.a. "simple assumptions") like the Symmetric eXternal Diffie-Hellman or Decision Linear assumptions. While much more efficient than group signatures built on general assumptions, these constructions incur a significant overhead w.r.t. constructions secure in the idealized random oracle model. Indeed, the best known solution based on simple assumptions requires 2.8 kB per signature for currently recommended parameters. Reducing this size and presenting techniques for shorter signatures are thus natural questions. In this paper, our first contribution is to significantly reduce this overhead. Namely, we obtain the first fully anonymous group signatures based on simple assumptions with signatures shorter than 2 kB at the 128-bit security level. In dynamic (resp. static) groups, our signature length drops to 1.8 kB (resp. 1 kB). This improvement is enabled by two technical tools. As a result of independent interest, we first construct a new structure-preserving signature based on simple assumptions which shortens the best previous scheme by 25%. Our second tool is a new method for attaining anonymity in the strongest sense using a new CCA2-secure encryption scheme which is simultaneously a Groth-Sahai commitment.

Keywords. Group signatures, standard model, simple assumptions, efficiency, structure-preserving cryptography, QA-NIZK arguments.

1 Introduction

As introduced by Chaum and van Heyst [29] in 1991, group signatures allow members of a group administered by some authority to anonymously sign messages on behalf of the group. In order to prevent abuses, an opening authority has the power to uncover a signer's identity if the need arises.

The usual approach for building a group signature consists in having the signer encrypt his group membership credential under the public key of the opening authority while appending a non-interactive zero-knowledge (NIZK) proof, which is associated with the message, claiming that things were done correctly. Until 2006, efficient instantiations of this primitive were only available under the random oracle idealization [14], which is limited to only provide heuristic arguments in terms of security [24]. This state of affairs changed in the last decade, with the emergence of solutions [20,21,37,38] enabled by breakthrough results in the design of relatively efficient non-interactive witness indistinguishable (NIWI) proofs [39]. While drastically more efficient than solutions based on general NIZK proofs [12,15], the constructions of [20,21,37,38] still incur a substantial overhead when compared with their random-oracle-based counterparts [10,32,18]. Moreover, their most efficient variants [21,38] tend to rely on parametrized assumptions – often referred to as "q-type" assumptions – where the number of input elements is determined by a parameter q which, in turn, depends on the number of users in the system or the number of adversarial queries (or both). Since the assumption becomes stronger as q increases, a different assumption is needed for every adversary (based on its number of queries) and every maximal number of users in the group. Not only does it limit the scalability of realizations, it also restricts the level of confidence in their security.

* This is the full version of a paper published at Crypto 2015.

In this paper, we consider the problem of devising as short as possible group signatures based on simple assumptions. By "simple assumption", we mean a well-established assumption, like the Decision Diffie-Hellman assumption, which is simultaneously non-interactive (and thus falsifiable [56]) and described using a constant number of elements, regardless of the number of users in the system or the number of adversarial queries. We remark that even in the random oracle model, this problem turns out to be highly non-trivial as non-simple assumptions (like the Strong RSA [10,46] or Strong Diffie-Hellman [18,32]) are frequently relied on. In the standard model, our main contribution is designing the first group signatures based on simple assumptions and whose size is less than 2 kB for the currently recommended 128-bit security level. In static groups, our most efficient scheme features signatures slightly longer than 1 kB. So far, the best standard-model group signature based on simple assumptions was obtained from the structure-preserving signatures (SPS) of Abe *et al* [1,2] and required 2.875 kB per signature. Along the way and as a result of independent interest, we also build a new structure-preserving signature (SPS) with the shortest length among those based on simple assumptions. Concretely, the best previous SPS based on similar assumptions [1,2] is shortened by 25%.

Related Work. Group signatures have a long history. Still, efficient and provably coalition-resistant constructions (in the random oracle model) remained elusive until the work of Ateniese, Camenisch, Joye and Tsudik [10] in 2000. At that time, however, there was no proper formalization of the security properties that can be naturally expected from group signatures. This gap was filled in 2003 by Bellare, Micciancio and Warinschi [12] (BMW) who captured all the requirements of group signatures in three properties. In (a variant of) this model, Boneh, Boyen and Shacham [18] obtained very short signatures using the random oracle methodology [14].

The BMW model assumes static groups where the set of members is frozen after the setup phase beyond which no new member can be added. The setting of dynamic groups was explored later on by Bellare-Shi-Zhang [15] and, independently, by Kiayias and Yung [46]. In these models [15,46], short signature lengths were obtained in [57,32]. A construction based on interactive assumptions in the standard model was also put forth by Ateniese *et al.* [9]. Using standard assumptions, Boyen and Waters gave a different solution [20] based on the Groth-Ostrovsky-Sahai NIZK proof system [36]. They subsequently managed to obtain $O(1)$-size signatures at the expense of appealing to a q-type assumption [21] . Their constructions [20,21] were both analyzed in (a relaxation of) the BMW model [12] where the adversary is not granted access to a signature opening oracle. In dynamic groups [15], Groth [37] obtained constant-size signatures in the standard model but, due to huge hidden constants, his result was mostly a proof of concept. By making the most of Groth-Sahai NIWI proofs [39], he subsequently reduced signatures to 48 group elements [38] with the caveat of resting on relatively *ad hoc* q-type assumptions. For the time being, the best group signatures based on standard assumptions are enabled by the structure-preserving signatures of Abe, Chase, David, Kohlweiss, Nishimaki, and Ohkubo [1]. In asymmetric pairings $e : \mathbb{G} \times \hat{\mathbb{G}} \rightarrow \mathbb{G}_T$ (where $\mathbb{G} \neq \hat{\mathbb{G}}$), anonymously signing messages requires at least 40 elements of $\mathbb{G}$ and 26 elements of $\hat{\mathbb{G}}$.

In 2010, Abe *et al.* [8,3] advocated the use of *structure-preserving* cryptography as a general tool for building privacy-preserving protocols in a modular fashion. In short, structure-preserving signatures (SPS) are signature schemes that smoothly interact with Groth-Sahai proofs [39] as messages, signatures public keys all live in the source groups $(\mathbb{G}, \hat{\mathbb{G}})$ of a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \rightarrow \mathbb{G}_T$. SPS schemes were initially introduced by Groth [37] and further studied in [26,33]. In the last three years, a large body of work was devoted to the feasibility and efficiency of structure-preserving signatures [37,26,33,8,3,4,23,28,40,1,2]. In Type III pairings (i.e., where $\mathbb{G} \neq \hat{\mathbb{G}}$ and no isomorphism is computable from $\hat{\mathbb{G}}$ to $\mathbb{G}$ or backwards), Abe *et al.* [4] showed that any SPS scheme must contain at least 3 group elements per signature. For a natural class of reductions, the security of optimally short signatures was also shown [5] *unprovable* under any non-interactive assumption. These impossibility results were recently found [7] not to carry over to Type II pairings (i.e., where $\mathbb{G} \neq \hat{\mathbb{G}}$ and an

efficiently computable isomorphism $\psi : \hat{\mathbb{G}} \to \mathbb{G}$ is available).

To the best of our knowledge, the minimal length of structure-preserving signatures based on simple assumptions remains an unsettled open question. We believe it to be of primary importance considering the versatility of structure-preserving cryptography in the design of privacy-related protocols, including group signatures [8], group encryption [26] or adaptive oblivious transfer [35].

Our Results. The first contribution of this paper is to describe a new structure-preserving signature based on the standard Symmetric eXternal Diffie-Hellman (SXDH) assumption and an asymmetric variant of the Decision Linear assumption with only 10 group elements (more precisely, 9 elements of $\mathbb{G}$ and one element of $\hat{\mathbb{G}}$) per signature. So far, the best instantiation of [1,2] required 7 elements of $\mathbb{G}$ and 4 elements of $\hat{\mathbb{G}}$. Since the representation of $\hat{\mathbb{G}}$ elements is at least twice as long as that of $\mathbb{G}$ elements, our scheme thus saves 26% in terms of signature length. Armed with our new SPS and other tools, we then construct dynamic group signatures using only 32 elements of $\mathbb{G}$ and 14 elements of $\hat{\mathbb{G}}$ in each signature, where Abe *et al.* [1,2] need at least 40 elements of $\mathbb{G}$ and 26 elements of $\hat{\mathbb{G}}$. For typical parameters, our signatures are thus 37% shorter with a total length of only 1.8 kB at the 128-bit security level. In an independent work, Kiltz, Pan and Wee [49] managed to obtain even shorter structure-preserving signatures than ours under the SXDH assumption. If their construction is used in our dynamic group signature, it allows eliminating at least 4 more elements of $\mathbb{G}$ from the group signatures.

In the static model of Bellare, Micciancio and Warinschi [12], we also describe an even more efficient realization where the signature length decreases to almost 1 kB.

Our Techniques. Our structure-preserving signature can be seen as a non-trivial optimization of a modular design, suggested by Abe *et al* [1], which combines a weakly secure SPS scheme and a tagged one-time signature (TOTS). In a TOTS scheme, each signature contains a fresh tag and, without knowing the private key, it should be computationally infeasible to generate a signature on a new message for a previously used tag. The construction of [1] obtains a full-fledged SPS by combining a TOTS scheme with an SPS system that is only secure against extended random message attacks (XRMA). As defined in [1], XRMA security basically captures security against an adversary that only obtains signatures on random group elements *even knowing* some auxiliary information used to sample these elements (typically their discrete logarithms). While Abe *et al.* [1] make use of the discrete logs of signed messages in their proofs of XRMA security, their modular construction does not. Here, by explicitly using the discrete logarithms in the construction, we obtain significant efficiency improvements. Using Waters' dual system techniques [62], we construct an SXDH-based F-unforgeable signature scheme which, according to the terminology of Belenkiy *et al.* [11], is a signature scheme that remains verifiable and unforgeable even if the adversary only outputs an injective function of the forgery message. Our new SPS is the result of combining our F-unforgeable signature and the TOTS system of [2]. We stress that our scheme can no longer be seen as an instantiation of a generic construction. Still, at the natural expense of sacrificing modularity, it does provide shorter signatures.

In turn, our F-unforgeable signatures are obtained by taking advantage of the quasi-adaptive NIZK (QA-NIZK) arguments of linear subspace membership suggested by Jutla and Roy [43] and further studied in [53,44], where the CRS may depend on the language for which proofs have to be generated. In a nutshell, our starting point is a signature scheme suggested by Jutla and Roy (inspired by ideas due to Camenisch *et al.* [22]) where each signature is a CCA2-secure encryption of the private key (made verifiable via QA-NIZK proofs) and the message is included in the label [60]. We rely on the observation that QA-NIZK proofs for linear subspaces [43] (or their optimized variants [53,44]) make it possible to verify signatures even if the message is only available in the exponent.

In order to save the equivalent of 15 elements of the group $\mathbb{G}$ and make the group signature as short as possible, we also design a new CCA2-secure tag-based encryption (TBE) scheme [55,48] which

incorporates a Groth-Sahai commitment. In fully anonymous group signatures, CCA2-anonymity is usually acquired by verifiably encrypting the signer's credential using a CCA2-secure cryptosystem while providing evidence that the plaintext coincides with a committed group element. Inspired by a lossy encryption scheme [13] suggested by Hemenway *et al.* [41], we depart from this approach and rather use a CCA2-secure encryption scheme which simultaneously plays the role of a Groth-Sahai commitment. That is, even when the Groth-Sahai CRS is a perfectly hiding CRS, we are able to extract committed group elements for any tag but a specific one, where the encryption scheme behaves like a perfectly hiding commitment and induces perfectly NIWI proofs. In order to make the validity of TBE ciphertexts publicly verifiable, we rely on the QA-NIZK proofs of Libert *et al.* [53] which are well-suited to the specific subspaces encountered[4] in this context. We believe this encryption scheme to be of interest in its own right since it allows shortening other group signatures based on Groth-Sahai proofs (e.g., [38]) in a similar way.

Our group signature in the static BMW model [12] does not build on structure-preserving signatures but rather follows the same design principle as the constructions of Boyen and Waters [20,21]. It is obtained by extending our F-unforgeable signature into a 2-level hierarchical signature [47] (or, equivalently, an identity-based signature [59]) where first-level messages are implicit in the exponent. In spirit and from an efficiency standpoint, our static group signature is thus similar to the second construction [21] of Boyen and Waters, with the benefit of providing full anonymity while relying on the sole SXDH assumption.

2 Background

2.1 Hardness Assumptions

We use bilinear maps $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ over groups of prime order p where $e(g, \hat{h}) \neq 1_{\mathbb{G}_T}$ if and only if $g \neq 1_{\mathbb{G}}$ and $\hat{h} \neq 1_{\hat{\mathbb{G}}}$. We rely on hardness assumptions that are non-interactive and described using a constant number of elements.

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

In the following, we will rely on the Symmetric external Diffie-Hellman (SXDH) assumption which posits the hardness of DDH in $\mathbb{G}$ and $\hat{\mathbb{G}}$ in asymmetric pairing configurations. We also assume the hardness of the following problem, which generalizes the Decision Linear problem [18] to asymmetric pairings.

Definition 2 ([1]). *In bilinear groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime order p, the* **eXternal Decision Linear Problem** 2 *(XDLIN$_2$) is to distinguish the distribution*

$$D_1 = \{(g, g^a, g^b, g^{ac}, g^{bd}, \hat{g}, \hat{g}^a, \hat{g}^b, \hat{g}^{ac}, \hat{g}^{bd}, \hat{g}^{c+d}) \in \mathbb{G}^5 \times \hat{\mathbb{G}}^6 \mid a, b, c, d \xleftarrow{R} \mathbb{Z}_p\}$$
$$D_2 = \{(g, g^a, g^b, g^{ac}, g^{bd}, \hat{g}, \hat{g}^a, \hat{g}^b, \hat{g}^{ac}, \hat{g}^{bd}, \hat{g}^{z}) \in \mathbb{G}^5 \times \hat{\mathbb{G}}^6 \mid a, b, c, d, z \xleftarrow{R} \mathbb{Z}_p\}.$$

The XDLIN$_1$ assumption is defined analogously and posits the infeasibility of distinguishing g^{c+d} and g^z given $(g, g^a, g^b, g^{ac}, g^{bd}, \hat{g}, \hat{g}^a, \hat{g}^b, \hat{g}^{ac}, \hat{g}^{bd})$.

[4] Specifically, we have to prove membership of a $t \times n$ subspace of rank t described by a $2t \times n$ matrix and the security proofs of [52,53] still work in this case.

2.2 Linearly Homomorphic Structure-Preserving Signatures

Structure-preserving signatures [8,3] are signature schemes where messages and public keys all consist of elements of a group over which a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ is efficiently computable.

Libert *et al.* [52] considered structure-preserving signatures with linear homomorphic properties. This section recalls the one-time linearly homomorphic structure-preserving signature (LHSPS) of [52]. In the description below, we assume that all algorithms take as input the description of common public parameters cp consisting of asymmetric bilinear groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, p)$ of prime order $p > 2^\lambda$, where λ is the security parameter.

In [52], Libert *et al.* suggested the following construction which can be proved secure under the SXDH assumption.

Keygen(cp, n): Given common public parameters $\mathsf{cp} = (\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, p)$ and the dimension $n \in \mathbb{N}$ of the subspace to be signed. Then, choose $\hat{g}_z, \hat{g}_r \xleftarrow{R} \hat{\mathbb{G}}$. For $i = 1$ to n, pick $\chi_i, \gamma_i \xleftarrow{R} \mathbb{Z}_p$ and compute $\hat{g}_i = \hat{g}_z^{\chi_i} \hat{g}_r^{\gamma_i}$. The private key is $\mathsf{sk} = \{(\chi_i, \gamma_i)\}_{i=1}^n$ while the public key is

$$\mathsf{pk} = \big(\hat{g}_z,\ \hat{g}_r,\ \{\hat{g}_i\}_{i=1}^n\big) \in \hat{\mathbb{G}}^{n+2}.$$

Sign($\mathsf{sk}, (M_1, \ldots, M_n)$): In order to sign a vector $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $\mathsf{sk} = \{(\chi_i, \gamma_i)\}_{i=1}^n$, output $\sigma = (z, r) = \big(\prod_{i=1}^n M_i^{-\chi_i}, \prod_{i=1}^n, M_i^{-\gamma_i}\big)$.

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

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

$$1_{\mathbb{G}_T} = e(z, \hat{g}_z) \cdot e(r, \hat{g}_r) \cdot \prod_{i=1}^n e(M_i, \hat{g}_i).$$

In [53], (a variant of) this scheme was used to construct constant-size QA-NIZK arguments [43] showing that a vector $\boldsymbol{v} \in \mathbb{G}^n$ belongs to a linear subspace subspace of rank t spanned by a matrix $\boldsymbol{\rho} \in \mathbb{G}^{t \times n}$. Under the SXDH assumption, each argument is comprised of two elements of $\mathbb{G}$, independently of t or n.

3 An F-Unforgeable Signature

As a technical tool, our constructions rely on a signature scheme which we prove F-unforgeable under the SXDH assumption. As defined by Belenkiy *et al.* [11], F-unforgeability refers to the inability of the adversary to output a valid signature for a non-trivial message M without outputting the message itself. Instead, the adversary is only required to output $F(M)$, for an injective but not necessarily efficiently invertible function F.

The scheme extends ideas used in signature schemes suggested in [22,43,54], where each signature is a CCA2-secure encryption —using the message to be signed as a label— of the private key accompanied with a QA-NIZK proof that the encrypted value is the private key. In their most efficient variant, Jutla and Roy observed [43, Section 5] that it suffices to encrypt private keys g^ω with a projective hash value $(v^M \cdot w)^r$ [31] so as to obtain signatures of the form $(\sigma_1, \sigma_3, \sigma_3) = (g^\omega \cdot (v^M \cdot w)^r, g^r, h^r)$, which is reminiscent of selectively secure Boneh-Boyen signatures [16].

As in [62,51,34], the security proof proceeds with a sequence of hybrid games to gradually reach a game where the signing oracle never uses the private key, in which case it becomes much easier to prove security. In the final game, signatures always encrypt a random value while QA-NIZK proofs are simulated. When transitioning from one hybrid game to the next one, the crucial step is to argue that, even if the signing oracle produces fewer and fewer signatures using the real private key, the

adversary's forgery will still encrypt the private key. This is achieved via an information theoretic argument borrowed from 2-universal hash proof systems [30,31].

In order to obtain an F-unforgeable signature which is verifiable given only $F(M)$, our key observation is that QA-NIZK proofs make it possible to verify signatures even if M appears only implicitly in a tuple $(g^{s \cdot M}, g^s, h^{s \cdot M}, h^s) \in \mathbb{G}^4$.

Keygen(cp) : Given common public parameters $\mathsf{cp} = (\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, p)$ consisting of asymmetric bilinear groups of prime order $p > 2^\lambda$, do the following.

1. Choose $\omega, a \xleftarrow{R} \mathbb{Z}_p$, $g, v, w \xleftarrow{R} \mathbb{G}$, $\hat{g} \xleftarrow{R} \hat{\mathbb{G}}$ and set $h = g^a$, $\Omega = h^\omega$.
2. Define a matrix $\mathbf{M} = (M_{j,i})_{j,i}$ given by

$$\mathbf{M} = \left(\begin{array}{c|c|c|c|c|c} g & 1 & 1 & 1 & 1 & h \\ \hline v & g & 1 & h & 1 & 1 \\ \hline w & 1 & g & 1 & h & 1 \end{array} \right) \in \mathbb{G}^{3 \times 6}. \tag{1}$$

3. Generate a key pair $(\mathsf{sk}_{hsps}, \mathsf{pk}_{hsps})$ for the one-time linearly homomorphic signature of Section 2.2 in order to sign vectors of dimension $n = 6$. Let $\mathsf{sk}_{hsps} = \{(\chi_i, \gamma_i)\}_{i=1}^6$ be the private key, of which the corresponding public key is $\mathsf{pk}_{hsps} = \big(\hat{g}_z,\ \hat{g}_r,\ \{\hat{g}_i\}_{i=1}^6\big)$.
4. Using $\mathsf{sk}_{hsps} = \{\chi_i, \gamma_i\}_{i=1}^6$, generate one-time homomorphic signatures $\{(z_j, r_j)\}_{j=1}^3$ on the rows $\boldsymbol{M}_j = (M_{j,1}, \ldots, M_{j,6}) \in \mathbb{G}^6$ of $\mathbf{M}$. These are obtained as

$$(z_j, r_j) = \left(\prod_{i=1}^6 M_{j,i}^{-\chi_i},\ \prod_{i=1}^6 M_{j,i}^{-\gamma_i} \right), \qquad \forall j \in \{1,2,3\}$$

and, as part of the common reference string for the QA-NIZK proof system of [53], they will be included in the public key.

The private key is $\mathsf{sk} := \omega$ and the public key is defined as

$$\mathsf{pk} = \Big((\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T),\ p,\ g,\ h,\ \hat{g},\ (v, w),\ \Omega = h^\omega,\ \mathsf{pk}_{hsps},\ \{(z_j, r_j)\}_{j=1}^3 \Big).$$

Sign(sk, M) : given the private key $\mathsf{sk} = \omega$ and a message $M \in \mathbb{Z}_p$, choose $s \xleftarrow{R} \mathbb{Z}_p$ to compute

$$\sigma_1 = g^\omega \cdot (v^M \cdot w)^s, \qquad \sigma_2 = g^{s \cdot M}, \qquad \sigma_3 = g^s$$
$$\sigma_4 = h^{s \cdot M} \qquad \sigma_5 = h^s$$

Then, generate a QA-NIZK proof that the vector $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \Omega) \in \mathbb{G}^6$ is in the row space of $\mathbf{M}$. This QA-NIZK proof $(z, r) \in \mathbb{G}^2$ is obtained as

$$z = z_1^\omega \cdot (z_2^M \cdot z_3)^s, \qquad r = r_1^\omega \cdot (r_2^M \cdot r_3)^s. \tag{2}$$

Return the signature $\sigma = \big(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, z, r\big)$.

Verify(pk, σ, M) : parse σ as above and return 1 if and only if it holds that

$$e(z, \hat{g}_z) \cdot e(r, \hat{g}_r) = e(\sigma_1, \hat{g}_1)^{-1} \cdot e(\sigma_3, \hat{g}_3 \cdot \hat{g}_2{}^M)^{-1} \cdot e(\sigma_5, \hat{g}_5 \cdot \hat{g}_4{}^M)^{-1} \cdot e(\Omega, \hat{g}_6)^{-1}$$

and $(\sigma_2, \sigma_4) = (\sigma_3^M, \sigma_5^M)$.

Note that a signature can be verified given only $F(M) = \hat{g}^M$ by testing the equalities

$$e(\sigma_2, \hat{g}) = e(\sigma_3, F(M)), \qquad e(\sigma_4, \hat{g}) = e(\sigma_5, F(M))$$

and

$$e(z,\hat{g}_z)\cdot e(r,\hat{g}_r) = e(\sigma_1,\hat{g}_1)^{-1}\cdot e(\sigma_2,\hat{g}_2)^{-1}\cdot e(\sigma_3,\hat{g}_3)^{-1}\cdot e(\sigma_4,\hat{g}_4)^{-1}\cdot e(\sigma_5,\hat{g}_5)^{-1}\cdot e(\Omega,\hat{g}_6)^{-1}$$

In order to keep the description as simple as possible, the above description uses the QA-NIZK argument system of [53], which is based on linearly homomorphic signatures. However, the security proof goes through if we use the more efficient SXDH-based QA-NIZK argument of Jutla and Roy [44], as explained in Appendix D. The pair (z,r) can thus be replaced by a single element of $\mathbb{G}$.

Under the SXDH assumption, the scheme can be proved to be F-unforgeable for the injective function $F(M) = \hat{g}^M$. The proof of this result is implied by the security result of Section 4 where we describe a generalization of the scheme that will be used to build a group signature in the BMW model.

4 A Two-Level Hierarchical Signature from the SXDH Assumption

This section extends our F-unforgeable signature into a 2-level hierarchical signature with partially hidden messages. In a 2-level hierarchical signature [47] (a.k.a. identity-based signature), a signature on a message ID (called "identity") can be used as a delegated key for signing messages of the form (ID, M) for any M. In order to construct group signatures, Boyen and Waters [21] used hierarchical signatures that can be verified even when identities (i.e., first-level messages) are not explicitly given to the verifier, but only appear implicitly in the exponent. The syntax and security definition are recalled in Appendix C.3.

In their most efficient construction [21], Boyen and Waters used a non-standard q-type assumption. This section gives a very efficient solution based on the standard SXDH assumption. It is obtained from our signature of Section 3 by having a signature $(g^\omega\cdot(v^{\mathsf{ID}}\cdot w)^s, g^s, h^s)$ on a given identity ID serve as a private key for this identity modulo the introduction of a delegation component t^s akin to those of the Boneh-Boyen-Goh hierarchical IBE [17]. For the security proof to go through, we need to make sure that pairs $(g^{s\cdot M}, g^s)$, $(h^{s\cdot M}, h^s)$ hide the same message M, which is not immediately verifiable in the SXDH setting. To enforce this condition, we thus include $\hat{g}^M$ in each signature.

Setup(cp) : Given public parameters $\mathsf{cp} = (\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, p)$, do the following.

1. Choose $\omega, a \xleftarrow{R} \mathbb{Z}_p$, $g, t, v, w \xleftarrow{R} \mathbb{G}$, $\hat{g} \xleftarrow{R} \hat{\mathbb{G}}$ and set $h = g^a$, $\Omega = h^\omega$.
2. Define a matrix $\mathbf{M} = (M_{j,i})_{j,i}$ given by

$$\mathbf{M} = \left(\begin{array}{c|c|c|c|c|c|c|c} g & 1 & 1 & 1 & 1 & 1 & 1 & h \\ \hline v & g & 1 & h & 1 & 1 & 1 & 1 \\ \hline w & 1 & g & 1 & h & 1 & 1 & 1 \\ \hline t & 1 & 1 & 1 & 1 & g & h & 1 \end{array}\right) \in \mathbb{G}^{4\times 8}. \tag{3}$$

3. Generate a key pair $(\mathsf{sk}_{hsps}, \mathsf{pk}_{hsps})$ for the one-time linearly homomorphic signature of Section 2.2 in order to sign vectors of dimension $n = 8$. Let $\mathsf{sk}_{hsps} = \{(\chi_i, \gamma_i)\}_{i=1}^8$ be the private key, of which the corresponding public key is $\mathsf{pk}_{hsps} = \big(\hat{g}_z,\ \hat{g}_r,\ \{\hat{g}_i\}_{i=1}^8\big)$.
4. Using $\mathsf{sk}_{hsps} = \{\chi_i, \gamma_i\}_{i=1}^8$, generate one-time homomorphic signatures $\{(z_j, r_j)\}_{j=1}^4$ on the rows $\boldsymbol{M}_j = (M_{j,1}, \ldots, M_{j,8}) \in \mathbb{G}^8$ of $\mathbf{M}$. These are obtained as

$$(z_j, r_j) = \left(\prod_{i=1}^8 M_{j,i}^{-\chi_i},\ \prod_{i=1}^8 M_{j,i}^{-\gamma_i}\right), \qquad \forall j \in \{1,\ldots,4\}$$

and, as part of the common reference string for the QA-NIZK proof system of [53], they will be included in the public key.

The master secret key is $\mathsf{msk} := \omega$ and the master public key is defined as

$$\mathsf{mpk} = \Big((\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T),\ p,\ g,\ h,\ \hat{g},\ (t, v, w),\ \Omega = h^{\omega},\ \mathsf{pk}_{hsps},\ \{(z_j, r_j)\}_{j=1}^{4}\Big).$$

Extract$(\mathsf{msk}, \mathsf{ID})$: given the master secret key $\mathsf{msk} = \omega$ and an identity $\mathsf{ID} \in \mathbb{Z}_p$, choose $s \xleftarrow{R} \mathbb{Z}_p$ to compute

$$\begin{array}{lll} K_1 = g^{\omega} \cdot (v^{\mathsf{ID}} \cdot w)^{s}, & K_2 = g^{s \cdot \mathsf{ID}}, & K_3 = g^{s} \\ K_4 = h^{s \cdot \mathsf{ID}} & K_5 = h^{s} & K_6 = t^{s} \end{array}$$

as well as $\hat{K}_7 = \hat{g}^{\mathsf{ID}}$. Looking ahead, K_6 will serve as a delegation component in the generation of level 2 signatures. Then, generate a QA-NIZK proof that $(K_1, K_2, K_3, K_4, K_5, 1, 1, \Omega) \in \mathbb{G}^8$ is in the row space of the first 3 rows of $\mathbf{M}$. This QA-NIZK proof $(z, r) \in \mathbb{G}^2$ is obtained as

$$z = z_1^{\omega} \cdot (z_2^{\mathsf{ID}} \cdot z_3)^{s}, \qquad r = r_1^{\omega} \cdot (r_2^{\mathsf{ID}} \cdot r_3)^{s}. \tag{4}$$

Then, generate a QA-NIZK proof (z_d, r_d) that the delegation component K_6 is well-formed. This proof consists of $(z_d, r_d) = (z_4^{s}, r_4^{s})$. The private key is

$$K_{\mathsf{ID}} = \big(K_1, K_2, K_3, K_4, K_5, K_6, \hat{K}_7, z, r, z_d, r_d\big). \tag{5}$$

Sign$(\mathsf{mpk}, K_{\mathsf{ID}}, M)$: to sign $M \in \mathbb{Z}_p$, parse K_{ID} as in (5) and do the following.

1. Choose $s' \xleftarrow{R} \mathbb{Z}_p$ and compute

$$\sigma_1 = K_1 \cdot K_6^{M} \cdot (v^{\mathsf{ID}} \cdot t^{M} \cdot w)^{s'} = g^{\omega} \cdot (v^{\mathsf{ID}} \cdot t^{M} \cdot w)^{\tilde{s}},$$

where $\tilde{s} = s + s'$, as well as

$$\begin{array}{lll} \sigma_2 = K_2 \cdot g^{s' \cdot \mathsf{ID}} = g^{\tilde{s} \cdot \mathsf{ID}}, & \sigma_3 = K_3 \cdot g^{s'} = g^{\tilde{s}}, & \hat{\sigma_6} = \hat{K}_7 = \hat{g}^{\mathsf{ID}} \\ \sigma_4 = K_4 \cdot h^{s' \cdot \mathsf{ID}} = h^{\tilde{s} \cdot \mathsf{ID}}, & \sigma_5 = K_5 \cdot h^{s'} = h^{\tilde{s}}. & \end{array}$$

2. Using (z, r) and (z_d, r_d), generate a QA-NIZK proof $(\tilde{z}, \tilde{r}) \in \mathbb{G}^2$ that

$$(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_3^{M}, \sigma_5^{M}, \Omega) \in \mathbb{G}^8 \tag{6}$$

is in the row space of $\mathbf{M}$. Namely, compute

$$\tilde{z} = z \cdot z_d^{M} \cdot (z_2^{\mathsf{ID}} \cdot z_4^{M} \cdot z_3)^{s'}, \qquad \tilde{r} = r \cdot r_d^{M} \cdot (r_2^{\mathsf{ID}} \cdot r_4^{M} \cdot r_3)^{s'}. \tag{7}$$

Return the signature

$$\sigma = \big(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \tilde{z}, \tilde{r}, \hat{\sigma_6}\big) \in \mathbb{G}^7 \times \hat{\mathbb{G}}. \tag{8}$$

Verify$(\mathsf{mpk}, \sigma, M)$: parse σ as per (8) and return 1 if and only if it holds that

$$\begin{aligned} e(\tilde{z}, \hat{g}_z) \cdot e(\tilde{r}, \hat{g}_r) = e(\sigma_1, \hat{g}_1)^{-1} \cdot e(\sigma_2, \hat{g}_2)^{-1} \cdot e(\sigma_3, \hat{g}_3 \cdot \hat{g}_6{}^{M})^{-1} \\ \cdot e(\sigma_4, \hat{g}_4)^{-1} \cdot e(\sigma_5, \hat{g}_5 \cdot \hat{g}_7{}^{M})^{-1} \cdot e(\Omega, \hat{g}_8)^{-1} \end{aligned}$$

as well as $e(\sigma_2, \hat{g}) = e(\sigma_3, \hat{\sigma_6})$ and $e(\sigma_4, \hat{g}) = e(\sigma_5, \hat{\sigma_6})$.

As in Section 3, the technique of [44] can be used to shorten the signature by one element of $\mathbb{G}$ as it allows replacing $(\tilde{z}, \tilde{r})$ by one element of $\mathbb{G}$.

We prove that, under the sole SXDH assumption, the scheme is secure in the sense of the natural security definition used by Boyen and Waters [20,21] which is recalled in Appendix C.3. In short, this definition requires that the adversary be unable to forge a valid signature for a pair $(\mathsf{ID}^{\star}, M^{\star})$ such that no private key query was made for $\mathsf{ID}^{\star}$ and no signing query was made for the pair $(\mathsf{ID}^{\star}, M^{\star})$.

Theorem 1. *The above hierarchical signature is secure under chosen-message attacks if the SXDH assumption holds in* $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$. (The proof is available in Appendix D).

A simple reduction shows that the signature scheme of Section 3 is F-unforgeable so long as the above scheme is a secure 2-level hierarchical signature.

Theorem 2. *The signature scheme of Section 3 is F-unforgeable under chosen-message attacks for the function $F(M) = \hat{g}^M$ if the SXDH assumption holds in* $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$. (The proof is available in Appendix E).

5 A Structure-Preserving Signature from the SXDH and XDLIN$_2$ Assumptions

Our F-unforgeable signature of Section 3 can be combined with the tagged one-time signature of Abe *et al.* [2] (or, more precisely, an adaption of [2] to asymmetric pairings) so as to obtain a new structure-preserving signature based on the SXDH and XDLIN$_2$ assumptions. Like [1], we obtain an SPS scheme based on simple assumptions with only 11 group elements per signature. However, only one of them has to be in $\hat{\mathbb{G}}$, instead of 4 in [1]. Considering that $\hat{\mathbb{G}}$ elements are at least twice as long to represent as those of $\mathbb{G}$, we thus shorten signatures by the equivalent of 3 elements of $\mathbb{G}$ (or 20%).

Our construction can be seen as an optimized instantiation of a general construction [1] that combines a tagged one-time signature and an SPS scheme which is only secure against extended random-message (XRMA) attacks. A tagged one-time signature (TOTS) is a signature scheme where each signature contains a single-use tag: namely, only one signature is generated w.r.t. each tag. The generic construction of [1] proceeds by certifying the tag of the TOTS scheme using an XRMA-secure SPS scheme. Specifically, our F-unforgeable signature assumes the role of the XRMA-secure signature and its shorter message space allows us to make the most of the optimal tag size of [2]. In [1], the proofs of XMRA security rely on the property that, when the reduction signs random groups elements of its choice, it is allowed to know their discrete logarithms. However, this property is only used in the security proof and not in the scheme itself. Here, we also use the discrete logarithm of the tag in the SPS construction itself, which allows our F-unforgeable signature to supersede the XRMA-secure signature. By exploiting the smaller message space of our F-unforgeable signature, we can leverage the optimal tag size of [2]. Unlike the SPS of [2], we do not need to expand the tag from one to three group elements before certifying it.

Keygen(cp, n) : given the length n of messages to be signed and common parameters cp specifying the description of bilinear groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$, do the following.

a. Generate a key pair $(\mathsf{sk}_{fsig}, \mathsf{pk}_{fsig}) \leftarrow \mathsf{Setup}(\mathsf{cp})$ for the F-unforgeable signature of Section 3. Namely,
   1. Choose $\omega, a \xleftarrow{R} \mathbb{Z}_p$, $g \xleftarrow{R} \mathbb{G}$, $\hat{g} \xleftarrow{R} \hat{\mathbb{G}}$ and set $h = g^a$, $\Omega = h^\omega$. Then, choose $v, w \xleftarrow{R} \mathbb{G}$.
   2. Define a matrix $\mathbf{M} = (M_{j,i})_{j,i}$ given by

$$\mathbf{M} = \left(\begin{array}{c|c|c|c|c|c} g & 1 & 1 & 1 & 1 & h \\ \hline v & g & 1 & h & 1 & 1 \\ \hline w & 1 & g & 1 & h & 1 \end{array}\right) \in \mathbb{G}^{3\times 6}. \tag{9}$$

   3. Generate a key pair $(\mathsf{sk}_{hsps}, \mathsf{pk}_{hsps})$ for the one-time linearly homomorphic signature of Section 2.2 in order to sign vectors of dimension $n = 6$. Let $\mathsf{sk}_{hsps} = \{(\chi_{0,i}, \gamma_{0,i})\}_{i=1}^6$ be the private key, of which the corresponding public key is $\mathsf{pk}_{hsps} = \big(\hat{g}_z,\ \hat{g}_r,\ \{\hat{g}_i\}_{i=1}^6\big)$.
   4. Using $\mathsf{sk}_{hsps} = \{\chi_{0,i}, \gamma_{0,i}\}_{i=1}^6$, generate one-time homomorphic signatures $\{(z_j, r_j)\}_{j=1}^3$ on the rows $\boldsymbol{M}_j = (M_{j,1}, \ldots, M_{j,6}) \in \mathbb{G}^6$ of $\mathbf{M}$. These are obtained as

$$(z_j, r_j) = \left(\prod_{i=1}^6 M_{j,i}^{-\chi_{0,i}},\ \prod_{i=1}^6 M_{j,i}^{-\gamma_{0,i}}\right), \qquad \forall j \in \{1,2,3\}$$

and, as part of the common reference string for the QA-NIZK proof system of [53], they will be included in the public key.

b. Generate a key pair $(\mathsf{pk}_{pots}, \mathsf{sk}_{pots})$ for the partial one-time structure-preserving signature of Abe *et al.* [1]. Namely, choose $w_z, w_r, \mu_z, \mu_u, w_t \xleftarrow{R} \mathbb{Z}_p$ and set

$$\begin{array}{lllll} \hat{G}_z = \hat{g}^{w_z}, & \hat{G}_r = \hat{g}^{w_r}, & \hat{G}_t = \hat{g}^{w_t}, & \hat{H}_z = \hat{g}^{\mu_z}, & \hat{H}_u = \hat{g}^{\mu_u} \\ G_z = g^{w_z}, & G_r = g^{w_r}, & G_t = g^{w_t}, & H_z = g^{\mu_z}, & H_u = g^{\mu_u} \end{array}$$

Then, for $i = 1$ to n, choose $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ and compute $\hat{G}_i = \hat{G}_z{}^{\chi_i} \cdot \hat{G}_r{}^{\gamma_i}$ and $\hat{H}_i = \hat{G}_z{}^{\chi_i} \cdot \hat{G}_r{}^{\delta_i}$. Define

$$\mathsf{pk}_{pots} := \big(G_z, G_r, G_t, H_z, H_u, \hat{G}_z, \hat{G}_r, \hat{G}_t, \hat{H}_z, \hat{H}_u, \{\hat{G}_i, \hat{H}_i\}_{i=1}^n\big)$$

and $\mathsf{sk}_{pots} := \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$.

The private key is $SK = (\omega, w_r, \mu_u, \mathsf{sk}_{pots})$ and the public key consists of

$$PK = \Big(g,\ h,\ \hat{g},\ (v, w),\ \Omega = h^{\omega},\ \mathsf{pk}_{pots},\ \mathsf{pk}_{hsps},\ \{(z_j, r_j)\}_{j=1}^3 \Big).$$

Sign$(SK, \boldsymbol{M})$: given $SK = (\omega, w_r, \mu_u, \mathsf{sk}_{pots})$ and $\boldsymbol{M} = (M_1, \ldots, M_n) \in \mathbb{G}^n$,

1. Choose $s, \tau \xleftarrow{R} \mathbb{Z}_p$ to compute

$$\begin{array}{lll} \sigma_1 = g^{\omega} \cdot (v^{\tau} \cdot w)^s, & \sigma_2 = g^{s \cdot \tau}, & \sigma_3 = g^s, \\ \sigma_4 = h^{s \cdot \tau} & \sigma_5 = h^s, & \tilde{\sigma_6} = \hat{g}^{\tau}. \end{array}$$

Then, generate a QA-NIZK proof that the vector $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \Omega)$ is in the row space of $\mathbf{M}$. This proof $(z, r) \in \mathbb{G}^2$ is computed as

$$z = z_1^{\omega} \cdot (z_2^{\tau} \cdot z_3)^s, \qquad r = r_1^{\omega} \cdot (r_2^{\tau} \cdot r_3)^s. \tag{10}$$

2. Choose $\zeta \xleftarrow{R} \mathbb{Z}_p$ and compute $Z = g^{\zeta} \cdot \prod_{i=1}^n M_i^{-\chi_i}$ as well as

$$R = (G_t^{\tau} \cdot G_z{}^{-\zeta})^{1/w_r} \cdot \prod_{i=1}^n M_i^{-\gamma_i}, \qquad U = (H_z^{-\zeta})^{1/\mu_u} \cdot \prod_{i=1}^n M_i^{-\delta_i}$$

Return the signature

$$\sigma = \big(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \tilde{\sigma_6}, z, r, Z, R, U\big) \in \mathbb{G}^5 \times \hat{\mathbb{G}} \times \mathbb{G}^5. \tag{11}$$

Verify$(PK, \sigma, \boldsymbol{M})$: given $\boldsymbol{M} = (M_1, \ldots, M_n) \in \mathbb{G}^n$, parse σ as per (11). Return 1 if and only if $e(\sigma_2, \hat{g}) = e(\sigma_3, \tilde{\sigma_6})$ and $e(\sigma_4, \hat{g}) = e(\sigma_5, \tilde{\sigma_6})$ as well as

$$\begin{aligned} e(z, \hat{g_z}) \cdot e(r, \hat{g_r}) &= \prod_{i=1}^5 e(\sigma_i, \hat{g}_i)^{-1} \cdot e(\Omega, \hat{g}_6)^{-1} \\ e(G_t, \tilde{\sigma_6}) &= e(Z, \hat{G_z}) \cdot e(R, \hat{G_r}) \cdot \prod_{i=1}^n e(M_i, \hat{G}_i) \\ 1_{\mathbb{G}_T} &= e(Z, \hat{H_z}) \cdot e(U, \hat{H_u}) \cdot \prod_{i=1}^n e(M_i, \hat{H}_i). \end{aligned} \tag{12}$$

Each signature requires 10 elements of $\mathbb{G}$ and one element of $\hat{\mathbb{G}}$. Using the optimized F-unforgeable signature based on the Jutla-Roy QA-NIZK proof [44], we can also save one more element of $\mathbb{G}$ and obtain signatures in $\mathbb{G}^9 \times \hat{\mathbb{G}}$, which shortens the signatures of Abe *et al.* [1] by 26%. In Appendix B, we give more detailed comparisons among all SPS based on non-interactive assumptions.

In the application to group signatures, it is desirable to minimize the number of signature components that need to appear in committed form. To this end, signatures must be randomizable in such a way that (σ_3, σ_5) can appear in the clear modulo a re-randomization of the underlying $s \in \mathbb{Z}_p$. To enable this randomization, it is necessary to augment signatures (similarly to [6]) with a randomization token $(g^\tau, h^\tau, v^\tau, z_2^\tau, r_2^\tau)$. We will prove that the scheme remains unforgeable even when the signing oracle also outputs these randomization tokens at each invocation.[5] We call this notion *extended existential unforgeability* (or EUF-CMA* for short).

When the re-randomization tokens are used, proving the knowledge of a signature on a committed message $\boldsymbol{M} \in \mathbb{G}^n$ requires $2n + 24$ elements of $\mathbb{G}$ and 12 elements of $\hat{\mathbb{G}}$. In comparison, the best previous solution of Abe *et al.* costs $2n + 26$ elements of $\mathbb{G}$ and 18 elements of $\hat{\mathbb{G}}$.

Theorem 3. *The scheme provides EUF-CMA* security if the SXDH and XDLIN$_2$ assumptions hold in* $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$. (The proof is given in Appendix F).

The details of the proof are available in Appendix F. In short, we consider two kinds of forgeries. In Type I forgeries, the adversary's forgery contains an element $\hat{\sigma_6}^\star$ that did not appear in any signature obtained by the forger during the game. In contrast, Type II forgeries are those for which $\hat{\sigma_6}^\star$ is recycled from a response of the signing oracle. It is easy to see that a Type I forger allows breaking the security of the F-unforgeable signature. As for Type II forgeries, they are shown to contradict the XDLIN$_2$ assumption via a careful adaptation of the proof given by Abe *et al.* for their TOTS scheme [2]. While the latter was originally presented in symmetric pairings, it goes through in Type 3 pairings modulo natural changes that consist in making sure that most handled elements of $\hat{\mathbb{G}}$ have a counterpart in $\mathbb{G}$. One difficulty is that, at each query, the reduction must properly simulate the randomization tokens $(v^\tau, g^\tau, h^\tau, z_2^\tau, r_2^\tau)$ as well as an instance of the F-unforgeable signature without knowing the discrete logarithm $\log_{\hat{g}}(\hat{\sigma_6}) = \hat{g}^\tau$ or that of its shadow $\log_g(\sigma_6) = g^\tau$ in $\mathbb{G}$. Fortunately, this issue can be addressed by letting the reduction know $\log_g(v)$ and $\log_g(w)$.

It is tempting to believe that our approach can generically combine a TOTS and an F-unforgeable signature. Unfortunately, we did not manage to get this idea to work in general. In particular, our F-unforgeable signature does not blend well with our attempts of SXDH-based TOTS schemes. When mixing it with the system of Abe *et al.* [1], we leveraged the algebraic compatibility of both schemes and the fact that the XDLIN$_2$ assumption allows input elements to be replicated in both $\mathbb{G}$ and $\hat{\mathbb{G}}$. Under the SXDH assumption, no such replication is possible and this makes it hard to rely on the SXDH assumption alone, for example.

In an independent work [49], Kiltz, Pan and Wee obtained even shorter signatures, which live in $\mathbb{G}^6 \times \hat{\mathbb{G}}$ under the SXDH assumption. On the other hand, their security reduction is looser than ours as the gap between the adversary's advantage and the reduction's probability to break the underlying assumption is quadratic (instead of linear in our case) in the number of signing queries.

6 A Publicly Verifiable Tag-Based Encryption Scheme

As a tool for constructing a CCA2-anonymous group signature, we describe a new tag-based encryption scheme [55,48] (see Appendix C.4 for definitions) which is inspired by the lossy encryption scheme [13] of [41]. In our group signature, we will exploit the fact that the DDH-based lossy encryption scheme of Bellare *et al.* [13] can also be seen as a Groth-Sahai commitment.

[5] Note, however, that the adversary is not required to produce any randomization token as part of its forgery.

Keygen(cp)**:** Given public parameters $\mathsf{cp} = (\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, p)$ specifying asymmetric bilinear groups of prime order $p > 2^\lambda$, conduct the following steps.

1. Choose $g, h \xleftarrow{R} \hat{\mathbb{G}}$. Choose $x, \alpha, \beta \xleftarrow{R} \mathbb{Z}_p$ and set $X_1 = g^x$, $X_2 = h^x$, $S = g^\alpha$, $T = g^\beta$, $W = h^\alpha$ and $V = h^\beta$.
2. Generate a key pair $(\mathsf{pk}'_{hsig}, \mathsf{sk}'_{hsig})$ for the homomorphic signature of Section 2.2 in order to sign vectors in $\mathbb{G}^3$. Let $\mathsf{pk}'_{hsig} = \big(\hat{G}_z, \hat{G}_r, \{\hat{G}_i\}_{i=1}^3\big)$ be the public key and let $\mathsf{sk}'_{hsig} = \{(\varphi_i, \vartheta_i)\}_{i=1}^3$ be the private key.
3. Use sk'_{hsig} to generate linearly homomorphic signatures $\{(Z_i, R_i)\}_{i=1}^4$ on the rows of the matrix

$$\mathbf{L} = \left(\begin{array}{c|c|c} g & 1 & T \\ \hline h & 1 & V \\ \hline 1 & g & S \\ \hline 1 & h & W \end{array}\right) \in \mathbb{G}^{4\times 3}$$

which form a subspace of rank 2 spanned by $(g, 1, T)$ and $(1, g, S)$. The private key is $\mathsf{sk} = x$ and the public key is

$$\mathsf{pk} := \Big(g,\ h,\ X_1,\ X_2,\ S,\ W,\ T,\ V,\ \mathsf{pk}'_{hsig},\ \{(Z_i, R_i)\}_{i=1}^4\Big).$$

Encrypt(pk, M, τ)**:** To encrypt $M \in \mathbb{G}$ under the tag τ, choose $\theta_1, \theta_2 \xleftarrow{R} \mathbb{Z}_p$ and compute the ciphertext $\boldsymbol{C} = (C_0, C_1, C_2, Z, R)$ as

$$\boldsymbol{C} = \big(M \cdot X_1^{\theta_1} \cdot X_2^{\theta_2},\ g^{\theta_1} \cdot h^{\theta_2},\ (S^\tau \cdot T)^{\theta_1} \cdot (W^\tau \cdot V)^{\theta_2},$$
$$(Z_3^\tau \cdot Z_1)^{\theta_1} \cdot (Z_4^\tau \cdot Z_2)^{\theta_2}, (R_3^\tau \cdot R_1)^{\theta_1} \cdot (R_4^\tau \cdot R_2)^{\theta_2}\big).$$

Here, (Z, R) serves as a proof that the vector (C_1, C_1^τ, C_2) is in the row space of $\mathbf{L}$ and satisfies

$$e(Z, \hat{G}_z) \cdot e(R, \hat{G}_r) = e(C_1, \hat{G}_1{}^\tau \cdot \hat{G}_2)^{-1} \cdot e(C_2, \hat{G}_2)^{-1} \tag{13}$$

Decrypt$(\mathsf{sk}, \boldsymbol{C}, \tau)$**:** Parse $\boldsymbol{C}$ as above. Return $\perp$ if (Z, R) does not satisfy (13). Otherwise, return $M = C_0/C_1^x$.

We note that the validity of $\boldsymbol{C}$ can also be verified by testing the equality $C_2 = C_1^{\alpha\cdot\tau+\beta}$ if α and β are included in the private key sk.

We also observe that, in the ciphertext, (C_0, C_1) form a Groth-Sahai commitment based on the DDH assumption in $\mathbb{G}$. If $\log_g(X_1) = \log_h(X_2)$, the commitment is extractable. Otherwise, it is perfectly hiding. In the following section, we will use this CCA2-secure scheme as a commitment that is extractable on all tags, except one $\tau^\star$ where it behaves as a perfectly hiding commitment. The above system achieves this while only expanding the original Groth-Sahai commitment (C_0, C_1) by 3 elements of $\mathbb{G}$.

This scheme will save our group signatures from having to contain (beyond (C_0, C_1)) an additional CCA2-secure encryption and a NIZK proof that the plaintext coincides with the content of a Groth-Sahai commitment. In its most efficient instantiation, the latter approach would require a publicly verifiable variant of the Cramer-Shoup encryption scheme (such as those suggested in [42,53]) with a ciphertext of the form $(g^r, h^r, M \cdot Y^r, (Y^\tau Z)^r, Z) \in \mathbb{G}^5$, where Z is a QA-NIZK proof of ciphertext validity obtained from [42,44]. Proving that the underlying M is consistent with a commitment $(C_0, C_1) = (M \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}, g^{\theta_1} \cdot h^{\theta_2})$ would require commitments to (θ_1, θ_2, r), which would incur 6 elements of $\hat{\mathbb{G}}$, and NIZK proofs for 2 linear multi-exponentiation equations. The overhead w.r.t. (C_0, C_1) would amount to 7 elements of $\mathbb{G}$ and 6 elements of $\hat{\mathbb{G}}$, instead of 3 elements of $\mathbb{G}$ here. The above technique thus allows saving the equivalent of 16 elements of $\mathbb{G}$. We thus believe this cryptosystem to be of interest in its own right since it can be used in a similar way to shorten other group signatures (e.g., [38]) based on Groth-Sahai proofs.

In Appendix G.1, we prove that the scheme satisfies the security definition given by Kiltz [48].

Theorem 4. *The above scheme is selective-tag weakly IND-CCA2-secure if the SXDH assumption holds.* (The proof is given in Appendix G.1).

7 Short Group Signatures in the BMW Model

The TBE scheme of Section 6 allows us to achieve anonymity in the CCA2 sense by encrypting an encoding of the group member's identifier. In order to minimize the signature length, we let the TBE ciphertext live in $\mathbb{G}$ instead of $\hat{\mathbb{G}}$. To open signatures in constant time, however, the opening algorithm uses the extraction trapdoor of a Groth-Sahai commitment in $\hat{\mathbb{G}}^2$ rather than the private key sk_{tbe} of the TBE system. The latter key is only used in the proof of anonymity where the reduction uses a somewhat inefficient opening algorithm of complexity $O(N)$.

Keygen(λ, N)**:** given a security parameter $\lambda \in \mathbb{N}$ and the desired number of users $N \in \mathsf{poly}(\lambda)$, choose asymmetric bilinear groups $\mathsf{cp} = (\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, p)$ of order $p > 2^\lambda$ and do the following.

1. Generate a key pair $(\mathsf{msk}, \mathsf{mpk})$ for the two-level hierarchical signature of Section 4. Let
$$\mathsf{mpk} := \Big((\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T),\ p,\ g,\ h,\ \hat{g},\ (t, v, w),\ \Omega = h^\omega,\ \mathsf{pk}_{hsps},\ \{(z_j, r_j)\}_{j=1}^4\Big)$$
be the master public key and $\mathsf{msk} := \omega \in \mathbb{Z}_p$ be the master secret key.
2. Generate a key pair $(\mathsf{sk}_{tbe}, \mathsf{pk}_{tbe})$ for the tag-based encryption scheme of Section 6. Let
$$\mathsf{pk}_{tbe} = \Big(g,\ h,\ X_1,\ X_2,\ S,\ W,\ T,\ V,\ \mathsf{pk}'_{hsig},\ \{(Z_i, R_i)\}_{i=1}^4\Big)$$
be the public key and $\mathsf{sk}_{tbe} = x$ be the underlying private key. For simplicity, the element g can be recycled from mpk.
3. Choose a vector $\hat{\boldsymbol{u}}_1 = (\hat{u}_{11}, \hat{u}_{12}) \xleftarrow{R} \hat{\mathbb{G}}^2$ and set $\hat{\boldsymbol{u}}_2 = \hat{\boldsymbol{u}}_1{}^\xi$, where $\xi \xleftarrow{R} \mathbb{Z}_p$. Also, define the vectors $\boldsymbol{u}_1 = (g, X_1)$ and $\boldsymbol{u}_2 = (h, X_2)$. These vectors will form Groth-Sahai CRSes $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ in the perfectly binding setting. Although sk_{tbe} serves as an extraction trapdoor for commitments generated on the CRS $(\boldsymbol{u}_1, \boldsymbol{u}_2)$, the group manager will more efficiently use $\zeta = \log_{\hat{u}_{11}}(\hat{u}_{12})$ to open signatures.
4. Choose a chameleon hash function $\mathsf{CMH} = (\mathsf{CMKg}, \mathsf{CMhash}, \mathsf{CMswitch})$ with a key pair (hk, tk) and randomness space $\mathcal{R}_{hash}$.
5. For each group member i, choose an identifier $\mathsf{ID}_i \xleftarrow{R} \mathbb{Z}_p$ and use msk to compute $K_{\mathsf{ID}_i} = (K_1, K_2, K_3, K_4, K_5, K_6, \hat{K}_7, z, r, z_d, r_d)$, where
$$\begin{array}{lll} K_1 = g^\omega \cdot (v^{\mathsf{ID}_i} \cdot w)^s \cdot, & K_2 = g^{s \cdot \mathsf{ID}_i}, & K_3 = g^s \\ K_4 = h^{s \cdot \mathsf{ID}_i} & K_5 = h^s & K_6 = t^s \\ z = z_1^\omega \cdot (z_2^{\mathsf{ID}_i} \cdot z_3)^s & r = r_1^\omega \cdot (r_2^{\mathsf{ID}_i} \cdot r_3)^s & \hat{K}_7 = \hat{g}^{\mathsf{ID}_i} \end{array}$$
and $(z_d, r_d) = (z_4^s, r_4^s)$. For each $i \in \{1, \ldots, N\}$, the i-th group member's private key is $\mathsf{gsk}[i] = (\mathsf{ID}_i, K_{\mathsf{ID}_i})$.

The group public key consists of
$$\mathsf{gpk} := \Big((\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T),\ \mathsf{mpk},\ \mathsf{pk}_{tbe},\ (\boldsymbol{u}_1, \boldsymbol{u}_2),\ (\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2),\ \mathsf{CMH},\ hk\Big),$$
whereas the group manager secret key is $\mathsf{gsk} := \big(\mathsf{msk}, \zeta = \log_{\hat{u}_{11}}(\hat{u}_{12})\big)$.

Sign$(\mathsf{gpk}, \mathsf{gsk}[i], M)$**:** In order to sign a message $M \in \mathbb{Z}_p$ using the i-th group member's private key $\mathsf{gsk}[i] = (\mathsf{ID}_i, K_{\mathsf{ID}_i})$, conduct the following steps.

1. Using $K_{\mathsf{ID}_i} = (K_1, K_2, K_3, K_4, K_5, K_6, \hat{K_7}, z, r, z_d, r_d)$, derive a second-level hierarchical signature. Namely, choose $s' \xleftarrow{R} \mathbb{Z}_p$ and compute

$$\begin{aligned}\sigma_1 &= K_1 \cdot K_6^M \cdot (v^{\mathsf{ID}_i} \cdot t^M \cdot w)^{s'} \\ &= g^{\omega} \cdot (v^{\mathsf{ID}_i} \cdot t^M \cdot w)^{\tilde{s}} \\ \sigma_4 &= K_4 \cdot h^{s' \cdot \mathsf{ID}_i} = h^{\tilde{s} \cdot \mathsf{ID}_i}\end{aligned} \qquad \begin{aligned}\sigma_2 &= K_2 \cdot g^{s' \cdot \mathsf{ID}_i} = g^{\tilde{s} \cdot \mathsf{ID}_i} \\ \sigma_3 &= K_3 \cdot g^{s'} = g^{\tilde{s}} \\ \sigma_5 &= K_5 \cdot h^{s'} = h^{\tilde{s}},\end{aligned}$$

and $\hat{\sigma_6} = \hat{K_7}$, where $\tilde{s} = s + s'$, as well as

$$\begin{aligned}\tilde{z} &= z \cdot z_d^M \cdot (z_2^{\mathsf{ID}_i} \cdot z_4^M \cdot z_3)^{s'} \\ &= z_1^{\omega} \cdot (z_2^{\mathsf{ID}_i} \cdot z_4^M \cdot z_3)^{\tilde{s}}\end{aligned} \qquad \begin{aligned}\tilde{r} &= r \cdot r_d^M \cdot (r_2^{\mathsf{ID}_i} \cdot r_4^M \cdot r_3)^{s'} \\ &= r_1^{\omega} \cdot (r_2^{\mathsf{ID}_i} \cdot r_4^M \cdot r_3)^{\tilde{s}}.\end{aligned}$$

2. Choose $\theta_1, \ldots, \theta_{12} \xleftarrow{R} \mathbb{Z}_p$ and compute Groth-Sahai commitments

$$\begin{aligned}\boldsymbol{C}_{\sigma_1} &= (1, \sigma_1) \cdot \boldsymbol{u_1}^{\theta_1} \cdot \boldsymbol{u_2}^{\theta_2}, \\ \boldsymbol{C}_{\sigma_4} &= (1, \sigma_4) \cdot \boldsymbol{u_1}^{\theta_5} \cdot \boldsymbol{u_2}^{\theta_6}, \\ \boldsymbol{C}_{\tilde{z}} &= (1, \tilde{z}) \cdot \boldsymbol{u_1}^{\theta_9} \cdot \boldsymbol{u_2}^{\theta_{10}},\end{aligned} \qquad \begin{aligned}\boldsymbol{C}_{\sigma_2} &= (1, \sigma_2) \cdot \boldsymbol{u_1}^{\theta_3} \cdot \boldsymbol{u_2}^{\theta_4}, \\ \boldsymbol{C}_{\hat{\sigma_6}} &= (1, \hat{\sigma_6}) \cdot \hat{\boldsymbol{u_1}}^{\theta_7} \cdot \hat{\boldsymbol{u_2}}^{\theta_8}. \\ \boldsymbol{C}_{\tilde{r}} &= (1, \tilde{r}) \cdot \boldsymbol{u_1}^{\theta_{11}} \cdot \boldsymbol{u_2}^{\theta_{12}}\end{aligned}$$

Note that $\boldsymbol{C}_{\sigma_2}$ can be written as $(C_1, C_0) = (g^{\theta_3} \cdot h^{\theta_4}, \sigma_2 \cdot X_1^{\theta_3} \cdot X_2^{\theta_4})$.

3. Generate Groth-Sahai NIWI proofs $\boldsymbol{\pi}_1 \in \hat{\mathbb{G}}^2$, $\boldsymbol{\pi}_2 \in \mathbb{G}^2 \times \hat{\mathbb{G}}^2$ and $\boldsymbol{\pi}_3 \in \mathbb{G}^2 \times \hat{\mathbb{G}}^2$ that committed variables $(\tilde{z}, \tilde{r}, \sigma_1, \sigma_2, \sigma_4, \hat{\sigma_6})$ satisfy

$$\begin{aligned}e(\boxed{\tilde{z}}, \hat{g_z}) \cdot e(\boxed{\tilde{r}}, \hat{g_r}) = e(\boxed{\sigma_1}, \hat{g_1})^{-1} \cdot e(\boxed{\sigma_2}, \hat{g_2})^{-1} \cdot e(\sigma_3, \hat{g_3} \cdot \hat{g_6}^M)^{-1} \\ \cdot e(\boxed{\sigma_4}, \hat{g_4})^{-1} \cdot e(\sigma_5, \hat{g_5} \cdot \hat{g_7}^M)^{-1} \cdot e(\Omega, \hat{g_8})^{-1}\end{aligned} \tag{14}$$

and

$$e(\boxed{\sigma_2}, \hat{g}) = e(\sigma_3, \boxed{\hat{\sigma_6}}), \qquad e(\boxed{\sigma_4}, \hat{g}) = e(\sigma_5, \boxed{\hat{\sigma_6}}). \tag{15}$$

Since (14) is a linear pairing product equation where all variables are in $\mathbb{G}$, it only costs two elements of $\hat{\mathbb{G}}$. Each equation of (15) contains one variable in $\mathbb{G}$ and $\hat{\mathbb{G}}$ and thus takes two proofs element in each group.

4. Choose $r_{hash} \xleftarrow{R} \mathcal{R}_{hash}$ and compute a chameleon hash value

$$\tau = \mathsf{CMhash}(hk, (\boldsymbol{C}_{\sigma_1}, \boldsymbol{C}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_{\tilde{z}}, \boldsymbol{C}_{\tilde{r}}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3), r_{hash}).$$

Then, using τ and $(\theta_3, \theta_4) \in \mathbb{Z}_p^2$, compute $C_2 = (S^{\tau} \cdot T)^{\theta_3} \cdot (W^{\tau} \cdot V)^{\theta_4}$. Using pk'_{hsig} as a CRS, generate a QA-NIZK argument

$$(Z, R) = \big((Z_3^{\tau} \cdot Z_1)^{\theta_3} \cdot (Z_4^{\tau} \cdot Z_2)^{\theta_4}, (R_3^{\tau} \cdot R_1)^{\theta_1} \cdot (R_4^{\tau} \cdot R_2)^{\theta_2}\big)$$

that the vector $(C_1, C_1^{\tau}, C_2) \in \mathbb{G}^3$ is in the row space of $\mathbf{L}$. This allows turning $\boldsymbol{C}_{\sigma_2} = (C_1, C_0)$ into a TBE ciphertext $\tilde{\boldsymbol{C}}_{\sigma_2} = (C_0, C_1, C_2, Z, R)$ as

$$\begin{aligned}\tilde{\boldsymbol{C}}_{\sigma_2} = \big(\sigma_2 \cdot X_1^{\theta_3} \cdot X_2^{\theta_4},\ g^{\theta_3} \cdot h^{\theta_4},\ (S^{\tau} \cdot T)^{\theta_3} \cdot (W^{\tau} \cdot V)^{\theta_4}, \\ (Z_3^{\tau} \cdot Z_1)^{\theta_3} \cdot (Z_4^{\tau} \cdot Z_2)^{\theta_4}, (R_3^{\tau} \cdot R_1)^{\theta_1} \cdot (R_4^{\tau} \cdot R_2)^{\theta_2}\big) \in \mathbb{G}^5\end{aligned}$$

for the tag τ. Note that $\tilde{\boldsymbol{C}}_{\sigma_2}$ contains the original commitment $\boldsymbol{C}_{\sigma_2}$.

Return the signature

$$\sigma = \big(\boldsymbol{C}_{\sigma_1}, \tilde{\boldsymbol{C}}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_{\tilde{z}}, \boldsymbol{C}_{\tilde{r}}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, r_{hash}\big) \tag{16}$$

Verify$(\mathsf{gpk}, M, \sigma)$**:** Parse the signature σ as above. Return 1 if and only if it holds that: (i) The proofs $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3$ verify; (ii) $\tilde{\boldsymbol{C}}_{\sigma_2}$ is a valid TBE ciphertext (i.e., (13) holds) for the tag $\tau = \mathsf{CMhash}(hk, (\boldsymbol{C}_{\sigma_1}, \boldsymbol{C}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_{\tilde{z}}, \boldsymbol{C}_{\tilde{r}}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3), r_{hash})$.

Open$(\mathsf{gpk}, \mathsf{gmsk}, M, \sigma)$**:** To open σ using $\mathsf{gmsk} = (\mathsf{msk}, \zeta)$, parse σ as in (16) and return $\perp$ if it is not a valid signature w.r.t. gpk and M. Otherwise, use $\zeta = \log_{\hat{u}_{11}}(\hat{u}_{12})$ to decrypt the Elgamal ciphertext $\boldsymbol{C}_{\hat{\sigma_6}} \in \hat{\mathbb{G}}^2$. Then, check if the resulting plaintext is $\hat{g}^{\mathsf{ID}}$ for some group member's identifier ID. If so, output ID. Otherwise, return $\perp$.

The signature consists of 19 elements of $\mathbb{G}$, 8 elements of $\hat{\mathbb{G}}$ and one element of $\mathbb{Z}_p$. If each element of $\mathbb{G}$ (resp. $\hat{\mathbb{G}}$) has a 256-bit (resp. 512-bit) representation, the entire signature fits within 9216 bits (or 1.125 kB). By using the technique of Jutla and Roy [44] to shorten the hierarchical signature, it is possible to shorten the latter by one group element (as explained in Section 4), which saves two elements of $\mathbb{G}$ in the group signature without modifying the underlying assumption. In this case, the signature length reduces to 8704 bits (or 1.062 kB). Using the technique of Boyen, Mei and Waters [19], it is also possible to eliminate the randomness r_{hash} and replace the chameleon hash function by an ordinary collision-resistant hash function, as explained in Appendix G.2. By doing so, at the expense of a group public key made of $\Theta(\lambda)$ elements of $\hat{\mathbb{G}}$, we can further compress signatures down to 8448 bits (or 1.031 kB).

If we settle for a weaker flavor of CPA-anonymity (i.e., without a signature opening oracle), the commitment $\tilde{\boldsymbol{C}}_{\sigma_2}$ can be replaced by an ordinary Groth-Sahai commitment $\boldsymbol{C}_{\sigma_2}$ so as to further save three elements of $\mathbb{G}$ (or 768 bits). In this CPA-anonymous variant, the signature length drops to 7936 bits.

To give a concrete comparison with earlier constructions, an implementation of the Boyen-Waters group signature [21] in asymmetric prime order groups requires 10 elements of $\mathbb{G}$ and 8 elements of $\hat{\mathbb{G}}$ for a total of 6656 bits per signature. However, besides the SXDH assumption, the resulting scheme relies on the non-standard q-Hidden Strong Diffie-Hellman assumption [21] and only provides anonymity in the CPA sense.

Theorem 5. *The scheme provides full traceability under the SXDH assumption.*

The proof of Theorem 5 relies on the unforgeability of the two-level hierarchical signature of Section 4. By preparing extractable Groth-Sahai CRSes $(\boldsymbol{u_1}, \boldsymbol{u_2})$ and $(\hat{\boldsymbol{u_1}}, \hat{\boldsymbol{u_2}})$, the reduction can always turn a full traceability adversary (whose definition is recalled in Appendix H.1) into a forger for the hierarchical signature. The proof is straightforward and the details are omitted.

Theorem 6. *The scheme provides full anonymity assuming that: (i) The SXDH assumption holds in* $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$*; (ii)* CMhash *is a collision-resistant chameleon hash function.* (The proof is given in Appendix I.1).

In Appendix J, we extend the above system to obtain dynamic group signatures based on the SXDH and XDLIN_2 assumption. The signature length is only 1.8 kB, which gives us the shortest dynamic group signatures based on constant-size assumptions to date. The construction builds on our structure-preserving signature and the encryption scheme of Section 6 in a modular manner. Detailed efficiency comparisons are given in the table in Appendix J.

Acknowledgements

The first author's work was supported by the "Programme Avenir Lyon Saint-Etienne de l'Université de Lyon" in the framework of the programme "Inverstissements d'Avenir" (ANR-11-IDEX-0007). The second author was supported by the European Research Council (FP7/2007-2013 Grant Agreement no. 339563 CryptoCloud). Part of this work of the third author was done while visiting the Simons Institute for Theory of Computing, U.C. Berkeley.

References

1. M. Abe, M. Chase, B. David, M. Kohlweiss, R. Nishimaki, M. Ohkubo. Constant-Size Structure-Preserving Signatures: Generic Constructions and Simple Assumptions. In *Asiacrypt'12*, *LNCS* 7658, pp. 4–24, 2012.
2. M. Abe, B. David, M. Kohlweiss, R. Nishimaki, M. Ohkubo. Tagged One-Time Signatures: Tight Security and Optimal Tag Size. In *PKC'13*, *LNCS* 7778, pp. 312–331, 2013.
3. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo. Structure-Preserving Signatures and Commitments to Group Elements. In *Crypto'10*, *LNCS* 6223, pp. 209–236, 2010.
4. M. Abe, J. Groth, K. Haralambiev, M. Ohkubo. Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups. In *Crypto'11*, *LNCS* 6841, pp. 649–666, 2011.
5. M. Abe, J. Groth, M. Ohkubo. Separating Short Structure-Preserving Signatures from Non-interactive Assumptions In *Asiacrypt'11*, *LNCS* 7073, pp. 628–646, 2011.
6. M. Abe, J. Groth, M. Ohkubo, M. Tibouchi. Unified, Minimal and Selectively Randomizable Structure-Preserving Signatures. In *TCC'14*, *LNCS* 8349, pp. 688–712, 2014.
7. M. Abe, J. Groth, M. Ohkubo, M. Tibouchi. Structure-Preserving Signatures in Type II Pairings. In *Crypto'14*, *LNCS* 8616, pp. 390–407, 2014.
8. M. Abe, K. Haralambiev, M. Ohkubo. Signing on Elements in Bilinear Groups for Modular Protocol Design. Cryptology ePrint Archive: Report 2010/133, 2010.
9. G. Ateniese, J. Camenisch, S. Hohenberger, B. de Medeiros. Practical group signatures without random oracles. Cryptology ePrint Archive: Report 2005/385, 2005.
10. G. Ateniese, J. Camenisch, M. Joye, G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Crypto'00*, *LNCS* 1880, pp. 255–270, 2000.
11. M. Belenkiy, M. Chase, M. Kohlweiss, A. Lysyanskaya. P-signatures and Non-interactive Anonymous Credentials. In *TCC'08*, *LNCS* 4948, pp. 356–374, 2008.
12. M. Bellare, D. Micciancio, B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Eurocrypt'03*, *LNCS* 2656, pp. 614–629, 2003.
13. M. Bellare, D. Hofheinz, S. Yilek. Possibility and Impossibility Results for Encryption and Commitment Secure under Selective Opening. In *Eurocrypt'09*, *LNCS* 5479, pp. 1–35, 2009.
14. M. Bellare, P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *1st ACM Conference on Computer and Communications Security*, pp. 62–73, ACM Press, 1993.
15. M. Bellare, H. Shi, C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA'05*, *LNCS* 3376, pp. 136–153, 2005.
16. D. Boneh, X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 223–238, 2004.
17. D. Boneh, X. Boyen, E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Eurocrypt'05*, *LNCS* 3494, pp. 440–456, 2005.
18. D. Boneh, X. Boyen, H. Shacham. Short Group Signatures. In *Crypto'04*, *LNCS* 3152, pp. 41–55. Springer, 2004.
19. X. Boyen, Q. Mei, B. Waters. Direct Chosen-Ciphertext Security from Identity-Based Techniques. In *ACM-CCS'05*, pp. 320–329, ACM Press, 2006.
20. X. Boyen, B. Waters. Compact Group Signatures Without Random Oracles. In *Eurocrypt'06*, *LNCS* 4004, pp. 427–444, Springer, 2006.
21. X. Boyen, B. Waters. Full-Domain Subgroup Hiding and Constant-Size Group Signatures. In *PKC'07*, *LNCS* 4450, pp. 1–15, 2007.
22. J. Camenisch, N. Chandran, V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *Eurocrypt '09*, LNCS 5479, pp. 351–368, Springer, 2009.
23. J. Camenisch, M. Dubovitskaya, K. Haralambiev. Efficient Structure-Preserving Signature Scheme from Standard Assumptions. In *SCN'12*, LNCS 7485, pp. 76–94, Springer, 2012.
24. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *Journal of the ACM*, 51(4):557–594, 2004.
25. R. Canetti, S. Halevi, J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *Eurocrypt'04*, *LNCS* 3027, pp. 207–222, 2004.
26. J. Cathalo, B. Libert, M. Yung. Group Encryption: Non-Interactive Realization in the Standard Model. In *Asiacrypt'09*, *LNCS* 5912, pp. 179–196, 2009.
27. M. Chase, M. Kohlweiss. A Domain Transformation for Structure-Preserving Signatures on Group Elements. Cryptology ePrint Archive: Report 2011/342, 2011.
28. M. Chase, M. Kohlweiss. A New Hash-and-Sign Approach and Structure-Preserving Signatures from DLIN. In *SCN'12*, *LNCS* 7485, pp. 131–148, 2012.
29. D. Chaum, E. van Heyst. Group Signatures. In *Eurocrypt'91*, *LNCS* 547, pp. 257–265, Springer, 1991.
30. R. Cramer, V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Crypto'98*, LNCS 1462, pp. 13–25, 1998.
31. R. Cramer, V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In *Eurocrypt'02*, *LNCS* 2332, pp. 45–64, 2002.

32. C. Delerablée, D. Pointcheval. Dynamic Fully Anonymous Short Group Signatures. In *Vietcrypt'06*, *LNCS* 4341, pp. 193–210, Springer, 2006.
33. G. Fuchsbauer. Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. Cryptology ePrint Archive: Report 2009/320, 2009.
34. M. Gerbush, A. Lewko, A. O 'Neill, B. Waters. Dual form signatures: An approach for proving security from static assumptions. In *Asiacrypt '12*, LNCS 7658, pp. 25–42, Springer, 2012.
35. M. Green, S. Hohenberger. Universally Composable Adaptive Oblivious Transfer. In *Asiacrypt'06*, *LNCS* 5350, pp. 179–197. Springer, 2006.
36. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In *Eurocrypt'06*, *LNCS* 4004, pp. 339–358. Springer, 2006.
37. J. Groth. Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In *Asiacrypt'06*, *LNCS* 4284, pp. 444–459, Springer, 2006.
38. J. Groth. Fully anonymous group signatures without random oracles. In *Asiacrypt'07*, *LNCS* 4833, pp. 164–180. Springer, 2007.
39. J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pp. 415–432, 2008.
40. D. Hofheinz, T. Jager. Tightly Secure Signatures and Public-Key Encryption. In *Crypto'12*, *LNCS* 7417, pp. 590–607, 2012.
41. B. Hemenway, B. Libert, R. Ostrovsky, D. Vergnaud. Lossy Encryption: Constructions from General Assumptions and Efficient Selective Opening Chosen Ciphertext Security. In *Asiacrypt'11*, LNCS 7073, pp. 70–88, 2011.
42. C. Jutla, A. Roy. Relatively-Sound NIZKs and Password-Based Key-Exchange. In *PKC'12*, *LNCS* 7293, pp. 485–503, 2012.
43. C. Jutla, A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In *Asiacrypt '13*, LNCS 8269, pp. 1–20, Springer, 2013.
44. C. Jutla, A. Roy. Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In *Crypto '14*, LNCS 8617, pp. 295–312, Springer, 2014.
45. A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In *Eurocrypt'05*, LNCS 3494, pages 198–214, 2005.
46. A. Kiayias, M. Yung. Secure scalable group signature with dynamic joins and separable authorities. International Journal of Security and Networks (IJSN) Vol. 1, No. 1/2, pp. 24–45, 2006.
47. E. Kiltz, A. Mityagin, S. Panjwani, B. Raghavan. Append-Only Signatures. In *ICALP'05*, LNCS 3580, pp. 434–445, 2005.
48. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC'06*, *LNCS* 3876, pp. 581–600, 2006.
49. E. Kiltz, J. Pan, H. Wee. Structure-Preserving Signatures from Standard Assumptions, Revisited. In *Crypto'15*, *LNCS* series, 2015.
50. H. Krawczyk and T. Rabin. Chameleon signatures. In *NDSS'00*, 2000.
51. A. Lewko, B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC '10*, LNCS 5978, pp. 455–479, Springer, 2010.
52. B. Libert, T. Peters, M. Joye, M. Yung. Linearly homomorphic structure-preserving signatures and their applications. In *Crypto '13*, LNCS 8043, pp. 289–307, Springer, 2013.
53. B. Libert, T. Peters, M. Joye, M. Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In *Eurocrypt '14*, LNCS 8441, pp. 514–532, Springer, 2014.
54. B. Libert, M. Joye, M. Yung, T. Peters. Concise multi-challenge CCA-secure encryption and signatures with almost tight security. In *Asiacrypt '14*, LNCS 8874, pp. 1–21, Springer, 2014.
55. P. MacKenzie, M. Reiter, K. Yang. Alternatives to Non-malleability: Definitions, Constructions, and Applications. In *TCC'04*, LNCS 2951, pp. 171–190, 2004.
56. M. Naor. On Cryptographic Assumptions and Challenges. In *Crypto'03*, *LNCS* 2729, pp. 96–109. Springer-Verlag, 2003.
57. L. Nguyen, R. Safavi-Naini. Efficient and Provably Secure Trapdoor-Free Group Signature Schemes from Bilinear Pairings. In *Asiacrypt'04*, *LNCS* 3329, pp. 372–386. Springer, 2004.
58. Y. Sakai, J. Schuldt, K. Emura, G. Hanaoka, K. Ohta. On the Security of Dynamic Group Signatures: Preventing Signature Hijacking. In *PKC 2012*, *LNCS* 7293, pp. 715–732, 2012.
59. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *Crypto'84*, *LNCS* 196, pp. 47–53, 1984.
60. V. Shoup. A proposal for an ISO standard for public key encryption. Manuscript, December 20, 2001.
61. B. Waters. Efficient identity-based encryption without random oracles. In *Eurocrypt'05*, *LNCS* 3494, pp. 114–127. Springer, 2005.
62. B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *Crypto '09*, LNCS 5677, pp. 619–636, Springer, 2009.

A Groth-Sahai Proofs

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

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

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

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

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

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

B Comparisons Among Structure-Preserving Signatures

Abe, Haralambiev, Groth, Ohkubo [4] gave lower bounds on the complexity of structure-preserving signatures. In asymmetric pairings, they showed that the signature size must be at least 3 group elements and they provided a scheme matching these bounds. The security proof, however, relies on interactive assumptions which are not falsifiable [56]. At the cost of increasing the signature size by 1 or 3 group elements (depending whether messages contain elements from both source groups), a variant of their scheme can be proved secure under a "q-type" assumption which, while non-interactive, remains pretty *ad hoc*. In 2011, Abe, Groth Ohkubo [5] subsequently showed that, under algebraic reductions, the optimal bound of 3 elements cannot be reached under any non-interactive

assumption in Type 3 pairings. Under a q-type assumption, the bound of 4 group elements was reached by the scheme of [4] for unilateral vectors of group elements with a restricted message space. To our knowledge, no impossibility result or lower bound is known for SPS schemes based on simple assumption.

Table 1. Efficiency comparison of constant-size structure preserving signatures.

Schemes	Public key†	Signature	#(PPEs)	Pairings	Assumptions
AHO10 [3,8]	$12+2n$	7	2	Type I	q-SFP
ACDKNO12 [1]	$25+2n$	17	9	Type I	DLIN
ADKNO13 [2]	$20+2n$	14	7	Type I	DLIN
AHO10 [3,8]	$(4, 8+2n)$	$(5, 2)$	2	Type III	q-SFP
AGHO11 [4]	$(1, 4+2n)$	$(3, 1)$	2	Type III	q-type
ACDKNO12 [1]	$(7, 13+n)$	$(7, 4)$	5	Type III	SXDH, XDLIN_1
Section 5	$(11, 14+2n)$	$(10, 1)$	5	Type III	SXDH, XDLIN_2
Section 5 + [44]	$(8, 13+2n)$	$(9, 1)$	5	Type III	SXDH, XDLIN_2

† We assume unilateral messages and n denotes the number of elements of $\mathbb{G}$ per message.

Under The DLIN assumption, Chase and Kohlweiss [27] presented a framework for building SPS schemes from a stateful signature that is F-unforgeable under weak chosen message attacks and and efficient non-interactive zero-knowledge proof. Their scheme is proven secure under the DLIN assumption, but the size of a signature is $100 + 24 \cdot n + 9\dot{x}$, where n is the number of group elements signed and x determines an upper bound on the number of signatures produced using a given key pair. Camenisch, Dubovitskaya and Haralambiev [23] proposed the first scheme under a simple assumption.

Later on, Abe *et al.* [1,2] gave the first constructions allowing to sign n group elements at once using a constant number of group elements.

C Definitions for Involved Primitives

C.1 Quasi-Adaptive NIZK Arguments

Quasi-Adaptive NIZK (QA-NIZK) proofs [43] are NIZK proofs where the CRS is allowed to depend on the specific language for which proofs have to be generated. The CRS is divided into a fixed part Γ, produced by an algorithm K_0, and a language-dependent part ψ. However, there should be a single simulator for the entire class of languages.

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

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

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

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

Quasi-Adaptive Completeness:

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

Quasi-Adaptive Soundness:

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

Quasi-Adaptive Zero-Knowledge:

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

where

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

C.2 *F*-Unforgeable Signatures

In the syntax of Belenkiy *et al.* [11], a signature scheme consists of algorithms $(\mathsf{Keygen},\mathsf{Sign},\mathsf{Verify})$ which operate in the usual way with the difference that Keygen takes as input common public parameters cp, which typically specifies the abelian groups to be used in the scheme, produced by a setup algorithm on input of a security parameter λ.

Definition 3. *Let F be an efficiently computable (but not necessarily efficiently invertible) bijection. A signature scheme $(\mathsf{Keygen},\mathsf{Sign},\mathsf{Verify})$ is F-unforgeable if no PPT adversary has non-negligible advantage in the following game:*

1. *The challenger generates $\mathsf{cp}\leftarrow\mathsf{Setup}(\lambda)$ and $(\mathsf{sk},\mathsf{pk})\leftarrow\mathsf{Keygen}(\mathsf{cp})$ and gives $(\mathsf{cp},\mathsf{pk})$ to $\mathcal{A}$.*
2. *On polynomially-many occasions, $\mathcal{A}$ chooses messages M and obtains $\sigma\leftarrow\mathsf{Sign}(\mathsf{sk},M)$.*
3. *$\mathcal{A}$ outputs a pair (y,σ) and wins if $\mathsf{Verify}(\mathsf{pk},F^{-1}(y),\sigma))=1$ and $F^{-1}(y)\notin Q$, where Q denotes the set of messages queried at step 2.*

C.3 Hierarchical Signatures

A two-level (blinded) hierarchical signature is a tuple $(\mathsf{Setup},\mathsf{Extract},\mathsf{Sign},\mathsf{Verify})$ with the following specification.

$\mathsf{Setup}(\lambda)$: Takes as input a security parameter $\lambda\in\mathbb{N}$ and outputs a master key pair $(\mathsf{msk},\mathsf{mpk})$.

$\mathsf{Extract}(\mathsf{msk},\mathsf{ID})$: Takes as input an identity ID and the master secret key msk. It outputs a private key K_{ID} for the identity ID.

$\mathsf{Sign}(\mathsf{mpk},K_{\mathsf{ID}},M)$: Given the master public key, a private key K_{ID} for the identity ID and a message, this randomized algorithm outputs an a signature for the hierarchical message (ID,M).

$\mathsf{Verify}(\mathsf{mpk},\sigma,M)$: Given a master public key mpk, a candidate signature σ and a message M, this algorithm outputs 1 if σ is deemed to be a valid signature for the identity-message pair (ID,M), where ID is implicitly encoded in σ, and 0 otherwise.

The difference with ordinary two-level hierarchical signatures (a.k.a. identity-based signatures) lies in that the signer's identity is not explicitly known to the verifier, but is still uniquely determined (e.g., via an injective but not necessarily efficiently invertible encoding) by the signature.

Definition 4. *A two-level (blinded) hierarchical signature is secure under chosen-message attacks if no PPT adversary has non-negligible advantage in the following game.*

1. *The challenger generates $(\mathsf{mpk},\mathsf{msk})\leftarrow\mathsf{Setup}(\lambda)$ and gives mpk to the adversary $\mathcal{A}$.*
2. *In a learning phase, $\mathcal{A}$ interleaves the following queries on a polynomial number of occasions.*
   - *Extraction queries: $\mathcal{A}$ chooses an identity ID and obtains the private key $K_{\mathsf{ID}}\leftarrow\mathsf{Extract}(\mathsf{msk},\mathsf{ID})$ from the challenger.*

- *Signing queries:* $\mathcal{A}$ *chooses an identity-message pair* (ID, M). *The challenger generates* $K_{\mathsf{ID}} \leftarrow \mathsf{Extract}(\mathsf{msk}, \mathsf{ID})$ *and returns a signature* $\sigma \leftarrow \mathsf{Sign}(\mathsf{mpk}, K_{\mathsf{ID}}, M)$ *to* $\mathcal{A}$.

3. $\mathcal{A}$ *outputs a pair* $(\sigma^\star, M^\star)$ *and wins if: (i)* $\mathsf{Verify}(\mathsf{mpk}, \sigma^\star, M^\star) = 1$*; (ii) The identity* $\mathsf{ID}^\star$*, which is uniquely determined by* $\sigma^\star$*, was never queried to the extraction oracle at step 2; (iii) The pair* $(\mathsf{ID}^\star, M^\star)$ *was never submitted to the signing oracle at step 2.*

The adversary's advantage is its probability of success taken over all coin tosses.

For simplicity, the above definition assumes that, for a given identity ID, the distribution of signatures does not depend on which specific private key K_{ID} is used to run Sign.

C.4 Tag-Based Encryption

A tag-based encryption scheme (TBE) [55,48] is a public key cryptosystem where the encryption and decryption algorithms take an additional input, named the *tag*, which is a binary string of appropriate length with no particular structure. A TBE scheme consists of a triple $(\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ of efficient algorithms where, on input of a security parameter λ, Keygen outputs a private/public key pair (pk, sk); $\mathsf{Encrypt}$ is a randomized algorithm that outputs a ciphertext C on input of a public key pk, a string τ – called *tag* – and a message M; $\mathsf{Decrypt}(sk, \tau, C)$ is the decryption algorithm that takes as input a secret key sk, a tag τ and a ciphertext C and returns a plaintext M or $\perp$. Correctness requires that for all $\lambda \in \mathbb{N}$, all key pairs $(pk, sk) \leftarrow \mathsf{Keygen}(\lambda)$, all tags τ and any plaintext M, it holds that $M \leftarrow \mathsf{Decrypt}(sk, \mathsf{Encrypt}(pk, M, \tau), \tau)$.

In Section 6, we use a similar syntax with the only difference that the Keygen algorithm takes as input pre-existing public parameters cp.

Kiltz [48] showed that, in order to build chosen-ciphertext-secure encryption schemes from tag-based encryption via the Canetti-Halevi-Katz paradigm [25], it is sufficient to start from a TBE system satisfying a weaker definition than in [55]. This weaker definition is as follows.

Definition 5 ([48]). *A tag-based encryption (TBE) scheme is weakly secure against selective-tag chosen-ciphertext attacks if no PPT adversary has non-negligible advantage in the following game:*

1. *The adversary* $\mathcal{A}$ *is run on input of the security parameter* $\lambda \in \mathbb{N}$ *and chooses a tag* $\tau^\star$*. The challenger generates* $(pk, sk) \leftarrow \mathsf{Keygen}(\lambda)$ *and gives* pk *to* $\mathcal{A}$.
2. *On polynomially-many occasions, the adversary chooses a pair* (τ, C) *such that* $\tau \neq \tau^\star$ *and obtains* $M \leftarrow \mathsf{Decrypt}(sk, \tau, C)$*, where* M *may be* $\perp$ *if the pair* (τ, C) *is deemed invalid by the decryption algorithm.*
3. *The adversary* $\mathcal{A}$ *chooses two equal-length messages* M_0, M_1 *and obtains* $C^\star \leftarrow \mathsf{Encrypt}(pk, M_\beta, \tau^\star)$ *for a random bit* $\beta \xleftarrow{R} \{0,1\}$ *chosen by the challenger.*
4. *The adversary* $\mathcal{A}$ *makes further decryption queries as in step 2.*
5. $\mathcal{A}$ *outputs a random bit* $\beta' \in \{0,1\}$ *and wins if* $\beta' = \beta$.

As always, $\mathcal{A}$*'s advantage is defined to be* $\mathbf{Adv}^{\text{stag-cca}}(\lambda) := |\Pr[\beta' = \beta] - 1/2|$.

The above definition relaxes the original one [55] in that no decryption query is allowed with respect to the challenge tag $\tau^\star$ chosen at step 1.

C.5 Chameleon Hash Functions

A chameleon hash function [50] consists of a tuple of algorithms $\mathsf{CMH} = (\mathsf{CMKg}, \mathsf{CMhash}, \mathsf{CMswitch})$ which contains an algorithm CMKg that, given a security parameter λ, outputs a key pair $(hk, tk) \leftarrow \mathcal{G}(\lambda)$. The randomized hashing algorithm outputs $y = \mathsf{CMhash}(hk, m, r)$ given the public key hk, a message m and random coins $r \in \mathcal{R}_{hash}$. On input of messages m, m', random coins $r \in \mathcal{R}_{hash}$ and

the trapdoor key tk, the switching algorithm computes random coins $r' \leftarrow \mathsf{CMswitch}(tk, m, r, m')$ such that $\mathsf{CMhash}(hk, m, r) = \mathsf{CMhash}(hk, m', r')$. The collision-resistance property mandates that it be infeasible to come up with pairs $(m', r') \neq (m, r)$ such that $\mathsf{CMhash}(hk, m, r) = \mathsf{CMhash}(hk, m', r')$ without knowing the trapdoor key tk. Uniformity guarantees that the distribution of hash values is independent of the message m: in particular, for all hk, and all messages m, m', the distributions $\{r \leftarrow \mathcal{R}_{hash} : \mathsf{CMHash}(hk, m, r)\}$ and $\{r \leftarrow \mathcal{R}_{hash} : \mathsf{CMHash}(hk, m', r)\}$ are identical.

D Proof of Theorem 1

Proof. To prove the result, we will consider a sequence of hybrid games involving two kinds of private keys and two kinds of signatures.

Type A private keys: These keys have the same distribution as those produced by the real Extract algorithm. They are obtained by computing

$$\begin{array}{llll} K_1 = g^{\omega} \cdot (v^{\mathsf{ID}} \cdot w)^s, & K_2 = g^{s \cdot \mathsf{ID}}, & K_3 = g^s, & \\ K_4 = h^{s \cdot \mathsf{ID}} & K_5 = h^s, & K_6 = t^s, & \hat{K_7} = \hat{g}^{\mathsf{ID}}, \end{array} \tag{17}$$

for a randomly chosen $s \xleftarrow{R} \mathbb{Z}_p$, and

$$\begin{array}{ll} z = z_1^{\omega} \cdot (z_2^{\mathsf{ID}} \cdot z_3)^s, & z_d = z_4^s, \\ r = r_1^{\omega} \cdot (r_2^{\mathsf{ID}} \cdot r_3)^s, & r_d = r_4^s. \end{array}$$

We observe that, since the vector $(K_1, K_2, K_3, K_4, K_5, 1, 1, \Omega)$ (resp. $(K_6, 1, 1, 1, 1, K_3, K_5, 1)$) is in the subspace spanned by the first three rows (resp. the last row) of $\mathbf{M}$, the QA-NIZK proofs (z, r) and (z_d, r_d) can equivalently be computed as

$$\begin{array}{ll} z = \prod_{i=1}^{5} K_i^{-\chi_i} \cdot \Omega^{-\chi_8}, & z_d = K_6^{-\chi_1} \cdot K_3^{-\chi_6} \cdot K_5^{-\chi_7}, \\ r = \prod_{i=1}^{5} K_i^{-\gamma_i} \cdot \Omega^{-\gamma_8}, & r_d = K_6^{-\gamma_1} \cdot K_3^{-\gamma_6} \cdot K_5^{-\gamma_7}. \end{array} \tag{18}$$

We further define **Type A'** private keys as a broader class of private keys where only conditions (17) are required. We do not impose any condition on (z, r) and (z_d, r_d) besides being valid homomorphic signatures on the vectors $(K_1, K_2, K_3, K_4, K_5, 1, 1, \Omega)$ and $(K_6, 1, 1, 1, 1, K_3, K_5, 1)$, respectively. Type A private keys are thus a special kind of Type A' private keys.

Type B private keys: In these keys, the secret exponent $\omega \in \mathbb{Z}_p$ is replaced by a random exponent $\omega' \in_R \mathbb{Z}_p$. Type B keys are obtained by computing

$$\begin{array}{llll} K_1 = g^{\omega'} \cdot (v^{\mathsf{ID}} \cdot w)^s, & K_2 = g^{s \cdot \mathsf{ID}}, & K_3 = g^s, & \\ K_4 = h^{(s+s_1) \cdot \mathsf{ID}}, & K_5 = h^{s+s_1}, & K_6 = t^s, & \hat{K_7} = \hat{g}^{\mathsf{ID}}, \end{array}$$

where $\omega', s, s_1 \xleftarrow{R} \mathbb{Z}_p$, and

$$\begin{array}{ll} z = \prod_{i=1}^{5} K_i^{-\chi_i} \cdot \Omega^{-\chi_8}, & r = \prod_{i=1}^{5} K_i^{-\gamma_i} \cdot \Omega^{-\gamma_8}, \\ z_d = K_6^{-\chi_1} \cdot K_3^{-\chi_6} \cdot K_5^{-\chi_7}, & r_d = K_6^{-\gamma_1} \cdot K_3^{-\gamma_6} \cdot K_5^{-\gamma_7}. \end{array}$$

Likewise, we consider two distributions of signatures.

Type A signatures: These signatures are obtained by computing

$$\begin{aligned} \sigma_1 &= g^{\omega} \cdot (v^{\mathsf{ID}} \cdot t^{M} \cdot w)^{s}, & \sigma_2 &= g^{s \cdot \mathsf{ID}}, \\ \sigma_3 &= g^{s}, & \sigma_4 &= h^{s \cdot \mathsf{ID}}, \\ \sigma_5 &= h^{s}, & \hat{\sigma_6} &= \hat{g}^{\mathsf{ID}}, \end{aligned} \tag{19}$$

and

$$\tilde{z} = z_1^{\omega} \cdot (z_2^{\mathsf{ID}} \cdot z_4^{M} \cdot z_3)^{s}, \qquad \tilde{r} = r_1^{\omega} \cdot (r_2^{\mathsf{ID}} \cdot r_4^{M} \cdot r_3)^{s}.$$

Note that, since $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_3^M, \sigma_5^M, \Omega)$ is in the row space of $\mathbf{M}$, the QA-NIZK proof $(\tilde{z}, \tilde{r})$ has the same distribution as if it were computed as

$$\begin{aligned} \tilde{z} &= \prod_{i=1}^{5} \sigma_i^{-\chi_i} \cdot \sigma_3^{-M \cdot \chi_6} \cdot \sigma_5^{-M \cdot \chi_7} \cdot \Omega^{-\chi_8}, \\ \tilde{r} &= \prod_{i=1}^{5} \sigma_i^{-\gamma_i} \cdot \sigma_3^{-M \cdot \gamma_6} \cdot \sigma_5^{-M \cdot \gamma_7} \cdot \Omega^{-\gamma_8}. \end{aligned} \tag{20}$$

Moreover, Type A signatures also have the same distribution as signatures derived from a Type A key $K_{\mathsf{ID}} = (K_1, K_2, K_3, K_4, K_5, K_6, \hat{K_7}, z, r, z_d, r_d)$ by faithfully running the real signing algorithm.

Similarly to private keys, we define **Type A'** signatures as a generalization of Type A signatures where only conditions (19) are imposed and no restriction is placed on $(\tilde{z}, \tilde{r})$ beyond the fact that it should be a valid homomorphic signature on the vector (6).

Type B signatures: In these signatures, the secret exponent $\omega \in \mathbb{Z}_p$ is replaced by a random value $\omega' \in_R \mathbb{Z}_p$. Type B signatures are obtained by computing

$$\begin{aligned} \sigma_1 &= g^{\omega'} \cdot (v^{\mathsf{ID}} \cdot t^{M} \cdot w)^{s}, & \sigma_2 &= g^{s \cdot \mathsf{ID}}, \\ \sigma_3 &= g^{s}, & \sigma_4 &= h^{(s+s_1) \cdot \mathsf{ID}}, \\ \sigma_5 &= h^{s+s_1}, & \hat{\sigma_6} &= \hat{g}^{\mathsf{ID}}, \end{aligned}$$

for random $\omega', s, s_1 \xleftarrow{R} \mathbb{Z}_p$. The QA-NIZK proof $(\tilde{z}, \tilde{r})$ is computed using $\{(\chi_i, \gamma_i)\}_{i=1}^{8}$ in the same way as in (20).

We note that, unlike their Type A counterparts, Type B private keys and Type B signature can both be generated without using the private exponent $\omega \in \mathbb{Z}_p$.

We consider a sequence of games. In Game i, we denote by S_i the event that $\mathcal{A}$ wins by producing a signature $\sigma^{\star} = (\sigma_1^{\star}, \sigma_2^{\star}, \sigma_3^{\star}, \sigma_4^{\star}, \sigma_5^{\star}, \tilde{z}^{\star}, \tilde{r}^{\star}, \hat{\sigma_6}^{\star})$ on a message $M^{\star}$ such that $\mathsf{ID}^{\star} = \log_{\sigma_3^{\star}}(\sigma_2^{\star}) = \log_{\sigma_5^{\star}}(\sigma_4^{\star})$ was not queried for private key extraction and no signing query was made on $M^{\star}$ for the identity $\mathsf{ID}^{\star}$.

Game 0: This game is the real game.

Game 1: This game is identical to Game 0 with the sole difference that, instead of generating signatures $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \tilde{z}, \tilde{r}, \hat{\sigma_6})$ by deriving them from a Type A private key, the challenger $\mathcal{B}$ directly computes Type A signatures according to (19)-(20). At the end of the game, we define E_1 as the event that the forgery $\sigma^{\star}$ produced by the adversary $\mathcal{A}$ is a Type A' signature. We clearly have $\Pr[S_1] = \Pr[S_1 \wedge E_1] + \Pr[S_1 \wedge \neg E_1]$. Lemma 1 provides evidence that event $S_1 \wedge \neg E_1$ would contradict the computational soundness of the QA-NIZK arguments of [53] and thus the DDH assumption in $\hat{\mathbb{G}}$. We have $\Pr[S_1 \wedge \neg E_1] \leq \mathbf{Adv}_{\hat{\mathbb{G}}}^{\mathrm{DDH}}(\lambda) + 1/p$, meaning that, unless the DDH assumption is false in $\hat{\mathbb{G}}$, $\mathcal{A}$ cannot produce anything but a Type A' forgery. Our task is now to upper-bound $\Pr[S_1 \wedge E_1]$.

Game 2: This game is like Game 1 with the difference that, when the challenger $\mathcal{B}$ answers private key queries and signing queries, the QA-NIZK proofs (z, r), (z_d, r_d) and $(\tilde{z}, \tilde{r})$ are computed as simulated QA-NIZK proofs by using $\{(\chi_i, \gamma_i)\}_{i=1}^{8}$ as in (18) and (20). These QA-NIZK proofs are thus simulated proofs for true statements, so that their distribution remains unchanged. We have $\Pr[S_2 \wedge E_2] = \Pr[S_1 \wedge E_1]$, where E_2 denotes the event that $\mathcal{A}$ produces a Type A' forgery in Game 2.

We now consider a sub-sequence of q_1 hybrid games, where q_1 is the number of private key queries, where we gradually modify the distribution of private keys obtained by the adversary. For convenience, we define Game 3.0 to be identical to Game 2.

Game 3.k ($0 \leq k \leq q_1$): In Game 3.k, the challenger returns a Type B private key at the first k private key queries. At the last $q_1 - k$ private key queries, the challenger outputs a Type A private key. Lemma 2 shows that, in Game 3.k, the adversary $\mathcal{A}$ outputs a Type A' signature with about the same probability as in Game 3.$(k-1)$ as long as the DDH assumption holds in $\mathbb{G}$. Specifically, we have $|\Pr[S_{3.k} \wedge E_{3.k}] - \Pr[S_{3.(k-1)} \wedge E_{3.(k-1)}]| \leq \mathbf{Adv}_{\mathbb{G}}^{\mathrm{DDH}}(\lambda) + 1/p$.

The next step is to consider a sub-sequence of q_2 hybrid games where the distribution of signatures is gradually modified. For convenience, we define Game 4.0 as being identical to Game 3.q_1.

Game 4.k ($0 \leq k \leq q_2$): In Game 4.k, all private key queries are answered by returning a Type B private key. As for the distribution of signatures produced by the signing oracle, the challenger returns a Type B signature at the first k signing queries. The last $q_2 - k$ signing queries are answered by returning a Type A signature. Lemma 3 demonstrates that, if the DDH assumption holds in $\mathbb{G}$, gradually changing the distribution of signatures does not significantly increase $\mathcal{A}$'s probability not to output a Type A' forgery. We have

$$|\Pr[S_{4.k} \wedge E_{4.k}] - \Pr[S_{4.(k-1)} \wedge E_{4.(k-1)}]| \leq \mathbf{Adv}_{\mathbb{G}}^{\mathrm{DDH}}(\lambda) + 1/p.$$

In Game 4.q_2, we know that, unless the SXDH assumption is false, $\mathcal{A}$ can only output a Type A' forgery although it only obtains Type B private keys and Type B signatures during the game. However, Lemma 4 shows that event $S_{4.q_2} \wedge E_{4.q_2}$ would contradict the Computational Diffie-Hellman assumption which is trivially implied by DDH: we thus have $\Pr[S_{4.q_2} \wedge E_{4.q_2}] \leq \mathbf{Adv}_{\mathbb{G}}^{\mathrm{DDH}}(\lambda)$. Putting the above altogether, we can upper-bound the adversary's advantage as

$$\begin{aligned}\Pr[S_0] &\leq \mathbf{Adv}_{\hat{\mathbb{G}}}^{\mathrm{DDH}}(\lambda) + \frac{1}{p} + (q_1+q_2)\cdot\left(\mathbf{Adv}_{\mathbb{G}}^{\mathrm{DDH}}(\lambda) + \frac{1}{p}\right) + \mathbf{Adv}_{\mathbb{G}}^{\mathrm{DDH}}(\lambda)\\ &< (q_1+q_2+2)\cdot\left(\mathbf{Adv}_{\mathbb{G},\hat{\mathbb{G}}}^{\mathrm{SXDH}}(\lambda) + \frac{1}{p}\right).\end{aligned}$$

□

Lemma 1. *In Game* 1*, if the DDH assumption holds in* $\hat{\mathbb{G}}$*, no PPT adversary can output anything but a Type A' forgery.*

Proof. Let us assume that a PPT adversary $\mathcal{A}$ has non-negligible probability ϵ of outputting a forgery $\sigma^\star = (\sigma_1^\star, \sigma_2^\star, \sigma_3^\star, \sigma_4^\star, \sigma_5^\star, \tilde{z}^\star, \tilde{r}^\star, \hat{\sigma_6}^\star)$ which is not a Type A' forgery in Game 1. We turn $\mathcal{A}$ into an algorithm $\mathcal{B}$ that inputs an instance $(\hat{g_z}, \hat{g_r}) \in \hat{\mathbb{G}}^2$ of the DP problem and finds a non-trivial pair $(z^\star, r^\star) \in \mathbb{G}^2$ such that $e(z^\star, \hat{g_z}) \cdot e(r^\star, \hat{g_r}) = 1_{\mathbb{G}_T}$ with probability $\epsilon \cdot (1 - 1/p)$. In turn, $\mathcal{B}$ implies a successful DDH distinguisher in $\hat{\mathbb{G}}$ since the DP assumption implies the DDH assumption in $\hat{\mathbb{G}}$.

Let us first define the vector $\boldsymbol{\sigma} = (\sigma_1^\star, \sigma_2^\star, \sigma_3^\star, \sigma_4^\star, \sigma_5^\star, \sigma_3^{\star M^\star}, \sigma_5^{\star M^\star}, \Omega) \in \mathbb{G}^8$. We know that, if $(M^\star, \sigma^\star)$ is not a Type A' forgery, then $\boldsymbol{\sigma}$ is not in the row space of $\mathbf{M}$.

Our algorithm $\mathcal{B}$ receives as input a public key pk_{hsps} for an instance of the LHSPS scheme

allowing to sign vectors of dimension $n = 8$. Then, $\mathcal{B}$ runs Steps 1, 2 and 3 of the real key generation algorithm on its own to obtain $g, v, t, w \xleftarrow{R} \mathbb{G}$ and $h = g^a$, $\hat{g}$ and $\Omega = h^\omega$ for randomly chosen $\omega, a \xleftarrow{R} \mathbb{Z}_p$. It then queries its own LHSPS challenger to obtain signatures $\{(z_i, r_i)\}_{j=1}^4$ on the rows of the matrix (3). The adversary $\mathcal{A}$ is run on input of

$$\mathsf{mpk} = \Big(g,\ h,\ \hat{g},\ (t, v, w),\ \Omega = h^\omega,\ \mathsf{pk}_{hsps},\ \{(z_j, r_j)\}_{j=1}^4 \Big)$$

Since $\mathcal{B}$ knows the master secret key $\omega \in \mathbb{Z}_p$, it can answer all private key and signing queries by faithfully running the $\mathsf{Extract}$ and Sign algorithms. In particular, it does not need $\{(\chi_i, \gamma_i)\}_{i=1}^8$ for this purpose. When $\mathcal{A}$ terminates, it outputs a message $M^\star$ along with a signature $\sigma^\star = (\sigma_1^\star, \sigma_2^\star, \sigma_3^\star, \sigma_4^\star, \sigma_5^\star, \tilde{z}^\star, \tilde{r}^\star, \hat{\sigma_6}^\star)$ which does not constitute a Type A' forgery. This implies that $(\tilde{z}^\star, \tilde{r}^\star)$ is a valid homomorphic signature on the vector $\boldsymbol{\sigma} = (\sigma_1^\star, \sigma_2^\star, \sigma_3^\star, \sigma_4^\star, \sigma_5^\star, {\sigma_3^\star}^{M^\star}, {\sigma_5^\star}^{M^\star}, \Omega)$. but $\boldsymbol{\sigma}$ is outside the row space of $\mathbf{M}$. Consequently, $\mathcal{B}$ can output $(\tilde{z}^\star, \tilde{r}^\star)$ and the above vector as a valid forgery for the LHSPS scheme. The result of [52] ensures that $\mathcal{B}$ implies an algorithm solving the DP problem with probability at least $\epsilon \cdot (1 - 1/p)$. □

Note that the proof of Lemma 1 does not rely on any specific property of the QA-NIZK proof system of Libert *et al.* [53]. Moreover, the proof of Lemma 1 goes through if the reduction $\mathcal{B}$ knows the discrete logarithms $\log_g(h)$, $\log_g(t)$, $\log_g(v)$ and $\log_g(w)$. For this reason, the QA-NIZK argument of Jutla and Roy [44] can be used so as to replace the pair $(\tilde{z}, \tilde{r})$ by a single group element. In order to give a reduction from the soundness of the underlying QA-NIZK argument, $\mathcal{B}$ takes as input a CRS for the language defined by the subspace spanned by the rows of $\mathbf{M}$. Since all signing queries are answered by honestly generating QA-NIZK proofs, it comes that any successful forger also breaks the soundness of the QA-NIZK argument. Indeed, the adversary still has to come up with a convincing argument for a vector $\boldsymbol{\sigma}$ outside the row space of $\mathbf{M}$. Since Lemma 1 is the only step where we appeal to the soundness of the argument system, we observe that any other QA-NIZK argument can be applied.

Lemma 2. *If the DDH assumption holds in $\mathbb{G}$, the adversary $\mathcal{A}$ produces a Type A' forgery with nearly the same probabilities in Game 3.k and Game 3.$(k-1)$ for each $k \in \{1, \ldots, q_1\}$.*

Proof. Towards a contradiction, let us assume that there exists $k \in \{1, \ldots, q_1\}$ such that the adversary $\mathcal{A}$ outputs a Type A' forgery with significantly smaller probability in Game 3.k than in Game 3.$(k-1)$. We show how to build a DDH distinguisher in $\mathbb{G}$ on top of this adversary $\mathcal{A}$.

Our distinguisher $\mathcal{B}$ takes as input a tuple $(g, g^a, g^b, \eta) \in \mathbb{G}^3$, where $\eta = g^{a(b+c)}$, and has to decide if $c = 0$ or $c \in_R \mathbb{Z}_p$. To this end, $\mathcal{B}$ defines $h = g^a$. It also chooses $\omega, a_v, a_w, a_t, b_v, b_w \xleftarrow{R} \mathbb{Z}_p$, and sets $\Omega = g^\omega$ as well as

$$v = g^{a_v} \cdot h^{b_v}, \qquad w = g^{a_w} \cdot h^{b_w}, \qquad t = g^{a_t}.$$

It also generates $\mathsf{sk}_{hsps} = \{(\chi_i, \gamma_i)\}_{i=1}^8$, $\mathsf{pk}_{hsps} = (\hat{g_z}, \hat{g_r}, \{\hat{g_i}\}_{i=1}^8)$ as well as $\{(z_j, r_j)\}_{j=1}^4$ as in steps 4-5 of the real key generation algorithm and runs the adversary on input of

$$\mathsf{mpk} = \Big(g,\ h,\ \hat{g},\ (t, v, w),\ \Omega = h^\omega,\ \mathsf{pk}_{hsps},\ \{(z_j, r_j)\}_{j=1}^4 \Big)$$

and retains $(\omega, \{(\chi_i, \gamma_i)\}_{i=1}^8)$ to handle signing queries and private key queries.

During the game, $\mathcal{B}$ uses $(\omega, \{(\chi_i, \gamma_i)\}_{i=1}^8)$ to compute and return a Type A signature at each signing query. The treatment of private key queries depends on the index $j \in \{1, \ldots, q_1\}$ of the query.

Case $j < k$: $\mathcal{B}$ computes $K_{\mathsf{ID}} = (K_1, K_2, K_3, K_4, K_5, K_6, \hat{K_7}, z, r, z_d, r_d)$ as a Type B private key and computes (z, r, z_d, r_d) using $\{(\chi_i, \gamma_i)\}_{i=1}^8$.

Case $j > k$: The last $q_1 - k - 1$ private keys are computed as Type A private keys. Note that $\mathcal{B}$ is able to compute them since it knows the master secret key $\omega \in \mathbb{Z}_p$ and $\{(\chi_i, \gamma_i)\}_{i=1}^{8}$.

Case $j = k$: In the k-th private key, $\mathcal{B}$ will embed the DDH instance by defining.

$$\begin{aligned} K_2 &= (g^b)^{\mathsf{ID}}, & K_3 &= g^b \\ K_4 &= (\eta)^{\mathsf{ID}} & K_5 &= \eta \\ K_6 &= (g^b)^{a_t} & \hat{K_7} &= \hat{g}^{\mathsf{ID}} \end{aligned}$$

and

$$K_1 = g^{\omega} \cdot K_2^{a_v} \cdot K_3^{a_w} \cdot K_4^{b_v} \cdot K_5^{b_w}.$$

Then, $\mathcal{B}$ generates the QA-NIZK proofs (z, r) and (z_d, r_d) as per (18).

We note that, if $\eta = g^{ab}$, $(K_1, K_2, K_3, K_4, K_5, K_6, \hat{K_7}, z, r, z_d, r_d)$ is distributed as a Type A private key with $s = b$. In contrast, if $\eta = g^{a(b+c)}$ for some $c \in_R \mathbb{Z}_p$, we can write

$$\begin{aligned} K_2 &= g^{b \cdot \mathsf{ID}} & K_3 &= g^b \\ K_4 &= h^{(b+c) \cdot \mathsf{ID}} & K_5 &= h^{b+c} \\ K_6 &= t^b & \hat{K_7} &= \hat{g}^{\mathsf{ID}} \end{aligned}$$

$$K_1 = g^{\omega} \cdot g^{ac \cdot (b_v \cdot \mathsf{ID} + b_w)} \cdot (v^{\mathsf{ID}} \cdot w)^b = g^{\omega'} \cdot (v^{\mathsf{ID}} \cdot w)^c,$$

where $\omega' = \omega + ac \cdot (b_v \cdot \mathsf{ID} + b_w)$. Since the latter value is uniformly distributed and independent of $\mathcal{A}$'s view, $(K_1, K_2, K_3, K_4, K_5, K_6, \hat{K_7}, z, r, z_d, r_d)$ is distributed as a Type B private key if $\eta = g^{a(b+c)}$ with $c \in_R \mathbb{Z}_p$.

At the end of the game, the adversary $\mathcal{A}$ halts and outputs a message $M^\star$ together with a forgery $\sigma^\star = (\sigma_1^\star, \sigma_2^\star, \sigma_3^\star, \sigma_4^\star, \sigma_5^\star, \tilde{z}^\star, \tilde{r}^\star, \hat{\sigma_6}^\star)$ such that the implicit identity $\mathsf{ID}^\star = \log_{\hat{g}}(\hat{\sigma_6}) = \log_{\sigma_3^\star}(\sigma_2^\star) = \log_{\sigma_5^\star}(\sigma_4^\star)$ has not been submitted to the private key extraction oracle. At this point, $\mathcal{B}$ must decide if $\sigma^\star$ is a Type A' forgery. To this end, it checks if the equality

$$\sigma_1^\star = g^{\omega} \cdot \sigma_2^{\star a_v} \cdot \sigma_3^{\star a_w + a_t \cdot M^\star} \cdot \sigma_4^{\star b_v} \cdot \sigma_5^{\star b_w} \tag{21}$$

is satisfied. If so, $\mathcal{B}$ outputs 1, meaning that $\eta = g^{ab}$. Otherwise, it outputs 0 to indicate its belief that $\eta = g^{a(b+c)}$ for some uniformly random $c \in_R \mathbb{Z}_p$.

To justify why $\mathcal{B}$ is a SXDH distinguisher if $\mathcal{A}$ has non-negligible chance of outputting a forgery that is not of Type A', we first note that $\sigma^\star$ is of the form

$$\begin{aligned} \sigma_2^\star &= g^{s \cdot \mathsf{ID}^\star} & \sigma_3^\star &= g^s \\ \sigma_4^\star &= h^{(s+s_1) \cdot \mathsf{ID}^\star} & \sigma_5^\star &= h^{s+s_1} \\ \hat{\sigma_6^\star} &= \hat{g}^{\mathsf{ID}^\star} & \sigma_1^\star &= g^{\omega + s_0} \cdot (v^{\mathsf{ID}^\star} \cdot w)^s, \end{aligned}$$

for some $s, s_0, s_1 \in \mathbb{Z}_p$. In particular, the verification equations imply that $\sigma_4^\star$ and $\sigma_5^\star$ depend on the same $s_1 \in \mathbb{Z}_p$. The distinguisher $\mathcal{B}$ thus has to decide if $(s_0, s_1) = (0, 0)$, which means that $\sigma^\star$ is a Type A' signature, or $(s_0, s_1) \neq 0$. We first remark that, if $s_0 \neq 0$ and $s_1 = 0$, the equality (21) can never be satisfied. We thus focus on the case $s_1 \neq 0$ and observe that, in this case, the equality (21) holds if and only if $s_0 = a \cdot s_1 \cdot (b_v \cdot \mathsf{ID}^\star + b_w)$. We claim that this occurs with probability at most $1/p$. To see this, we remark that the information that $\mathcal{A}$ can learn about $(a_v, a_w, b_v, b_w) \in \mathbb{Z}_p^4$ during the entire game amounts to the first three rows of the right-hand-side member in the following linear system

$$\begin{pmatrix} 1 & & a & \\ & 1 & & a \\ & & ac \cdot \mathsf{ID} & ac \\ & & as_1 \cdot \mathsf{ID}^\star & as_1 \end{pmatrix} \cdot \begin{pmatrix} a_v \\ a_w \\ b_v \\ b_w \end{pmatrix} = \begin{pmatrix} \log_g(v) \\ \log_g(w) \\ \omega' - \omega \\ s_0 \end{pmatrix},$$

where ID is the identity involved in the k-th private key query. Hence, the adversary can only predict $as_1 \cdot (b_v \cdot \mathsf{ID}^\star + b_w)$ with probability $1/p$ since the above matrix is non-singular when $\mathsf{ID} \neq \mathsf{ID}^\star$. □

Lemma 3. *Under the DDH assumption in $\mathbb{G}$, the adversary $\mathcal{A}$ produces a Type A' forgery with negligibly different probabilities in Game 4.k and Game 4.$(k-1)$ for each $k \in \{1, \ldots, q_2\}$.*

Proof. Let us assume that there exists an index $k \in \{1, \ldots, q_2\}$ and an adversary $\mathcal{A}$ which outputs a Type A' forgery with noticeably smaller probability in Game 4.k than in Game 4.$(k-1)$. We construct a SXDH distinguisher $\mathcal{B}$ using $\mathcal{A}$.

Our distinguisher $\mathcal{B}$ takes as input a SXDH instance $(g, g^a, g^b, \eta) \in \mathbb{G}^3$, where $\eta = g^{a(b+c)}$, and undertakes to decide if $c = 0$ or $c \in_R \mathbb{Z}_p$. To do this, $\mathcal{B}$ sets $h = g^a$. It also picks $\omega, a_v, a_w, a_t, b_v, b_w, b_t \xleftarrow{R} \mathbb{Z}_p$, and defines $\Omega = g^\omega$ as well as

$$v = g^{a_v} \cdot g^{b_v}, \qquad w = g^{a_w} \cdot h^{b_w} \qquad t = g^{a_t} \cdot h^{b_t}$$

It also chooses $\mathsf{sk}_{hsps} = \{(\chi_i, \gamma_i)\}_{i=1}^8$ and computes $\mathsf{pk}_{hsps} = (\hat{g}_z, \hat{g}_r, \{\hat{g}_i\}_{i=1}^8)$ and $\{(z_j, r_j)\}_{j=1}^4$ in the same way as in steps 4-5 of the actual key generation algorithm. The adversary is fed with

$$\mathsf{mpk} = \Big(g,\ h,\ \hat{g},\ (t, v, w),\ \Omega = h^\omega,\ \mathsf{pk}_{hsps},\ \{(z_j, r_j)\}_{j=1}^4 \Big)$$

and $\mathcal{B}$ retains $(\omega, \{(\chi_i, \gamma_i)\}_{i=1}^8)$ in order to handle adversarial queries.

Throughout the game, $\mathcal{B}$ uses $(\omega, \{(\chi_i, \gamma_i)\}_{i=1}^8)$ to output Type B private keys at each private key query. The way to answer signing queries depends on the index $j \in \{1, \ldots, q_2\}$ of those queries.

Case $j < k$: $\mathcal{B}$ computes a Type B signature $\sigma = \big(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \tilde{z}, \tilde{r}, \hat{\sigma_6}\big)$. In particular, it computes $(\tilde{z}, \tilde{r})$ as a simulated QA-NIZK proof using $\{(\chi_i, \gamma_i)\}_{i=1}^8$.

Case $j > k$: The last $q_2 - k - 1$ signing queries are computed as Type A signatures, which $\mathcal{B}$ is able to generate since it knows the master secret key $\omega \in \mathbb{Z}_p$ and $\{(\chi_i, \gamma_i)\}_{i=1}^8$.

Case $j = k$: In the k-th signing query (ID, M), $\mathcal{B}$ embeds the DDH instance in the signature and simulates either Game 4.k or Game 4.$(k-1)$ depending on whether $\eta = g^{ab}$ or $\eta = g^{a(b+c)}$ for some random $c \in_R \mathbb{Z}_p$. Namely, $\mathcal{B}$ computes

$$\sigma_2 = (g^b)^{\mathsf{ID}}, \qquad \sigma_3 = g^b$$
$$\sigma_4 = \eta^{\mathsf{ID}} \qquad \sigma_5 = \eta \qquad \hat{\sigma_6} = \hat{g}^{\mathsf{ID}}$$

and

$$\sigma_1 = g^\omega \cdot \sigma_2^{a_v} \cdot \sigma_3^{a_w + a_t \cdot M} \cdot \sigma_4^{b_v} \cdot \sigma_5^{b_w + b_t \cdot M}$$

Then, $\mathcal{B}$ generates the QA-NIZK proof $(\tilde{z}, \tilde{r})$ as per (20).

We observe that, if $\eta = g^{ab}$, $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \tilde{z}, \tilde{r}, \hat{\sigma_6})$ is distributed as a Type A signature with $s = b$. If $\eta = g^{a(b+c)}$ for some uniform $c \in_R \mathbb{Z}_p$, we can write

$$\sigma_2 = g^{b \cdot \mathsf{ID}} \qquad \sigma_3 = g^b$$
$$\sigma_4 = h^{(b+c) \cdot \mathsf{ID}} \qquad \sigma_5 = h^{b+c} \qquad \hat{\sigma_6} = \hat{g}^{\mathsf{ID}}$$

$$\sigma_1 = g^\omega \cdot g^{ac \cdot (b_v \cdot \mathsf{ID} + b_t \cdot M + b_w)} \cdot (v^{\mathsf{ID}} \cdot t^M \cdot w)^b = g^{\omega'} \cdot (v^{\mathsf{ID}} \cdot t^M \cdot w)^b,$$

where $\omega' = \omega + ac \cdot (b_v \cdot \mathsf{ID} + b_t \cdot M + b_w)$. Since the term $(b_v \cdot \mathsf{ID} + b_t \cdot M + b_w)$ is uniformly distributed and independent of $\mathcal{A}$'s view, σ is distributed as a Type B signature if $\eta = g^{a(b+c)}$.

When $\mathcal{A}$ terminates, it outputs a message $M^\star$ and a forgery $\sigma^\star = (\sigma_1^\star, \sigma_2^\star, \sigma_3^\star, \sigma_4^\star, \sigma_5^\star, \tilde{z}^\star, \tilde{r}^\star, \hat{\sigma_6}^\star)$

such that it did not submit the identity $\mathsf{ID}^\star = \log_{\hat{g}}(\hat{\sigma_6}) = \log_{\sigma_3^\star}(\sigma_2^\star) = \log_{\sigma_5^\star}(\sigma_4^\star)$ to the private key extraction oracle or the pair $(\mathsf{ID}^\star, M^\star)$ to the signing oracle. At this point, $\mathcal{B}$ has to figure out if $\sigma^\star$ is a Type A' forgery or not. To this end, it tests whether the equality

$$\sigma_1^\star = g^\omega \cdot \sigma_2^{\star a_v} \cdot \sigma_3^{\star a_w + a_t \cdot M^\star} \cdot \sigma_4^{\star b_v} \cdot \sigma_5^{\star b_w + b_t \cdot M^\star} \tag{22}$$

is satisfied. If it is, $\mathcal{B}$ outputs 1 to indicate that $\eta = g^{ab}$. Otherwise, it outputs 0 and rather bets that $\eta = g^{a(b+c)}$ for some random $c \in_R \mathbb{Z}_p$.

To see why this test allows $\mathcal{B}$ to tell apart Type A' forgeries from other forgeries, we first remark that $\sigma^\star$ is necessarily of the form

$$\begin{array}{lll} \sigma_2^\star = g^{s \cdot \mathsf{ID}^\star} & \sigma_3^\star = g^s & \hat{\sigma_6}^\star = \hat{g}^{\mathsf{ID}^\star} \\ \sigma_4^\star = h^{(s+s_1)\cdot \mathsf{ID}^\star} & \sigma_5^\star = h^{s+s_1} & \sigma_1^\star = g^{\omega + s_0} \cdot (v^{\mathsf{ID}^\star} \cdot t^{M^\star} \cdot w)^s, \end{array}$$

for some $s, s_0, s_1 \in \mathbb{Z}_p$ since the verification equations ensure that $\sigma_4^\star$ and $\sigma_5^\star$ depend on the same $s_1 \in \mathbb{Z}_p$. The task of $\mathcal{B}$ thus amounts to deciding if $(s_0, s_1) = (0, 0)$ or $(s_0, s_1) \neq 0$. First, we note that the equality (22) can never be satisfied if $s_0 \neq 0$ and $s_1 = 0$. We are thus left with the case $s_1 \neq 0$ where the equality (22) only holds when $s_0 = a \cdot s_1 \cdot (b_v \cdot \mathsf{ID}^\star + b_t \cdot M^\star + b_w)$. We claim that this can only happen with probability $1/p$. Indeed, the information that $\mathcal{A}$ can infer about $(a_v, a_w, a_t, b_v, b_w, b_t) \in \mathbb{Z}_p^6$ during the game amounts to the first four rows of the right-hand-side member in the following linear system

$$\begin{pmatrix} 1 & & & a & & \\ & 1 & & & a & \\ & & 1 & & & a \\ & & & ac \cdot \mathsf{ID} & ac & ac \cdot M \\ & & & as_1 \cdot \mathsf{ID}^\star & as_1 & as_1 \cdot M^\star \end{pmatrix} \cdot \begin{pmatrix} a_v \\ a_w \\ a_t \\ b_v \\ b_w \\ b_t \end{pmatrix} = \begin{pmatrix} \log_g(v) \\ \log_g(w) \\ \log_g(t) \\ \omega' - \omega \\ s_0 \end{pmatrix},$$

where (ID, M) is the identity-message pair involved in the k-th signing query. We find that $\mathcal{A}$ can only predict $a \cdot s_1 \cdot (b_v \cdot \mathsf{ID}^\star + b_t \cdot M^\star + b_w)$ with probability $1/p$ since the above matrix has full rank whenever $(\mathsf{ID}, M) \neq (\mathsf{ID}^\star, M^\star)$. □

Lemma 4. *In Game $4.q_2$, any PPT adversary outputting a Type A' forgery contradicts the DDH assumption in $\mathbb{G}$. We have* $\Pr[S_{4.q_2} \wedge E_{4.q_2}] \leq \mathbf{Adv}_{\mathbb{G}}^{\mathrm{DDH}}(\lambda)$

Proof. The proof is straightforward and builds an algorithm $\mathcal{B}$ for solving the Computational Diffie-Hellman problem which is at least as hard as the DDH problem. Algorithm $\mathcal{B}$ takes as input a tuple $(g, h, \Omega = h^\omega)$ and computes g^ω. To generate mpk, $\mathcal{B}$ picks $\hat{g} \xleftarrow{R} \hat{\mathbb{G}}$, $a_v, a_w, a_t \xleftarrow{R} \mathbb{Z}_p$ and computes

$$v = g^{a_v}, \qquad w = g^{a_w} \qquad t = g^{a_t}.$$

Next, $\mathcal{B}$ generates $\mathsf{sk}_{hsps} = \{(\chi_i, \gamma_i)\}_{i=1}^8$, $\mathsf{pk}_{hsps} = (\hat{g_z}, \hat{g_r}, \{\hat{g}_i\}_{i=1}^8)$ and $\{(z_j, r_j)\}_{j=1}^4$ as in steps 4-5 of the real key generation algorithm and runs the adversary on input of

$$\mathsf{mpk} = \Big(g,\ h,\ \hat{g},\ (t, v, w),\ \Omega = h^\omega,\ \mathsf{pk}_{hsps},\ \{(z_j, r_j)\}_{j=1}^4 \Big).$$

It also retains $\mathsf{sk}_{hsps} = \{(\chi_i, \gamma_i)\}_{i=1}^8$ to handle signing queries and private key queries. During the game, all private key queries and signing queries are answered by returning Type B private keys and Type B signatures, respectively. Using $\mathsf{sk}_{hsps} = \{(\chi_i, \gamma_i)\}_{i=1}^8$, $\mathcal{B}$ can thus answer all queries without knowing the exponent $\omega = \log_h(\Omega)$ which is part of the problem instance.

The results of Lemmas 2 and 3 imply that, although $\mathcal{A}$ only obtains Type B private keys and

Type B signatures, it will necessarily output a Type A' forgery $\sigma^\star = (\sigma_1^\star, \sigma_2^\star, \sigma_3^\star, \sigma_4^\star, \sigma_5^\star, \tilde{z}^\star, \tilde{r}^\star, \hat{\sigma_6}^\star)$ unless the SXDH assumption is false. However, this event allows $\mathcal{B}$ to compute the sought-after solution

$$g^\omega = \sigma_1^\star \cdot \sigma_2^{\star -a_v} \cdot \sigma_3^{\star -a_w - a_t \cdot M^\star},$$

which *a fortiori* contradicts the DDH assumption in $\mathbb{G}$. □

E Proof of Theorem 2

Proof. We show that an F-unforgeability adversary $\mathcal{A}$ implies an equally successful forger $\mathcal{B}$ against the 2-level hierarchical signature.

The reduction $\mathcal{B}$ receives as input a master public key

$$\mathsf{mpk} = \Big(g,\ h,\ \hat{g},\ (t, v, w),\ \Omega = h^\omega,\ \mathsf{pk}_{hsps} = \big(\hat{g_z}, \hat{g_r}, \{\hat{g_i}\}_{i=1}^8\big),\ \{(z_j, r_j)\}_{j=1}^4\Big)$$

for the hierarchical signature. From mpk, $\mathcal{B}$ constructs a public key pk for the F-unforgeable signature by defining the underlying homomorphic signature public key $\mathsf{pk}'_{hsps} = \big(\hat{g_z}, \hat{g_r}, \{\hat{g_i}'\}_{i=1}^6\big)$ where

$$(\hat{g_1}', \hat{g_2}', \hat{g_3}', \hat{g_4}', \hat{g_5}', \hat{g_6}') = (\hat{g_1}, \hat{g_2}, \hat{g_3}, \hat{g_4}, \hat{g_5}, \hat{g_8}).$$

The F-unforgeability adversary $\mathcal{A}$ is fed with the public key

$$\mathsf{pk} = \Big(g,\ h,\ \hat{g},\ (v, w),\ \Omega = h^\omega,\ \mathsf{pk}'_{hsps},\ \{(z_j, r_j)\}_{j=1}^3\Big).$$

When $\mathcal{A}$ queries a signature on a message M, $\mathcal{B}$ asks its own challenger for a private keys associated with the identity $\mathsf{ID} = M$ and obtains a private key $K_{\mathsf{ID}} = (K_1, K_2, K_3, K_4, K_5, K_6, \hat{K_7}, z, r, z_d, r_d)$ where the first 7 components are of the form

$$K_1 = g^\omega \cdot (v^M \cdot w)^s, \qquad K_2 = g^{s \cdot M}, \qquad K_3 = g^s$$
$$K_4 = h^{s \cdot M} \qquad K_5 = h^s \qquad K_6 = t^s, \qquad \hat{K_7} = \hat{g}^M.$$

To answer $\mathcal{A}$'s query, $\mathcal{B}$ simply discards $(\hat{K_7}, z_d, r_d)$ and returns the tuple $\sigma = (K_1, K_2, K_3, K_4, K_5, z, r)$.

At the end of the game, $\mathcal{A}$ outputs a signature $\sigma^\star = \big(\sigma_1^\star, \sigma_2^\star, \sigma_3^\star, \sigma_4^\star, \sigma_5^\star, z^\star, r^\star\big)$ together with $F(M^\star) = \hat{g}^{M^\star}$, which satisfy the equalities $e(\sigma_2^\star, \hat{g}) = e(\sigma_3^\star, F(M^\star))$, $e(\sigma_4^\star, \hat{g}) = e(\sigma_5^\star, F(M^\star))$ and

$$e(z^\star, \hat{g_z}) \cdot e(r^\star, \hat{g_r}) = e(\sigma_1^\star, \hat{g_1})^{-1} \cdot e(\sigma_2^\star, \hat{g_2})^{-1} \cdot e(\sigma_3^\star, \hat{g_3})^{-1} \cdot e(\sigma_4^\star, \hat{g_4})^{-1} \cdot e(\sigma_5^\star, \hat{g_5})^{-1} \cdot e(\Omega, \hat{g_8})^{-1}.$$

This allows $\mathcal{B}$ to defeat the security of the two-level hierarchical signature by producing the signature $\sigma^\dagger = \big(\sigma_1^\star, \sigma_2^\star, \sigma_3^\star, \sigma_4^\star, \sigma_5^\star, z^\star, r^\star, \hat{\sigma_6}^\star\big)$, where $\hat{\sigma_6}^\star = F(M^\star)$, for the hierarchical message $(M^\star, 0)$. (Recall that a hierarchical signature adversary is only required to output the second-level message, which is 0 here, since the first level message $M^\star$ appears implicitly in the signature). Since $\mathcal{B}$ did not query the private key for the identity $\mathsf{ID}^\star = M^\star$ at any time, $(\sigma^\dagger, 0)$ is easily seen to be a valid forgery. □

F Proof of Theorem 3

To prove the result, it is convenient to use the following assumption which is implied by the XDLIN_2 assumption.

Definition 6. *The* **eXternal Simultaneous Double Pairing problem** *(XSDP) in* $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ *is, given two 4-uples of group elements* $(g_z, g_r, h_z, h_u) \in \mathbb{G}^4$ *and* $(\hat{g_z}, \hat{g_r}, \hat{h_z}, \hat{h_u}) \in \hat{\mathbb{G}}^4$ *that satisfy the conditions* $\log_{\hat{g_z}}(\hat{g_r}) = \log_{g_z}(g_r)$, $\log_{\hat{g_z}}(\hat{h_z}) = \log_{g_z}(h_z)$ *and* $\log_{\hat{g_z}}(\hat{h_u}) = \log_{g_z}(h_u)$, *to find a non-trivial triple* $(z, r, u) \in \mathbb{G}^3 \backslash \{(1_{\mathbb{G}}, 1_{\mathbb{G}}, 1_{\mathbb{G}})\}$ *such that*

$$e(z, \hat{g_z}) \cdot e(r, \hat{g_r}) = 1_{\mathbb{G}_T}, \qquad e(z, \hat{h_z}) \cdot e(u, \hat{h_u}) = 1_{\mathbb{G}_T}.$$

The **eXternal Simultaneous Double Pairing assumption** *posits the intractability of the XSDP problem for any PPT algorithm.*

Any algorithm solving XSDP yields an XDLIN_2 distinguisher. From an XDLIN_2 instance, we can create an XSDP instance $(g^{ac}, g^a, g^{bd}, g^b, \hat{g}^{ac}, \hat{g}^a, \hat{g}^{bd}, \hat{g}^b)$. If the XSDP solver finds a non-trivial triple (z, r, u) such that $e(z, \hat{g}^{ac}) \cdot e(r, \hat{g}^a) = 1_{\mathbb{G}_T}$ and $e(z, \hat{g}^{bd}) \cdot e(u, \hat{g}^b) = 1_{\mathbb{G}_T}$, we know that $r = z^{-c}$ and $u = z^{-d}$. To decide if $\eta = \hat{g}^{c+d}$, it is sufficient to check whether $e(r \cdot u, \hat{g}) \cdot e(z, \eta) = 1_{\mathbb{G}_T}$.

Proof. The proof considers two kinds of forgeries $\left(\sigma_1^\star, \sigma_2^\star, \sigma_3^\star, \sigma_4^\star, \sigma_5^\star, \hat{\sigma_6}^\star, z^\star, r^\star, Z^\star, R^\star, U^\star\right)$.

Type I forgeries: are such that $\hat{\sigma_6}^\star$ did not appear in any signature produced by the signing oracle.
Type II forgeries: are those for which $\hat{\sigma_6}^\star$ is recycled from a signature produced by the signing oracle.

Type I forgeries immediately imply an chosen-message adversary against the F-unforgeable signature of Section 3, which contradicts the SXDH assumption. Moreover, this remains true when the signing oracle outputs randomization tokens $(g^\tau, h^\tau, v^\tau, z_2^\tau, r_2^\tau)$ at each query: indeed, in the game modeling the security of the F-unforgeable signature, τ is always chosen by the F-unforgeability adversary. When we reduce the F-unforgeability of the signature scheme of Section 3 to a Type I SPS forger, the reduction can thus always reveal $(g^\tau, h^\tau, v^\tau, z_2^\tau, r_2^\tau)$ to the Type I forger against the SPS scheme. In the following, we thus focus on Type II adversaries.

Assuming the existence of a Type II forger, we construct an algorithm $\mathcal{B}$ that breaks the XSDP assumption, which in turn contradicts the XDLIN_2 assumption. Algorithm $\mathcal{B}$ takes as input an instance $(G_z, G_r, H_z, H_u, \hat{G}_z, \hat{G}_r, \hat{H}_r, \hat{H}_u)$ of the XSDP assumption with the task of finding a non-trivial (Z, R, U) such that $e(Z, \hat{G}_z) \cdot e(R, \hat{G}_r) = 1$ and $e(Z, \hat{H}_z) \cdot e(U, \hat{H_U}) = 1$. Recall that, in the definition of the assumption, we have $\log_{\hat{G_z}}(\hat{G_r}) = \log_{G_z}(G_r)$, $\log_{\hat{G_z}}(\hat{H_z}) = \log_{G_z}(H_z)$ and $\log_{\hat{G_z}}(\hat{H_u}) = \log_{G_z}(H_u)$.

To construct the public key, $\mathcal{B}$ chooses $w_t \xleftarrow{R} \mathbb{Z}_p$ and defines $\hat{G_t} = \hat{H_u}^{w_t}$, $G_t = H_u^{w_t}$. For each $i \in \{1, \ldots, n\}$, it also picks $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ and sets

$$G_i = G_z^{\chi_i} \cdot G_r^{\gamma_i}, \qquad \hat{G_i} = \hat{G_z}^{\chi_i} \cdot \hat{G_r}^{\gamma_i}, \qquad H_i = H_z^{\chi_i} \cdot H_u^{\delta_i}, \qquad \hat{H_i} = \hat{H_z}^{\chi_i} \cdot \hat{H_u}^{\delta_i}.$$

At step a.1 of the key generation algorithm, it also defines $g = G_z^{1/w_z}$, $\hat{g} = \hat{G_z}^{1/w_z}$ for a randomly chosen $w_z \xleftarrow{R} \mathbb{Z}_p^*$. It also sets $h = g^a$, $\Omega = g^\omega$, $v = g^{\alpha_v}$, $w = g^{\alpha_w}$ for randomly drawn $a, \omega, \alpha_v, \alpha_w \xleftarrow{R} \mathbb{Z}_p$. Then, it faithfully conducts steps a.2 to a.4 of the real key generation phase in such a way that it knows both $\mathsf{sk}_{fsig} = \omega$, $\mathsf{sk}_{hsps} = \{(\chi_{0,i}, \gamma_{0,i})\}_{i=1}^6$ as well as $\mathsf{sk}_{pots} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$. The adversary $\mathcal{A}$ is run on input of the public key

$$PK = \Big(g,\ h,\ \hat{g},\ (v, w),\ \Omega = h^\omega,\ \mathsf{pk}_{pots},\ \mathsf{pk}_{hsps},\ \{(z_j, r_j)\}_{j=1}^3 \Big).$$

During the attack game, signing queries are answered as follows. At each query $\boldsymbol{M} = (M_1, \ldots, M_n)$, $\mathcal{B}$ defines the $\hat{\sigma_6}$ component of the signature as

$$\hat{\sigma_6} = (\hat{G_z}^{\zeta} \cdot \hat{G_r}^{\rho})^{1/w_t},$$

for randomly chosen $\zeta, \rho \xleftarrow{R} \mathbb{Z}_p$. It also computes a corresponding "shadow" element in $\mathbb{G}$ as

$$\sigma_6 = (G_z{}^{\zeta} \cdot G_r{}^{\rho})^{1/w_t}.$$

Note that, by construction, we have $\log_g(\sigma_6) = \log_{\hat{g}}(\hat{\sigma_6})$. Then, $\mathcal{B}$ computes

$$Z = H_u^{\zeta} \cdot \prod_{i=1}^{n} M_i^{-\chi_i}, \qquad R = H_u^{\rho} \cdot \prod_{i=1}^{n} M_i^{-\gamma_i}, \qquad U = H_z^{-\zeta} \cdot \prod_{i=1}^{n} M_i^{-\delta_i}. \tag{23}$$

It also chooses $s \xleftarrow{R} \mathbb{Z}_p$ and computes

$$\begin{aligned} \sigma_2 &= \sigma_6^s, & \sigma_3 &= g^s \\ \sigma_4 &= \sigma_6^{a \cdot s}, & \sigma_5 &= g^{a \cdot s} \end{aligned}$$

and $\sigma_1 = g^{\omega} \cdot \sigma_2^{\alpha_v} \cdot \sigma_3^{\alpha_w}$. Next, it uses $\mathsf{sk}_{hsps} = \{(\chi_{0,i}, \gamma_{0,i})\}_{i=1}^{6}$ to compute a linearly homomorphic signature (z, r) on the vector $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \Omega)$ by computing

$$z = \prod_{i=1}^{5} \sigma_i^{-\chi_{0,i}} \cdot \Omega^{-\chi_{0,6}}, \qquad r = \prod_{i=1}^{5} \sigma_i^{-\gamma_{0,i}} \cdot \Omega^{-\gamma_{0,6}}.$$

We observe that, if we define $\tau = \frac{1}{w_t} \cdot (\zeta \cdot \log_{\hat{g}}(\hat{G_z}) + \rho \cdot \log_{\hat{g}}(\hat{G_r}))$, $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \hat{\sigma_6})$ can be written

$$(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \hat{\sigma_6}) = \left(g^{\omega} \cdot (v^{\tau} \cdot w)^s,\ g^{\tau \cdot s},\ g^s,\ h^{\tau \cdot s},\ h^s,\ \hat{g}^{\tau}\right),$$

so that the vector $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \Omega)$ lives in the row space of the matrix $\mathbf{M}$ in (9). Consequently, the pair (z, r) has the same distribution as if it had been derived from public key components $\{(z_j, r_j)\}_{j=1}^{6}$ using the coefficients $(\omega, s \cdot \tau, s)$. Finally, the pair (Z, R) also matches the prescribed distribution since it satisfies the verification equations (12). Indeed, we have

$$\begin{aligned} e(Z, \hat{G_z}) \ \cdot\ & e(R, \hat{G_r}) \cdot \prod_{i=1}^{n} e(M_i, \hat{G_i}) \\ &= e(H_u^{\zeta}, \hat{G_z}) \cdot e(H_u^{\rho}, \hat{G_r}) \\ &\qquad \cdot e(\prod_{i=1}^{n} M_i^{-\chi_i}, \hat{G_z}) \cdot e(\prod_{i=1}^{n} M_i^{-\gamma_i}, \hat{G_r}) \cdot \prod_{i=1}^{n} e(M_i, \hat{G_z}{}^{\chi_i} \cdot \hat{G_r}{}^{\gamma_i}) \\ &= e(H_u, \hat{G_z}{}^{\zeta} \cdot \hat{G_r}{}^{\rho}) = e(H_u^{w_t}, (\hat{G_z}{}^{\zeta} \cdot \hat{G_r}{}^{\rho})^{1/w_t}) = e(G_t, \hat{\sigma_6}) \end{aligned}$$

and

$$\begin{aligned} e(Z, \hat{H_z}) \ \cdot\ & e(U, \hat{H_u}) \cdot \prod_{i=1}^{n} e(M_i, \hat{H_i}) \\ &= e(H_u^{\zeta}, \hat{H_z}) \cdot e(H_z^{-\zeta}, \hat{H_u}) \\ &\qquad \cdot e(\prod_{i=1}^{n} M_i^{-\chi_i}, \hat{H_z}) \cdot e(\prod_{i=1}^{n} M_i^{-\delta_i}, \hat{H_u}) \cdot \prod_{i=1}^{n} e(M_i, \hat{H_z}{}^{\chi_i} \cdot \hat{H_u}{}^{\delta_i}) = 1. \end{aligned}$$

The signature $\left(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \hat{\sigma_6}, z, r, Z, R, U\right)$ is thus returned as the output of the $j^{\star}$-th signing query. Moreover, since $\mathcal{B}$ knows $a = \log_g(h)$ and $\alpha_v = \log_g(v)$, it can also compute a randomization token $(g^{\tau}, h^{\tau}, v^{\tau}, z_2^{\tau}, r_2^{\tau})$ by setting

$$(g^{\tau}, h^{\tau}, v^{\tau}) = \left(\sigma_6, \sigma_6^a, \sigma_6^{\alpha_v}\right)$$

and computing $(z_2^\tau, r_2^\tau) = \big((\sigma_6^{\alpha_v})^{-\chi_{0,1}} \cdot \sigma_6^{-\chi_{0,2}} \cdot (\sigma_6^a)^{-\chi_{0,4}}, (\sigma_6^{\alpha_v})^{-\gamma_{0,1}} \cdot \sigma_6^{-\gamma_{0,2}} \cdot (\sigma_6^a)^{-\gamma_{0,4}}\big)$ using sk_{hsps}. The latter pair indeed equals (z_2^τ, r_2^τ) since it is obtained as a homomorphic signature on the vector $(v^\tau, g^\tau, 1, h^\tau, 1, 1) \in \mathbb{G}^4$ obtained by raising the second row of $\mathbf{M}$ in (9) to the power $\tau \in \mathbb{Z}_p$.

By hypothesis, $\mathcal{A}$ eventually outputs a Type II forgery

$$\sigma^\star = \big(\sigma_1^\star, \sigma_2^\star, \sigma_3^\star, \sigma_4^\star, \sigma_5^\star, \hat{\sigma_6}^\star, z^\star, r^\star, Z^\star, R^\star, U^\star\big)$$

and we denote by $j^\star$-th the index of the query that $\hat{\sigma_6}^\star$ was recycled from. Let $\boldsymbol{M}_{j^\star} = (M_{j^\star,1}, \ldots, M_{j^\star,n})$ and $(Z^{(j^\star)}, R^{(j^\star)}, U^{(j^\star)})$ denote the message and the signature components of the $j^\star$-th signing query and let $\boldsymbol{M}^\star = (M_1^\star, \ldots, M_n^\star)$ and $(Z^\star, R^\star, U^\star)$ be those of the forgery. We know that

$$e(Z^{(j^\star)}/Z^\star, \hat{G}_z) \cdot e(R^{(j^\star)}/R^\star, \hat{G}_r) \cdot \prod_{i=1}^n e(M_{j^\star,i}/M_i^\star, \hat{G}_i) = 1$$
$$e(Z^{(j^\star)}/Z^\star, \hat{H}_z) \cdot e(U^{(j^\star)}/U^\star, \hat{H}_r) \cdot \prod_{i=1}^n e(M_{j^\star,i}/M_i^\star, \hat{H}_i) = 1,$$

so that, if we define the triple $(Z^\dagger, R^\dagger, U^\dagger)$ as

$$Z^\dagger = \frac{Z^{(j^\star)}}{Z^\star} \cdot \prod_{i=1}^n \Big(\frac{M_{j^\star,i}}{M_i^\star}\Big)^{-\chi_i}, \qquad R^\dagger = \frac{R^{(j^\star)}}{R^\star} \cdot \prod_{i=1}^n \Big(\frac{M_{j^\star,i}}{M_i^\star}\Big)^{-\gamma_i},$$

and $U^\dagger = \frac{U^{(j^\star)}}{U^\star} \cdot \prod_{i=1}^n \big(\frac{M_{j^\star,i}}{M_i^\star}\big)^{-\delta_i}$, it necessarily satisfies $e(Z^\dagger, \hat{G}_z) \cdot e(R^\dagger, \hat{G}_r) = 1$ and $e(Z^\dagger, \hat{H}_z) \cdot e(U^\dagger, \hat{H}_U) = 1$. The same arguments as in [2] show that $Z^\dagger = 1_\mathbb{G}$ with probability at most $1/p$.

Namely, let us consider what $\mathcal{A}$ learns about

$$(\chi_1, \ldots, \chi_n, \gamma_1, \ldots, \gamma_n, \delta_1, \ldots, \delta_n, \zeta_1, \ldots, \zeta_q, \rho_1, \ldots, \rho_q)$$

in the entire game if (ζ_j, ρ_j) denote the random coins of the j-th signing query for each $j \in \{1, \ldots, q\}$. The discrete logarithms $\{(\log_{\hat{g}} \hat{G}_i, \log_{\hat{g}} \hat{H}_i)\}_{i=1}^n$ provide $\mathcal{A}$ with $2n$ linear equations and $\{(\log_g G_i, \log_g H_i)\}_{i=1}^n$ only reveal redundant information. In each signature, $\hat{\sigma_6}$ and (Z, R, U) only provide 2 new independent linear equations since (R, U) are uniquely determined by $(\hat{\sigma_6}, Z)$ and the message $\boldsymbol{M} \in \mathbb{G}^n$. If we recap what an unbounded adversary $\mathcal{A}$ can see about the $3n + 2q$ unknowns $\boldsymbol{\chi} = (\chi_1, \ldots, \chi_n)$, $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)$, $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_n)$, $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_q)$, $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_q)$, it amounts to the right-hand-side member of the following system of $2n + 2q$ equations.

$$\begin{pmatrix} w_z \cdot \mathbf{I}_n & w_r \cdot \mathbf{I}_n & & & \\ w_z \cdot \mathbf{I}_n & & \mu_u \cdot \mathbf{I}_n & & \\ & & & w_z \cdot \mathbf{I}_q & w_r \cdot \mathbf{I}_q \\ -\mathbf{Q}_M & & & \mu_u \cdot \mathbf{I}_q & \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{\chi} \\ \boldsymbol{\gamma} \\ \boldsymbol{\delta} \\ \boldsymbol{\zeta} \\ \boldsymbol{\rho} \end{pmatrix} = \begin{pmatrix} \log_{\hat{g}} \hat{\boldsymbol{G}} \\ \log_{\hat{g}} \hat{\boldsymbol{H}} \\ \log_{\hat{g}} \hat{\boldsymbol{S}}_6 \\ \log_g \boldsymbol{Z} \end{pmatrix},$$

where $(\mathbf{Q}_M)_{j,i} = \log_g(M_{j,i}) = m_{j,i}$ if $\boldsymbol{M}_j = (M_{j,1}, \ldots, M_{j,n})$ denotes the j-th queried message, $\hat{\boldsymbol{G}} = (\hat{G}_1, \ldots, \hat{G}_n)$, $\hat{\boldsymbol{H}} = (\hat{H}_1, \ldots, \hat{H}_n)$, $\hat{\boldsymbol{S}}_6 = (\hat{\sigma_6}^{(1)}, \ldots, \hat{\sigma_6}^{(q)})$ and $\boldsymbol{Z} = (Z^{(1)}, \ldots, Z^{(q)})$. It is easy to see that, as long as there exists $i \in \{1, \ldots, n\}$ such that $M_{j^\star,i} \neq M_i^\star$, the vector

$$(m_1^\star - m_{j^\star,1}, \ldots, m_n^\star - m_{j^\star,n}, 0, 0, \ldots, 0) \in \mathbb{Z}_p^{3n+q}$$

is independent of the rows of the above matrix. Consequently, $\prod_{i=1}^n \big(M_{j^\star,i}/M_i^\star\big)^{-\chi_i}$ is completely unpredictable to $\mathcal{A}$ if $\boldsymbol{M}_{j^\star} \neq \boldsymbol{M}^\star$. For this reason, we can only have $Z^\dagger = 1_\mathbb{G}$ with probability $1/p$, as claimed. □

G Details on the Underlying CCA2-Secure Encryption Schemes

G.1 Proof of Theorem 4

Proof. The proof considers a sequence of games. For each i, we call S_i the event that the adversary wins in Game_i.

Game_0**:** This is the real game. In this game, the adversary $\mathcal{A}$ chooses a target tag $\tau^\star$ at the outset of the game and obtains a public key pk. Then, $\mathcal{A}$ starts invoking the decryption oracle on tag-ciphertext pairs (τ, C) such that $\tau \neq \tau^\star$. In the challenge phase, $\mathcal{A}$ chooses distinct messages $M_0, M_1 \in \mathbb{G}$ and obtains in return an encryption $C^\star$ of M_β under the tag $\tau^\star$, where $\beta \xleftarrow{R} \{0,1\}$ is randomly chosen by $\mathcal{B}$. After a new series of decryption queries (τ, C) such that $\tau \neq \tau^\star$, $\mathcal{A}$ outputs a bit $\beta' \in \{0,1\}$ and wins if $\beta' = \beta$. We denote by S_0 the latter event.

Game_1**:** We modify the decryption oracle. At each decryption query, $\big(\tau, C = (C_0, C_1, C_2, Z, R)\big)$, the challenger $\mathcal{B}$ does not only return $\perp$ when

$$e(Z, \hat{G}_z) \cdot e(R, \hat{G}_r) \neq e(C_1, \hat{G_1}^{\tau} \cdot \hat{G}_2)^{-1} \cdot e(C_2, \hat{G}_2)^{-1},$$

but also rejects C if the equalities

$$Z = {C_1}^{-\varphi_1} \cdot (C_1^\tau)^{-\varphi_2} \cdot {C_2}^{-\varphi_3}, \qquad R = {C_1}^{-\vartheta_1} \cdot (C_1^\tau)^{-\vartheta_2} \cdot {C_2}^{-\vartheta_3} \tag{24}$$

are not satisfied. Otherwise, it computes and returns $M = C_0/{C_1}^x$.

Clearly, Game_1 and Game_0 are identical until the event F_1 that $\mathcal{A}$ rejects a ciphertext (τ, C) that would not have been rejected in Game_0. This only occurs if (τ, C) satisfies (13) but not (24). We show that event F_1 would contradict the DDH assumption in $\mathbb{G}$. Indeed, if F_1 occurs, $\mathcal{B}$ can compute its own homomorphic signature

$$Z^\dagger = {C_1}^{-\varphi_1} \cdot (C_1^\tau)^{-\varphi_2} \cdot {C_2}^{-\varphi_3}, \qquad R^\dagger = {C_1}^{-\vartheta_1} \cdot (C_1^\tau)^{-\vartheta_2} \cdot {C_2}^{-\vartheta_3},$$

on the vector (C_1, C_1^τ, C_2) which, by construction, satisfies

$$e(Z^\dagger, \hat{G}_z) \cdot e(R^\dagger, \hat{G}_r) = e(C_1, \hat{G_1}^{\tau} \cdot \hat{G}_2)^{-1} \cdot e(C_2, \hat{G}_2)^{-1}$$

and $(Z^\dagger, R^\dagger) \neq (Z, R)$. By doing so, $\mathcal{B}$ obtains two distinct homomorphic signatures on the vector (C_1, C_1^τ, C_2) and, by dividing them, obtains a pair $\big(Z^\dagger/Z, R^\dagger/R\big)$ such that

$$e(Z^\dagger/Z, \hat{G}_z) \cdot e(R^\dagger/R, \hat{G}_r) = 1_{\mathbb{G}_T}.$$

This implies that any occurrence of F_1 allows $\mathcal{B}$ to solve an instance $(\hat{G}_z, \hat{G}_r)$ of the Double Pairing problem and also become a DDH distinguisher in $\hat{\mathbb{G}}$.

We thus have the inequality $|\Pr[S_1] - \Pr[S_0]| \leq \Pr[F_1] \leq \mathbf{Adv}^{\mathrm{DDH}}_{\hat{\mathbb{G}}}(\lambda)$. Note that event F_1 also covers the case of an adversary $\mathcal{A}$ that manages to re-randomize the $(Z^\star, R^\star)$ components of the challenge ciphertext $C^\star$.

Game_2**:** We modify the generation of $\mathsf{pk} = \big(g, h, X_1, X_2, S, W, T, V, \mathsf{pk}'_{hsig}, \{(Z_i, R_i)\}_{i=1}^4\big)$. Namely, $\mathcal{B}$ defines

$$X_1 = g^x, \qquad X_2 = h^x$$

for a random $x \xleftarrow{R} \mathbb{Z}_p$. Then, it chooses $\alpha_s, \beta_s, \alpha_t \xleftarrow{R} \mathbb{Z}_p$ and sets

$$\begin{aligned} S &= g^{\alpha_s} \cdot X_1^{\beta_s}, & T &= X_1^{-\beta_s \cdot \tau^\star} \cdot g^{\alpha_t} \\ W &= h^{\alpha_s} \cdot X_2^{\beta_s}, & V &= X_2^{-\beta_s \cdot \tau^\star} \cdot h^{\alpha_t}. \end{aligned} \tag{25}$$

Note that public key components (X_1, X_2, S, T, W, V) have the same distribution as in Game_1 since we are implicitly defining $\alpha = \alpha_s + \beta_s \cdot x$ and $\beta = -\beta_s \cdot x \cdot \tau^\star + \alpha_t$.

In the challenge phase, $\mathcal{B}$ picks $\theta_1, \theta_2 \xleftarrow{R} \mathbb{Z}_p$ and sets

$$(C_0^\star, C_1^\star, C_2^\star, Z^\star, R^\star) = \big(M_\beta \cdot X_1^{\theta_1} \cdot X_2^{\theta_2},\ g^{\theta_1} \cdot h^{\theta_2},\ (g^{\theta_1} \cdot h^{\theta_2})^{\alpha_s \cdot \tau^\star + \alpha_t},\ Z^\star,\ R\big)$$

where

$$Z^\star = {C_1^\star}^{-\varphi_1} \cdot (C_1^{\star\tau^\star})^{-\varphi_2} \cdot {C_2^\star}^{-\varphi_3}, \qquad R^\star = {C_1^\star}^{-\vartheta_1} \cdot (C_1^{\star\tau^\star})^{-\vartheta_2} \cdot {C_2^\star}^{-\vartheta_3}.$$

We remark that $(Z^\star, R^\star)$ is computed using $\mathsf{sk}'_{hsig} = \{(\varphi_i, \vartheta_i)\}_{i=1}^3$ as a simulated QA-NIZK proof that $(C_1^\star, C_1^{\star\tau^\star}, C_2^\star)$ belongs to the row space of $\mathbf{L}$. Note, however, that it is a simulated proof for a true statement, so that $(Z^\star, R^\star)$ has the same distribution as a real proof computed using the witnesses $(\theta_1, \theta_2) \in \mathbb{Z}_p^2$. The challenge ciphertext is thus distributed as in Game_1 and $\Pr[S_2] = \Pr[S_1]$.

Game_3**:** In this game, we modify again the decryption oracle. When $\mathcal{A}$ queries a pair (τ, C), $\mathcal{B}$ parses C as $(C_0, C_1, C_2, Z, R) \in \mathbb{G}^5$. If the latter elements satisfy (24), $\mathcal{B}$ computes and returns

$$M = C_0 \cdot \big(C_2 / C_1^{\alpha_s \cdot \tau + \alpha_t}\big)^{-\frac{1}{\beta_s \cdot (\tau - \tau^\star)}},$$

which is well-defined since $\tau \neq \tau^\star$. If (24) does not hold, $\mathcal{B}$ outputs $\perp$.

$\mathcal{A}$'s view remains the same as in Game_2 until the event F_3 that the input-output behavior of the decryption oracle departs from that of the decryption oracle in Game_2. This only happens if (τ, C) is such that $C_2 \neq C_1^{\alpha \cdot \tau + \beta}$ (so that (C_1, C_1^τ, C_2) is outside the row space of $\mathbf{L}$) but still satisfies (24). We claim that this only occurs with negligible probability $\Pr[F_3] \leq q/(p-q)$.

Indeed, let us consider what a computationally unbounded adversary $\mathcal{A}$ can infer about $\mathsf{sk}'_{hsig} = \{(\varphi_1, \varphi_2, \varphi_3, \vartheta_1, \vartheta_2, \vartheta_3)\}$ during the whole game. In the public key pk, the discrete logarithms of $\{\hat{G}_i = \hat{G}_z^{\varphi_i} \cdot \hat{G}_r^{\vartheta_i}\}_{i=1}^3$ provide 3 linear equations and those of $\{(Z_i, R_i)\}_{i=1}^4$ only provide $\mathcal{A}$ with two additional independent equations. The reason is that, since $\mathbf{L}$ has rank 2, the information provided by (Z_2, R_2) and (Z_4, R_4) is redundant with that supplied by (Z_1, R_1) and (Z_3, R_3). In addition, $\{R_i\}_{i=1}^4$ are uniquely determined $\{Z_i\}_{i=1}^4$ and do not reveal any more information than them. As a consequence, from $\mathcal{A}$'s point view, the vector $(\varphi_1, \varphi_2, \varphi_3, \vartheta_1, \vartheta_2, \vartheta_3)$ is uniformly distributed in a one-dimensional subspace. Hence, at the first decryption query such that (C_1, C_1^τ, C_2) evades the row space of $\mathbf{L}$, the equalities

$$Z = {C_1}^{-\varphi_1} \cdot (C_1^\tau)^{-\varphi_2} \cdot {C_2}^{-\varphi_3}, \qquad R = {C_1}^{-\vartheta_1} \cdot (C_1^\tau)^{-\vartheta_2} \cdot {C_2}^{-\vartheta_3} \tag{26}$$

can only hold with probability $1/p$. However, each query potentially allows $\mathcal{A}$ to eliminate one candidate for the vector $(\varphi_1, \varphi_2, \varphi_3, \vartheta_1, \vartheta_2, \vartheta_3)$. At the k-th query, the equalities (26) thus hold with probability smaller than $1/(q-k)$. The inequality $|\Pr[S_3] - \Pr[S_2]| \leq q/(p-q)$ follows.

Game_4**:** This game like Game_3 with one last change in the generation of pk. Namely, $\mathcal{B}$ defines

$$X_1 = g^x, \qquad X_2 = h^{x'}$$

for random $x, x' \xleftarrow{R} \mathbb{Z}_p$. All other public key components are computed as previously. In particular, $\mathcal{B}$ still obtains (S, W, T, V) as per (25). Clearly, any significant change in $\mathcal{A}$'s behavior would imply a DDH distinguisher in $\mathbb{G}$ and we find $|\Pr[S_4] - \Pr[S_3]| \leq \mathbf{Adv}_{\mathbb{G}}^{\mathrm{DDH}}(\lambda)$.

A consequence of this modified distribution of pk is that the opening oracle now always gives the correct answer. Indeed, for any ciphertext, (C_0, C_1, C_2, Z, R), there always exist exponents $\theta_1, \theta_2 \in \mathbb{Z}_p$ such that

$$(C_1, C_2) = \big(g^{\theta_1} \cdot h^{\theta_2},\ (S^\tau \cdot T)^{\theta_1} \cdot (W^\tau \cdot V)^{\theta_2}\big),$$

so that the decryption oracle computes $M = C_0 \cdot X_1^{-\theta_1} \cdot X_2^{-\theta_1}$.

In Game_4, we claim that $\Pr[S_4] = 1/2$, so that the adversary has no advantage at all. Indeed, the challenge ciphertext $C^\star$ is computed as

$$\tilde{\boldsymbol{C}}_{\sigma_2} = (C_0^\star, C_1^\star, C_2^\star, Z^\star, R^\star) = \left(\sigma_2^\star \cdot X_1^{\theta_3} \cdot X_2^{\theta_4},\ g^{\theta_3} \cdot h^{\theta_4},\ (g^{\theta_3} \cdot h^{\theta_4})^{\alpha_s \cdot \tau^\star + \alpha_t},\ Z^\star,\ R^\star\right)$$

where

$$Z^\star = {C_1^\star}^{-\varphi_1} \cdot (C_1^{\star\tau^\star})^{-\varphi_2} \cdot {C_2^\star}^{-\varphi_3}, \qquad R^\star = {C_1^\star}^{-\vartheta_1} \cdot (C_1^{\star\tau^\star})^{-\vartheta_2} \cdot {C_2^\star}^{-\vartheta_3}.$$

This means that $(C_2^\star, Z^\star, R^\star)$ do not reveal any more information about (θ_1, θ_2) than $C_1^\star$ does. Hence, even given $(C_2^\star, Z^\star, R^\star)$, the pair $(C_0^\star, C_1^\star)$ is a perfectly hiding commitment to M_β.

When putting the above altogether, we can bound $\mathcal{A}$'s advantage by

$$\mathbf{Adv}(\lambda) \leq \mathbf{Adv}_{\mathbb{G}}^{\mathrm{DDH}}(\lambda) + \mathbf{Adv}_{\hat{\mathbb{G}}}^{\mathrm{DDH}}(\lambda) + \frac{2q}{p-q},$$

which is negligible. □

G.2 On the Use of the Boyen-Mei-Waters Technique

Our shortest group signatures are obtained by notably eliminating the randomness of the chameleon hash function from each signature. To this end, instead of using a tag-based encryption scheme, we encrypt σ_2 using the following CCA2-secure encryption scheme.

Keygen(cp)**:** Given common public parameters $\mathsf{cp} = (\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, p)$, conduct the following steps.

1. Choose $g, h \xleftarrow{R} \mathbb{G}$ and a collision-resistant hash function $H : \{0,1\}^* \to \{0,1\}^\ell$, for some $\ell = \mathsf{poly}(\lambda)$.
2. Choose $x, \alpha_0, \alpha_1, \ldots, \alpha_\ell \xleftarrow{R} \mathbb{Z}_p$ and set $X_1 = g^x$, $X_2 = h^x$ and $S_i = g^{\alpha_i}$, and $V_i = h^{\alpha_i}$ for $i \in \{0, 1, \ldots, \ell\}$.
3. Generate a key pair $(\mathsf{pk}'_{hsig}, \mathsf{sk}'_{hsig})$ for the homomorphic signature of Section in order to sign vectors in $\mathbb{G}^{\ell+2}$. Let $\mathsf{pk}'_{hsig} = \left(G_z, G_r, \{G_i\}_{i=1}^{\ell+2}\right)$ be the public key and let $\mathsf{sk}'_{hsig} = \{(\varphi_i, \vartheta_i)\}_{i=1}^3$ be the corresponding private key.
4. Use sk'_{hsig} to generate linearly homomorphic signatures $\{(Z_i, R_i)\}_{i=1}^{2\ell+2}$ on the rows of the matrix

$$\mathbf{L} = \left(\begin{array}{cccccc|c}
g & & & & & & S_0 \\
h & & & & & & V_0 \\
 & g & & & & & S_1 \\
 & & g & & & & S_2 \\
 & & & \ddots & & & \vdots \\
 & & & & g & & S_\ell \\
 & h & & & & & V_1 \\
 & & h & & & & V_2 \\
 & & & \ddots & & & \vdots \\
 & & & & h & & V_\ell
\end{array}\right) \in \mathbb{G}^{(2\ell+2)\times(\ell+2)}$$

which form a subspace of rank $\ell + 1$ spanned by row 1 and rows 3 to $\ell + 2$. The public key

$$\mathsf{pk} := \left(g,\ h,\ X_1,\ X_2,\ \{(S_i, V_i)\}_{i=0}^{\ell},\ \mathsf{pk}'_{hsig},\ \{(Z_i, R_i)\}_{i=1}^{2\ell+2},\ H\right)$$

and the secret key is $\mathsf{sk} = x$.

Encrypt(pk, M)**:** To encrypt $M \in \mathbb{G}$, choose θ_1, θ_2 and compute

$$\begin{aligned}\boldsymbol{C} &= (C_0, C_1, C_2, Z, R)\\ &= \Big(M \cdot X_1^{\theta_1} \cdot X_2^{\theta_2},\ g^{\theta_1} \cdot h^{\theta_2},\ (S_0 \cdot \prod_{i=1}^{\ell} S_i^{\tau[i]})^{\theta_1} \cdot\ (V_0 \cdot \prod_{i=1}^{\ell} V_i^{\tau[i]})^{\theta_2},\\ &\quad (\prod_{i=1}^{\ell} Z_{i+2}^{\tau} \cdot Z_1)^{\theta_1} \cdot (\prod_{i=1}^{\ell} Z_{i+\ell+3}^{\tau} \cdot Z_2)^{\theta_2},\ (\prod_{i=1}^{\ell} R_{i+2}^{\tau} \cdot R_1)^{\theta_1} \cdot (\prod_{i=1}^{\ell} R_{i+\ell+2}^{\tau} \cdot R_2)^{\theta_2}\Big).\end{aligned}$$

where $\tau = H(C_0, C_1)$. Here, (Z, R) serves as a QA-NIZK argument showing that the vector $(C_1, C_1^{\tau[1]}, \ldots, C_1^{\tau[\ell]}, C_2) \in \mathbb{G}^{\ell+2}$ is in the row space of $\mathbf{L}$ and satisfies

$$e(Z, \hat{G}_z) \cdot e(R, \hat{G}_r) = e(C_1, \hat{G}_1 \cdot \prod_{i=1}^{\ell} \hat{G}_{1+i}^{\tau[i]} \cdot G_2)^{-1} \cdot e(C_2, \hat{G}_{\ell+2})^{-1} \tag{27}$$

Decrypt$(\mathsf{sk}, \boldsymbol{C})$**:** Parse $\boldsymbol{C}$ as above. Return $\perp$ if (Z, R) does not satisfy (27). Otherwise, return $M = C_0/C_1^x$.

The proof of IND-CCA2 security proceeds in the same way as that of the TBE scheme with the difference that the reduction uses Waters' technique [61] as in [19, Theorem 3.1] instead of the all-but-one technique of Boneh and Boyen [16].

Of course, in the group signature scheme, $(\boldsymbol{C}_{\sigma_1}, \boldsymbol{C}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_{\tilde{z}}, \boldsymbol{C}_{\tilde{r}}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3)$ must all be hashed in the computation of $\tau \in \{0,1\}^{\ell}$ in order to guarantee anonymity in the CCA2 sense.

H Definitions for Group Signatures

H.1 Static Groups

In the BMW model [12], a group signature scheme is a tuple $(\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$ of efficient algorithms with the following specifications:

Keygen(λ, N)**:** takes as input a security parameter λ and an integer $N \in \mathsf{poly}(N)$, which denotes the maximal number of group members. It outputs a tuple $(\mathsf{gpk}, \mathsf{gmsk}, \mathbf{gsk})$ where gpk is the *group public key*, gmsk is the group manager secret key, and $\mathbf{gsk}$ is a vector of secret keys where, for each $j \in \{1, \ldots, N\}$, $\mathsf{gsk}[j]$ is the signing key of the j-th user.

Sign$(\mathsf{gpk}, \mathsf{gsk}[j])$**:** takes as input the group public key gpk, a signing key $\mathsf{gsk}[j]$ and a message $M \in \{0,1\}^*$. It outputs a signature σ.

Verify$(\mathsf{gpk}, \sigma, M)$**:** is deterministic and takes as input the group public key gpk, a message M and a candidate signature σ of M. It outputs either 0 or 1.

Open$(\mathsf{gpk}, \mathsf{gmsk}, M, \sigma)$**:** is deterministic and takes as inputs the group public key gpk, the group manager secret key gmsk, a message M and a valid group signature σ w.r.t. gpk. It returns an index $j \in \{1, \ldots, N\}$ or a special symbol $\perp$ in case of opening failure.

The *correctness* requirement mandates that, for all integers λ and N, all $(\mathsf{gpk}, \mathsf{gmsk}, \mathbf{gsk})$ produced by Keygen with (λ, N) as input, all indexes $j \in \{1, \ldots, N\}$ and $M \in \{0,1\}^*$, we have $\mathsf{Verify}(\mathsf{gpk}, M, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)) = 1$ and $\mathsf{Open}(\mathsf{gpk}, \mathsf{gmsk}, M, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)) = j$, with probability negligibly close to 1 over the internal randomness of Keygen and Sign.

Bellare *et al.* [12] gave a rigorous security model where all security properties of group signatures are subsumed by two notions called *full anonymity* and *full traceability*. Informally, traceability refers to the inability of colluding group members to create a signature that cannot be opened to one of them. As for the anonymity property, it mandates that it be infeasible to distinguish signatures produced by distinct group members, even if all group members' private keys are revealed.

$\underline{\mathbf{Exp}^{\mathsf{anon\text{-}b}}_{\mathcal{GS},\mathcal{A}}(t,N)}$
$(\mathsf{gpk},\mathsf{gmsk},\mathbf{gsk}) \leftarrow \mathsf{Keygen}(1^t,1^N)$
$(\mathtt{st},j_0,j_1,M) \leftarrow \mathcal{A}(choose,\mathsf{gpk},\mathbf{gsk})$
$\Sigma^\star \leftarrow \mathsf{Sign}(\mathsf{gpk},\mathsf{gsk}[j_b],M)$
$b' \leftarrow \mathcal{A}(guess,\mathtt{st},\Sigma^\star)$
Return b'

$\underline{\mathbf{Exp}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(t,N)}$
$(\mathsf{gpk},\mathsf{gmsk},\mathbf{gsk}) \leftarrow \mathsf{Keygen}(1^t,1^N)$
$\mathtt{st} \leftarrow (\mathsf{gmsk},\mathsf{gpk})$
$\mathcal{C} \leftarrow \emptyset$; $K \leftarrow \varepsilon$; $Cont \leftarrow \mathtt{true}$
while $(Cont = \mathtt{true})$ do
  $(Cont,\mathtt{st},j) \leftarrow \mathcal{A}^{\mathcal{GS}.\mathsf{Sign}(\cdot,\mathsf{gsk}[\cdot],\cdot)}(choose,\mathtt{st},K)$
  if $Cont = \mathtt{true}$ then $\mathcal{C} \leftarrow \mathcal{C} \cup \{j\}$;
    $K \leftarrow \mathsf{gsk}[j]$
  end if
end while;
$(M^\star,\Sigma^\star) \leftarrow \mathcal{A}^{\mathcal{GS}.\mathsf{Sign}(\cdot,\mathsf{gsk}[\cdot],\cdot)}(guess,\mathtt{st})$
if $\mathsf{Verify}(\mathsf{gpk},M^\star,\Sigma^\star) = 0$ then Return 0
if $\mathsf{Open}(\mathsf{gpk},\mathsf{gmsk},M^\star,\Sigma^\star) = \perp$ then Return 1
if $\exists j^\star \in \{1,\ldots,N\}$ such that
  $(\mathsf{Open}(\mathsf{gpk},\mathsf{gmsk},M^\star,\Sigma^\star) = j^\star) \wedge (j^\star \notin \mathcal{C})$
  $\wedge$ ($(j^\star,M^\star)$ not queried by $\mathcal{A}$)
then Return 1 else Return 0

Fig. 1. Experiments for the definitions of anonymity and full traceability

Anonymity. Anonymity requires that, without the group manager's secret key, an adversary cannot recognize the identity of a user given its signature. More formally, the attacker, modeled as a two-stage adversary (*choose* and *guess*), is engaged in the first random experiment depicted in Figure 1. The *advantage* of such an adversary $\mathcal{A}$ against a group signature $\mathcal{GS}$ with N members is defined as

$$\mathbf{Adv}^{\mathsf{anon}}_{\mathcal{GS},\mathcal{A}}(t,N) = \left|\Pr[\mathbf{Exp}^{\mathsf{anon\text{-}1}}_{\mathcal{GS},\mathcal{A}}(\lambda,N)=1] - \Pr[\mathbf{Exp}^{\mathsf{anon\text{-}0}}_{\mathcal{GS},\mathcal{A}}(\lambda,N)=1]\right|.$$

Definition 7 (Full anonymity, [12]). *A group signature is* fully anonymous *if, for any polynomial N and any* PPT *adversaries $\mathcal{A}$ (resp.* PPT *adversaries $\mathcal{A}$ with access to an opening oracle which cannot be queried for the challenge signature), $\boldsymbol{Adv}^{\mathsf{anon}}_{\mathcal{GS},\mathcal{A}}(\lambda,N)$ is a negligible function in the security parameter λ.*

Full traceability. Full traceability ensures that all signatures, even those created by a coalition of users *and* the group manager, pooling their secret keys together, can be traced to a member of the forging coalition. Once again, the attacker is modeled as a two-stage adversary who is run within the second experiment described in Figure 1. Its success probability against $\mathcal{GS}$ is defined as

$$\mathbf{Succ}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(\lambda,N) = \Pr[\mathbf{Exp}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(\lambda,N)=1].$$

Definition 8 (Full traceability, [12]). *A group signature scheme $\mathcal{GS}$ is said to be* fully traceable *if for all polynomial $N(t)$ and all* PPT *adversaries $\mathcal{A}$, its success probability $\boldsymbol{Succ}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(t,N)$ is negligible in the security parameter λ.*

H.2 Dynamic Groups

In the setting of dynamic groups, the syntax of group signatures includes an interactive protocol which allows users to register as new members of the group at any time. The syntax and the security model are those defined by Kiayias and Yung [46]. Like the very similar Bellare-Shi-Zhang model [15], the Kiayias-Yung (KY) model assumes an interactive *join* protocol whereby a prospective user becomes a group member by interacting with the group manager. This protocol provides the user with a membership certificate and a membership secret.

We denote by $N \in \mathsf{poly}(\lambda)$ the maximal number of group members. A dynamic group signature scheme consists of the following algorithms or protocols.

Setup(λ, N)**:** given a security parameter λ and a maximal number of group members $N \in \mathbb{N}$, this algorithm is run by a trusted party to generate a group public key $\mathcal{Y}$, the group manager's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$. Each key is given to the appropriate authority while $\mathcal{Y}$ is made public. The algorithm also initializes a public state St comprising a set data structure $St_{\mathsf{users}} = \emptyset$ and a string data structure $St_{\mathsf{trans}} = \epsilon$.

Join: is an interactive protocol between the group manager GM and a user $\mathcal{U}_i$ where the latter becomes a group member. The protocol involves two interactive Turing machines $\mathsf{J}_{\mathsf{user}}$ and J_{GM} that both take $\mathcal{Y}$ as input. The execution, denoted as $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$, ends with user $\mathcal{U}_i$ obtaining a membership secret sec_i, that no one else knows, and a membership certificate cert_i. If the protocol is successful, the group manager updates the public state St by setting $St_{\mathsf{users}} := St_{\mathsf{users}} \cup \{i\}$ as well as $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle i, \mathsf{transcript}_i \rangle$.

Sign: given a membership certificate cert_i, a membership secret sec_i and a message M, this algorithm outputs a signature σ.

Verify: given a signature σ, a message M and a group public key $\mathcal{Y}$, this deterministic algorithm returns either 0 or 1.

Open: takes as input a message M, a valid signature σ w.r.t. $\mathcal{Y}$, the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ and the public state St. It outputs $i \in St_{\mathsf{users}} \cup \{\perp\}$, which is the identity of a group member or a symbol indicating an opening failure.

Each membership certificate contains a unique tag that identifies the user.

The correctness requirement basically captures that, if all parties *honestly* run the protocols, all algorithms are correct w.r.t. their specification (the formal definition is recalled in Appendix H.2).

The Kiayias-Yung model [46] considers three security notions defined in Appendix H.2. The notion of security against *misidentification attacks* requires that, even if the adversary can introduce users under its control in the group, it cannot produce a signature that traces outside the set of dishonest users. The security against *framing attacks* implies that honest users can never be accused of having signed messages that they did not sign, even if the whole system conspired against them. The *anonymity* property is also formalized by granting the adversary access to a signature opening oracle as in the models of [15].

Correctness for Dynamic Group Signatures. Following the Kiayias-Yung terminology [46], we say that a public state St is *valid* if it can be reached from $St = (\emptyset, \varepsilon)$ by a Turing machine having oracle access to J_{GM}. Also, a state St' is said to *extend* another state St if it is within reach from St.

Moreover, as in [46], when we write $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$, it means that there exists coin tosses ϖ for J_{GM} and J_{user} such that, for some valid public state St', the execution of $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St', \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})](\varpi)$ provides $\mathsf{J}_{\mathsf{user}}$ with $\langle i, \mathsf{sec}_i, \mathsf{cert}_i \rangle$.

Definition 9 (Correctness). *A dynamic group signature scheme is correct if the following conditions are all satisfied:*

1. *In a valid state St, $|St_{users}| = |St_{trans}|$ always holds and two distinct entries of St_{trans} always contain certificates with distinct tag.*
2. *If $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$ is run by two honest parties following the protocol and $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ is obtained by $\mathsf{J}_{\mathsf{user}}$, then it holds that $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$.*
3. *For each $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ such that $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$, satisfying condition 2, it always holds that $\mathsf{Verify}\big(\mathsf{Sign}(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, M), M, \mathcal{Y}\big) = 1$.*
4. *For any outcome $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle$ of the interaction $[\mathsf{J}_{\mathsf{user}}(.,.), \mathsf{J}_{\mathsf{GM}}(., St, ., .)]$ for some valid state St, if $\sigma = \mathsf{Sign}(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, M)$, then*

$$\mathsf{Open}(M, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St') = i.$$

We formalize security properties via experiments where the adversary interacts with a stateful interface $\mathcal{I}$ that maintains the following variables:

- $\mathsf{state}_{\mathcal{I}}$: is a data structure representing the state of the interface as the adversary invokes the various oracles available in the attack games. It is initialized as $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda, N)$. It includes the (initially empty) set St_{users} of group members and a dynamically growing database St_{trans} storing the transcripts of previously executed join protocols.
- $n = |St_{users}| < N$ denotes the current cardinality of the group.
- Sigs: is a database of signatures created by the signing oracle. Each entry consists of a triple (i, M, σ) indicating that message M was signed by user i.
- U^a: is the set of users that were introduced by the adversary in the system in an execution of the join protocol.
- U^b: is the set of honest users that the adversary, acting as a dishonest group manager, introduced in the system. For these users, the adversary obtains the transcript of the join protocol but not the user's membership secret.

When mounting attacks, adversaries will be granted access to the following oracles.

- Q_{pub}, Q_{keyGM} and Q_{keyOA}: when these oracles are invoked, the interface looks up $\mathsf{state}_{\mathcal{I}}$ and returns the group public key $\mathcal{Y}$, the GM's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ respectively.
- $Q_{\mathsf{a\text{-}join}}$: allows the adversary to introduce users under his control in the group. On behalf of the GM, the interface runs J_{GM} in interaction with the $\mathsf{J}_{\mathsf{user}}$-executing adversary who plays the role of the prospective user in the join protocol. If this protocol successfully ends, the interface increments n, updates St by inserting the new user n in both sets St_{users} and U^a. It also sets $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle n, \mathsf{transcript}_n \rangle$.
- $Q_{\mathsf{b\text{-}join}}$: allows the adversary, acting as a corrupted group manager, to introduce new honest group members of his/her choice. The interface triggers an execution of $[\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}]$ and runs $\mathsf{J}_{\mathsf{user}}$ in interaction with the adversary who runs J_{GM}. If the protocol successfully completes, the interface increments n, adds user n to St_{users} and U^b and sets $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle n, \mathsf{transcript}_n \rangle$. It stores the membership certificate cert_n and the membership secret sec_n in a *private* part of $\mathsf{state}_{\mathcal{I}}$.
- Q_{sig}: given a message M, an index i, the interface checks whether the private area of $\mathsf{state}_{\mathcal{I}}$ contains a certificate cert_i and a membership secret sec_i. If no such elements $(\mathsf{cert}_i, \mathsf{sec}_i)$ exist or if $i \notin U^b$, the interface returns $\perp$. Otherwise, it outputs a signature σ on behalf of user i and also sets $\mathsf{Sigs} \leftarrow \mathsf{Sigs} || (i, M, \sigma)$.
- Q_{open}: when this oracle is invoked on input of a valid pair (M, σ), the interface runs algorithm Open using the current state St. When S is a set of pairs of the form (M, σ), $Q_{\mathsf{open}}^{\neg S}$ denotes a restricted oracle that only applies the opening algorithm to pairs (M, σ) which are not in S.
- Q_{read} and Q_{write}: are used by the adversary to read and write the content of $\mathsf{state}_{\mathcal{I}}$. Namely, at each invocation, Q_{read} outputs the whole $\mathsf{state}_{\mathcal{I}}$ but the public/private keys and the private part of $\mathsf{state}_{\mathcal{I}}$ where membership secrets are stored after $Q_{\mathsf{b\text{-}join}}$-queries. By using Q_{write}, the adversary can modify $\mathsf{state}_{\mathcal{I}}$ at will as long as it does not remove or alter elements of St_{users}, St_{trans} or invalidate the public state St: for example, the adversary is allowed to create dummy users as long as he/she does not re-use already existing certificate tags.

Using this formalism, we can now properly define the three announced security properties.

Security Against Misidentification Attacks In a misidentification attack, the adversary can corrupt the opening authority using the Q_{keyOA} oracle. Moreover, he/she can also introduce malicious users in the group via $Q_{\mathsf{a\text{-}join}}$-queries. His/her purpose is to come up with a valid signature $\sigma^\star$. He/she is deemed successful if the produced signature $\sigma^\star$ does not open to any adversarially-controlled.

Definition 10. *A dynamic group signature scheme is secure against misidentification attacks if, for any PPT adversary $\mathcal{A}$ involved in the experiment hereunder, we have*

$$\mathbf{Adv}^{\text{mis-id}}_{\mathcal{A}}(\lambda) = \Pr[\,\mathbf{Exp}^{\text{mis-id}}_{\mathcal{A}}(\lambda) = 1] \in \mathsf{negl}(\lambda).$$

Experiment $\mathbf{Exp}^{\text{mis-id}}_{\mathcal{A}}(\lambda)$
1. $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda, N)$;
2. $(M^\star, \sigma^\star) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{a\text{-}join}}, Q_{\mathsf{read}}, Q_{\mathsf{keyOA}})$;
3. *If* $\mathsf{Verify}(\sigma^\star, M^\star, \mathcal{Y}) = 0$ *returns* 0;
4. $i = \mathsf{Open}(M^\star, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St')$;
5. *If* $i \notin U^a$ *returns* 1;
6. *Returns* 0;

Non-Frameability Framing attacks consider the situation where the entire system, including the group manager and the opening authority, is colluding against some honest user. The adversary can corrupt the group manager as well as the opening authority (via oracles Q_{keyGM} and Q_{keyOA}, respectively). He/she is also allowed to introduce honest group members (via $Q_{\mathsf{b\text{-}join}}$-queries), observe the system while these users sign messages and create dummy users using Q_{write}. The adversary eventually aims at framing an honest group member.

Definition 11. *A dynamic group signature scheme is secure against framing attacks if, for any PPT adversary $\mathcal{A}$ involved in the experiment below, it holds that $\mathbf{Adv}^{\text{fra}}_{\mathcal{A}}(\lambda) = \Pr[\,\mathbf{Exp}^{\text{fra}}_{\mathcal{A}}(\lambda) = 1] \in \mathsf{negl}(\lambda)$.*

Experiment $\mathbf{Exp}^{\text{fra}}_{\mathcal{A}}(\lambda)$
1. $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda, N)$;
2. $(M^\star, \sigma^\star) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{keyOA}}, Q_{\mathsf{b\text{-}join}}, Q_{\mathsf{sig}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
3. *If* $\mathsf{Verify}(\sigma^\star, M^\star, \mathcal{Y}) = 0$ *returns* 0;
4. *If* $i = \mathsf{Open}(M^\star, \sigma^\star, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St') \notin U^b$ *returns* 0;
5. *If* $\big(\bigwedge_{j \in U^b \text{ s.t. } j=i} (j, M^\star, *) \notin \mathsf{Sigs}\big)$ *returns* 1;
6. *Returns* 0;

Full Anonymity The notion of anonymity is formalized by means of a game involving a two-stage adversary. The first stage is called play stage and allows the adversary $\mathcal{A}$ to modify $\mathsf{state}_{\mathcal{I}}$ via Q_{write}-queries and open arbitrary signatures by probing Q_{open}. When the play stage ends, $\mathcal{A}$ chooses a message $M^\star$ as well as two pairs $(\mathsf{sec}^\star_0, \mathsf{cert}^\star_0)$ and $(\mathsf{sec}^\star_1, \mathsf{cert}^\star_1)$, consisting of a valid membership certificate and a corresponding membership secret. Then, the challenger flips a coin $d \leftarrow \{0,1\}$ and computes a challenge signature $\sigma^\star$ using $(\mathsf{sec}^\star_d, \mathsf{cert}^\star_d)$. The adversary is given $\sigma^\star$ with the task of eventually guessing the bit $d \in \{0,1\}$. Before doing so, he/she is allowed further oracle queries throughout the second stage, called guess stage, but is restricted not to query Q_{open} for $(M^\star, \sigma^\star)$.

Definition 12. *A dynamic group signature scheme is fully anonymous if, for any PPT adversary $\mathcal{A}$, $\mathbf{Adv}^{\text{anon}}_{\mathcal{A}}(\lambda) := |\Pr[\,\mathbf{Exp}^{\text{anon}}_{\mathcal{A}}(\lambda) = 1] - 1/2|$ is negligible.*

Experiment $\mathbf{Exp}^{\text{anon}}_{\mathcal{A}}(\lambda)$
1. $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(\lambda)$;
2. $\big(aux, M^\star, (\mathsf{sec}^\star_0, \mathsf{cert}^\star_0), (\mathsf{sec}^\star_1, \mathsf{cert}^\star_1)\big)$
   $\leftarrow \mathcal{A}(\mathsf{play};\, Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{open}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
3. *If* $\neg(\mathsf{cert}^\star_b \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}^\star_b)$ *for* $b \in \{0,1\}$, *returns* 0;
4. *If* $\mathsf{cert}^\star_0 = \mathsf{cert}^\star_1$, *returns* 0;
5. *Picks random* $d \leftarrow \{0,1\}$; $\sigma^\star \leftarrow \mathsf{Sign}(\mathcal{Y}, \mathsf{cert}^\star_d, \mathsf{sec}^\star_d, M^\star)$;
6. $d' \leftarrow \mathcal{A}(\mathsf{guess};\, \sigma^\star, aux, Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{open}}^{\neg\{(M^\star,\sigma^\star)\}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
7. *If* $d' = d$ *then returns* 1;
8. *Returns* 0;

I Deferred Proofs for the Scheme of Section 7

I.1 Proof of Theorem 6

Proof. The result is proved via a sequence of games that begins with the real anonymity game and ends with a game where no advantage is left to the adversary. In each game, we define S_i to be the event that the adversary wins.

Game_0**:** This is the real game. Namely, the challenger generates the group public key gpk, the group manager's secret key gmsk as well as all group members' private keys $\{\mathsf{gsk}[j]\}_{j=1}^{N}$. The adversary is run on input of gpk and $\{\mathsf{gsk}[j]\}_{j=1}^{N}$ and is granted access to a signature opening oracle. In the challenge phase, $\mathcal{A}$ outputs two indices $j_0, j_1 \in \{1, \ldots, N\}$ and a message $M^\star$. The challenger picks $b \xleftarrow{R} \{0,1\}$ and returns a challenge $\sigma^\star \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j_b], M^\star)$ which we denote as $\sigma^\star = \big(\boldsymbol{C}^\star_{\sigma_1}, \tilde{\boldsymbol{C}}^\star_{\sigma_2}, \sigma^\star_3, \boldsymbol{C}^\star_{\sigma_4}, \sigma^\star_5, \boldsymbol{C}^\star_{\hat{\sigma_6}}, \boldsymbol{C}^\star_{\tilde{z}}, \boldsymbol{C}^\star_{\tilde{r}}, \boldsymbol{\pi}^\star_1, \boldsymbol{\pi}^\star_2, \boldsymbol{\pi}^\star_3, r^\star_{hash}\big)$. The adversary is allowed further access to the opening oracle for arbitrary signatures but $\sigma^\star$. When $\mathcal{A}$ halts, it outputs a bit $b' \in \{0,1\}$ and wins if $b' = b$. We call S_0 the latter event.

Game_1**:** In this game, we bring a first modification to the signature opening oracle and do not use the extraction trapdoor $\zeta = \log_{\hat{u}_{11}}(\hat{u}_{12})$ of $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ any longer. Instead, at each opening query

$$\sigma = \big(\boldsymbol{C}_{\sigma_1}, \tilde{\boldsymbol{C}}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_{\tilde{z}}, \boldsymbol{C}_{\tilde{r}}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, r_{hash}\big),$$

$\mathcal{B}$ uses $\mathsf{sk}_{tbe} = x$ to decrypt $\tilde{\boldsymbol{C}}_{\sigma_2} = (C_0, C_1, C_2, Z, R)$ when interpreting it as a TBE ciphertext. From the resulting plaintext $\sigma_2 = C_0/C_1^x$, $\mathcal{B}$ checks if $\sigma_2 = \sigma_3^{\mathsf{ID}_j}$ for a registered user's identifier ID_j. If so, $\mathcal{B}$ returns the resulting user index $j \in \{1, \ldots, N\}$. Note that the perfect soundness of Groth-Sahai proofs $\boldsymbol{\pi}_2$, $\boldsymbol{\pi}_3$ on the CRSes $(\boldsymbol{u}_1, \boldsymbol{u}_2)$, $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ ensures that $\log_{\sigma_2}(\sigma_3) = \log_{\hat{g}}(\hat{\sigma_6})$. For this reason, the opening oracle of Game_1 is guaranteed to give the same result as the one of Game_0. Hence, $\mathcal{A}$'s view is not modified by this change and we have $\Pr[S_1] = \Pr[S_0]$.

Game_2**:** We modify again the opening oracle. When the adversary $\mathcal{A}$ queries the opening of a signature $\sigma = \big(\boldsymbol{C}_{\sigma_1}, \tilde{\boldsymbol{C}}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_{\tilde{z}}, \boldsymbol{C}_{\tilde{r}}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, r_{hash}\big)$, $\mathcal{B}$ parses $\tilde{\boldsymbol{C}}_{\sigma_2}$ as (C_0, C_1, C_2, Z, R) and aborts the game in the event that C_1 coincides with the $C_1^\star$ component of $\tilde{\boldsymbol{C}}^\star_{\sigma_2}$ in the challenge signature $\sigma^\star$ (we assume w.l.o.g. that $C_1^\star$ is chosen at the outset of the game). Since $C_1^\star$ is independent of $\mathcal{A}$'s view until the challenge phase, the probability of this failure event F_2 is at most q/p, where q is the number of opening queries. We have $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2] \leq q/p$.

Game_3**:** We modify again the opening oracle and introduce another failure event F_3 which also causes the challenger $\mathcal{B}$ to halt and output 0. The latter is defined to be the event that $\mathcal{A}$ queries the opening of a group signature $\sigma = \big(\boldsymbol{C}_{\sigma_1}, \tilde{\boldsymbol{C}}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_{\tilde{z}}, \boldsymbol{C}_{\tilde{r}}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, r_{hash}\big)$ such that

$$\begin{aligned}\tau &= \mathsf{CMhash}(hk, (\boldsymbol{C}_{\sigma_1}, \boldsymbol{C}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_{\tilde{z}}, \boldsymbol{C}_{\tilde{r}}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3), r_{hash}) \\ &= \mathsf{CMhash}(hk, (\boldsymbol{C}^\star_{\sigma_1}, \boldsymbol{C}^\star_{\sigma_2}, \sigma^\star_3, \boldsymbol{C}^\star_{\sigma_4}, \sigma^\star_5, \boldsymbol{C}^\star_{\hat{\sigma_6}}, \boldsymbol{C}^\star_{\tilde{z}}, \boldsymbol{C}^\star_{\tilde{r}}, \boldsymbol{\pi}^\star_1, \boldsymbol{\pi}^\star_2, \boldsymbol{\pi}^\star_3), r^\star_{hash}) = \tau^\star\end{aligned}$$

Since event F_3 would contradict the collision-resistance of the chameleon hash function, we have $|\Pr[S_3] - \Pr[S_2]| \leq \mathbf{Adv}^{\text{CR-CMhash}}(\lambda)$.

From now on, we are done with relying on the collision-resistance of CMH. We are thus henceforth free to use tk in the following games.

Game_4**:** We bring yet another modification to the opening oracle. At each signature opening query $\sigma = \big(\boldsymbol{C}_{\sigma_1}, \tilde{\boldsymbol{C}}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_{\tilde{z}}, \boldsymbol{C}_{\tilde{r}}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, r_{hash}\big)$, the challenger $\mathcal{B}$ parses the augmented commitment $\tilde{\boldsymbol{C}}_{\sigma_2}$ as (C_0, C_1, C_2, Z, R). The difference with Game_3 is that $\mathcal{B}$ does not only return $\perp$ when

$$e(Z, \hat{G}_z) \cdot e(R, \hat{G}_r) \neq e(C_1, \hat{G}_1{}^{\tau} \cdot \hat{G}_2)^{-1} \cdot e(C_2, \hat{G}_2)^{-1},$$

but also returns $\perp$ if the equalities

$$Z = {C_1}^{-\varphi_1} \cdot (C_1^{\tau})^{-\varphi_2} \cdot {C_2}^{-\varphi_3}, \qquad R = {C_1}^{-\vartheta_1} \cdot (C_1^{\tau})^{-\vartheta_2} \cdot {C_2}^{-\vartheta_3} \tag{28}$$

are not satisfied. Otherwise, it computes $\sigma_2 = C_0/{C_1}^x$ and checks if $\sigma_2 = \sigma_3^{\mathsf{ID}_j}$ for some registered group member's identifier ID_j. If so, $\mathcal{B}$ outputs the corresponding index $j \in \{1, \ldots, N\}$. Otherwise, it outputs $\perp$.

Clearly, Game_4 and Game_3 proceed identically until the event F_4 that $\mathcal{A}$ queries the opening of a signature where $\tilde{\boldsymbol{C}}_{\sigma_2}$ passes the verification test of Game_3 but fails the verification test of Game_4. This means that the TBE ciphertext $\tilde{\boldsymbol{C}}_{\sigma_2} = (\hat{C}_0, C_1, C_2, Z, R)$ satisfies (13) but not (28). We claim that event F_3 would contradict the DDH assumption in $\mathbb{G}$. Indeed, if this event occurs, $\mathcal{B}$ can compute its own linearly homomorphic signature

$$Z^{\dagger} = {C_1}^{-\varphi_1} \cdot (C_1^{\tau})^{-\varphi_2} \cdot {C_2}^{-\varphi_3}, \qquad R^{\dagger} = {C_1}^{-\vartheta_1} \cdot (C_1^{\tau})^{-\vartheta_2} \cdot {C_2}^{-\vartheta_3},$$

on the vector (C_1, C_1^{τ}, C_2) which necessarily satisfies

$$e(Z^{\dagger}, \hat{G}_z) \cdot e(R^{\dagger}, \hat{G}_r) = e(C_1, {\hat{G}_1}^{\tau} \cdot \hat{G}_2)^{-1} \cdot e(C_2, \hat{G}_2)^{-1}$$

and $(Z^{\dagger}, R^{\dagger}) \neq (Z, R)$. This provides $\mathcal{B}$ with two distinct homomorphic signatures on the vector (C_1, C_1^{τ}, C_2), which in turn yield

$$e(Z^{\dagger}/Z, \hat{G}_z) \cdot e(R^{\dagger}/R, \hat{G}_r) = 1_{\mathbb{G}_T}.$$

Hence, if F_4 occurs with non-negligible probability, $\mathcal{B}$ can solve an instance $(\hat{G}_z, \hat{G}_r)$ of the Double Pairing problem and also break the DDH assumption in $\hat{\mathbb{G}}$.

We thus have the inequality $|\Pr[S_4] - \Pr[S_3]| \leq \Pr[F_4] \leq \mathbf{Adv}^{\mathrm{DDH}}_{\hat{\mathbb{G}}}(\lambda)$. Note that event F_4 also covers the event that the adversary $\mathcal{A}$ somehow manages to re-randomize the $(Z^{\star}, R^{\star})$ components of $\tilde{\boldsymbol{C}}^{\star}_{\sigma_2}$ in the challenge signature $\sigma^{\star}$.

Game_5**:** We modify the generation of

$$\mathsf{pk}_{tbe} = \big(g, h, X_1, X_2, S, W, T, V, \mathsf{pk}'_{hsig}, \{(Z_i, R_i)\}_{i=1}^{4}\big)$$

in the group public key. Namely, $\mathcal{B}$ defines

$$X_1 = g^x, \qquad X_2 = h^x$$

for a randomly chosen $x \xleftarrow{R} \mathbb{Z}_p$. Then, it picks $\alpha_s, \beta_s, \alpha_t \xleftarrow{R} \mathbb{Z}_p$ and sets

$$\begin{aligned} S &= g^{\alpha_s} \cdot X_1^{\beta_s}, & T &= X_1^{-\beta_s \cdot \tau^{\star}} \cdot g^{\alpha_t} \\ W &= h^{\alpha_s} \cdot X_2^{\beta_s}, & V &= X_2^{-\beta_s \cdot \tau^{\star}} \cdot h^{\alpha_t}, \end{aligned} \tag{29}$$

where $\tau^{\star}$ is a random element in the range of the chameleon hash function CMhash. Note that (X_1, X_2, S, T, W, V) have the same distribution as in Game_4 since, in the private key of the TBE scheme, we are implicitly defining $\alpha = \alpha_s + \beta_s \cdot x$ and $\beta = -\beta_s \cdot x \cdot \tau^{\star} + \alpha_t$.

When $\tilde{\boldsymbol{C}}^{\star}_{\sigma_2} = (C_0^{\star}, C_1^{\star}, C_2^{\star}, Z^{\star}, R^{\star})$ is computed in the challenge phase, $\mathcal{B}$ picks $\theta_7, \theta_8 \xleftarrow{R} \mathbb{Z}_p$ and sets

$$(C_0^{\star}, C_1^{\star}, C_2^{\star}, Z^{\star}, R^{\star}) = \big(\sigma_2^{\star} \cdot X_1^{\theta_3} \cdot X_2^{\theta_4}, g^{\theta_3} \cdot h^{\theta_4}, (g^{\theta_3} \cdot h^{\theta_4})^{\alpha_s \cdot \tau^{\star} + \alpha_t}, Z^{\star}, R^{\star}\big)$$

where

$$Z^{\star} = {C_1^{\star}}^{-\varphi_1} \cdot (C_1^{\star \tau^{\star}})^{-\varphi_2} \cdot {C_2^{\star}}^{-\varphi_3}, \qquad R^{\star} = {C_1^{\star}}^{-\vartheta_1} \cdot (C_1^{\star \tau^{\star}})^{-\vartheta_2} \cdot {C_2^{\star}}^{-\vartheta_3}.$$

Note that the QA-NIZK proof $(Z^\star, R^\star)$ is computed using the trapdoor $\mathsf{sk}'_{hsig} = \{(\varphi_i, \vartheta_i)\}_{i=1}^3$ as a simulated QA-NIZK proof that the vector $(C_1^\star, C_1^{\star\tau^\star}, C_2^\star)$ belongs to the row space of $\mathbf{L}$. However, it is a simulated proof for a true statement and $(Z^\star, R^\star)$ has the same distribution as if it were computed using the witnesses $(\theta_3, \theta_4) \in \mathbb{Z}_p^2$. Next, $\mathcal{B}$ computes $\boldsymbol{C}^\star_{\sigma_1}, \sigma_3^\star, \boldsymbol{C}^\star_{\sigma_4}, \sigma_5^\star, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}^\star_{\tilde{z}}, \boldsymbol{C}^\star_{\tilde{r}}$ as well as the NIWI proofs $\boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star, \boldsymbol{\pi}_3^\star$ and uses the trapdoor tk of the chameleon hash function to determine $r^\star_{hash} \in \mathcal{R}_{hash}$ such that

$$\tau^\star = \mathsf{CMhash}(hk, (\boldsymbol{C}^\star_{\sigma_1}, \boldsymbol{C}^\star_{\sigma_2}, \sigma_3^\star, \boldsymbol{C}^\star_{\sigma_4}, \sigma_5^\star, \boldsymbol{C}^\star_{\hat{\sigma_6}}, \boldsymbol{C}^\star_{\tilde{z}}, \boldsymbol{C}^\star_{\tilde{r}}, \boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star, \boldsymbol{\pi}_3^\star), r^\star_{hash}).$$

The challenge signature

$$\sigma^\star = \left(\boldsymbol{C}^\star_{\sigma_1}, \tilde{\boldsymbol{C}}^\star_{\sigma_2}, \sigma_3^\star, \boldsymbol{C}^\star_{\sigma_4}, \sigma_5^\star, \boldsymbol{C}^\star_{\hat{\sigma_6}}, \boldsymbol{C}^\star_{\tilde{z}}, \boldsymbol{C}^\star_{\tilde{r}}, \boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star, \boldsymbol{\pi}_3^\star, r^\star_{hash}\right)$$

is thus distributed as in Game_3. It comes that $\Pr[S_5] = \Pr[S_4]$.

Game_6**:** In this game, we modify again the opening oracle. When $\mathcal{A}$ queries the opening of a signature $\sigma = \left(\boldsymbol{C}_{\sigma_1}, \tilde{\boldsymbol{C}}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_{\tilde{z}}, \boldsymbol{C}_{\tilde{r}}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, r_{hash}\right)$, $\mathcal{B}$ parses the commitment $\tilde{\boldsymbol{C}}_{\sigma_2}$ as $(C_0, C_1, C_2, Z, R) \in \mathbb{G}^5$. If the latter tuple satisfies the test (28), $\mathcal{B}$ computes

$$\sigma_2 = C_0 \cdot \left(C_2 / C_1^{\alpha_s \cdot \tau + \alpha_t}\right)^{-\frac{1}{\beta_s \cdot (\tau - \tau^\star)}},$$

which is well-defined unless the failure event of Game_3 occurs, and checks if $\sigma_2 = \sigma_3^{\mathsf{ID}_j}$ for one of the registered members' identifiers ID_j. If so, $\mathcal{B}$ returns the corresponding user index $j \in \{1, \ldots, N\}$. Otherwise, $\mathcal{B}$ outputs $\perp$.

The adversary's view is identical to its view in Game_5 until the event F_6 that the opening oracle gives a different result than the opening oracle of Game_5. This only happens if $\tilde{\boldsymbol{C}}_{\sigma_2}$ is such that $C_2 \neq C_1^{\alpha \cdot \tau + \beta}$ (so that (C_1, C_1^τ, C_2) is outside the row space of $\mathbf{L}$) but still satisfies the test (28). We claim that this only occurs with negligible probability $\Pr[F_6] \leq q/(p-q)$.

To see this, let us consider what an all powerful adversary $\mathcal{A}$ can infer about $\mathsf{sk}'_{hsig} = \{(\varphi_1, \varphi_2, \varphi_3, \vartheta_1, \vartheta_2, \vartheta_3)\}$. In the public key pk_{tbe} of the TBE scheme, the discrete logarithms of $\{\hat{G}_i = \hat{G}_z^{\ \varphi_i} \cdot \hat{G}_r^{\ \vartheta_i}\}_{i=1}^3$ provide 3 linear equations and those of $\{(Z_i, R_i)\}_{i=1}^4$ only provide $\mathcal{A}$ with two more independent equations. Indeed, since $\mathbf{L}$ has rank 2, the information provided by (Z_2, R_2) and (Z_4, R_4) is redundant with that revealed by (Z_1, R_1) and (Z_3, R_3). Moreover, $\{R_i\}_{i=1}^4$ do not reveal any more information than $\{Z_i\}_{i=1}^4$ since they are uniquely determined by $\{Z_i\}_{i=1}^4$. Consequently, in $\mathcal{A}$'s view the vector $(\varphi_1, \varphi_2, \varphi_3, \vartheta_1, \vartheta_2, \vartheta_3)$ is uniformly distributed in a one-dimensional subspace. This implies that, at the first opening query such that (C_1, C_1^τ, C_2) is outside the row space of $\mathbf{L}$, the equalities

$$Z = {C_1}^{-\varphi_1} \cdot (C_1^\tau)^{-\varphi_2} \cdot {C_2}^{-\varphi_3}, \qquad R = {C_1}^{-\vartheta_1} \cdot (C_1^\tau)^{-\vartheta_2} \cdot {C_2}^{-\vartheta_3} \tag{30}$$

can only hold with probability $1/p$. However, each opening query where $\mathcal{B}$ returns $\perp$ potentially allows $\mathcal{A}$ to rule out one candidate for the vector $(\varphi_1, \varphi_2, \varphi_3, \vartheta_1, \vartheta_2, \vartheta_3)$. At the k-th query, the equalities (30) thus hold with probability $\leq 1/(q-k)$. We thus find $|\Pr[S_6] - \Pr[S_5]| \leq q/(p-q)$.

Game_7**:** This game is identical to Game_6 with one final change in the generation of the public key $\mathsf{pk}_{tbe} = \left(g, h, X_1, X_2, S, \hat{W}, T, V, \mathsf{pk}'_{hsig}, \{(Z_i, R_i)\}_{i=1}^4\right)$ of the TBE scheme. Namely, $\mathcal{B}$ defines

$$X_1 = g^x, \qquad X_2 = h^{x'}$$

for random $x, x' \xleftarrow{R} \mathbb{Z}_p$. All remaining components are computed as previously. In particular, $\mathcal{B}$ still computes (S, W, T, V) as per (29). A simple reduction shows that any noticeable change in $\mathcal{A}$'s behavior would imply a DDH distinguisher in $\mathbb{G}$. It comes that $|\Pr[S_7] - \Pr[S_6]| \leq \mathbf{Adv}^{\mathrm{DDH}}_{\mathbb{G}}(\lambda)$.

As a side effect of this modified distribution of pk_{tbe}, we remark that the opening oracle always gives the correct answer since, for any TBE ciphertext, $\tilde{\boldsymbol{C}}_{\sigma_2} = (C_0, C_1, C_2, Z, R)$, there always exist exponents $\theta_3, \theta_4 \in \mathbb{Z}_p$ such that

$$(C_1, C_2) = \big(g^{\theta_3} \cdot h^{\theta_4},\ (S^{\tau} \cdot T)^{\theta_3} \cdot (W^{\tau} \cdot V)^{\theta_4}\big),$$

so that the opening oracle computes $\sigma_2 = C_0 \cdot X_1^{-\theta_3} \cdot X_2^{-\theta_4}$.

Game_8**:** In this game, we modify the distribution of the group public key. Namely, at step 3 of the group key generation phase, we replace $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ by a perfectly hiding Groth-Sahai CRS, where $\hat{\boldsymbol{u}}_2$ is uniformly random in $\hat{\mathbb{G}}^2$ instead of being linearly dependent with $\hat{\boldsymbol{u}}_1$. Clearly, under the DDH assumption in $\mathbb{G}$, $\mathcal{A}$'s view should not be significantly affected by this change and we have $|\Pr[S_8] - \Pr[S_7]| \leq \mathbf{Adv}^{\mathrm{DDH}}_{\hat{\mathbb{G}}}(\lambda)$.

In Game_8, we claim that $\Pr[S_8] = 1/2$, so that the adversary's advantage is zero. Indeed, $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ is a perfectly hiding Groth-Sahai CRS and the same holds for $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ since $\boldsymbol{u}_2 = (X_1, X_2)$ is now linearly independent of $\boldsymbol{u}_1 = (g, h)$. Moreover, in the challenge signature $\sigma^{\star}$, $\tilde{\boldsymbol{C}}^{\star}_{\sigma_2}$ is computed as

$$\begin{aligned}\tilde{\boldsymbol{C}}_{\sigma_2} &= (C_0^{\star}, C_1^{\star}, C_2^{\star}, Z^{\star}, R^{\star}) \\ &= \big(\sigma_2^{\star} \cdot X_1^{\theta_3} \cdot X_2^{\theta_4},\ g^{\theta_3} \cdot h^{\theta_4},\ (g^{\theta_3} \cdot h^{\theta_4})^{\alpha_s \cdot \tau^{\star} + \alpha_t},\ Z^{\star},\ R^{\star}\big)\end{aligned}$$

where

$$Z^{\star} = {C_1^{\star}}^{-\varphi_1} \cdot (C_1^{\star\tau^{\star}})^{-\varphi_2} \cdot {C_2}^{\star - \varphi_3}, \qquad R^{\star} = {C_1^{\star}}^{-\vartheta_1} \cdot (C_1^{\star\tau^{\star}})^{-\vartheta_2} \cdot {C_2}^{\star - \vartheta_3},$$

which means that $(C_2^{\star}, Z^{\star}, R^{\star})$ do not reveal any more information about the exponents (θ_3, θ_4) than $C_1^{\star}$ does. Hence, even if $(C_2^{\star}, Z^{\star}, R^{\star})$ is publicized, $\boldsymbol{C}^{\star}_{\sigma_2} = (C_0^{\star}, C_1^{\star})$ remains a perfectly hiding commitment to $\sigma_2^{\star}$ and $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$ and $\boldsymbol{\pi}_3$ remain perfectly NIWI Groth-Sahai proofs.

When combining the above, the adversary's advantage is at most

$$\mathbf{Adv}(\lambda) \leq \mathbf{Adv}^{\text{CR-CMhash}}_{\mathbb{G}}(\lambda) + \mathbf{Adv}^{\mathrm{DDH}}_{\mathbb{G}}(\lambda) + 2 \cdot \mathbf{Adv}^{\mathrm{DDH}}_{\hat{\mathbb{G}}}(\lambda) + \frac{2q}{p-q},$$

which is negligible under the stated assumptions. □

J Shorter Dynamic Group Signatures under Simple Assumptions

The construction follows the modular design of group signature used in [8,3]. In a nutshell, the group manager holds a key pair for a structure-preserving signature, which is used to generate users' membership certificate.

When new members join the group, they thus obtain a structure-preserving signature σ_{Φ} on a group element $\Phi \in \mathbb{G}$ of their choice, which will serve as the public key of a Waters signature [61]. Waters signatures are well-suited to our purposes since they rely on a simple assumption and, when used in combination with our SPS scheme, they make it possible to prove all statements using only linear pairing product equations, which allows for shorter proofs.

At each signature generation, users thus re-randomize their membership certificate σ_{Φ} and use the discrete logarithm $\phi = \log_g(\Phi)$ – which is only known to the user and serves as a membership secret – to generate a Waters signature $\sigma_W = (\sigma_{W,1}, \sigma_{W,2}) = (f^{\phi} \cdot H_{\mathbb{G}}(M)^{\rho}, g^{\rho})$ on the message M. The group signature is eventually comprised of commitments to Φ, σ_{Φ} and σ_W (or, more precisely, their components that still carry information on the signer's identity after re-randomization) and NIWI proofs that they satisfy the appropriate verification equations.

In order to achieve anonymity in the CCA2 sense, the commitment $\boldsymbol{C}_{\Phi}$ to Φ is computed using the TBE encryption scheme of Section 6 where the tag is obtained by hashing all other commitments and proof elements using a chameleon hash function.

Setup(λ, N)**:** given a security parameter $\lambda \in \mathbb{N}$ and the permitted number of users $N \in \mathsf{poly}(\lambda)$,

1. Choose bilinear groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ and define $\mathsf{cp} = (\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, p)$. Choose generators $g, f \xleftarrow{R} \mathbb{G}$, $\hat{g}, \hat{f} \xleftarrow{R} \hat{\mathbb{G}}$ such that $\log_g(f) = \log_{\hat{g}}(\hat{f})$.
2. Generate a key pair $(SK_{\mathsf{SPS}}, PK_{\mathsf{SPS}})$ for the structure-preserving signature of Section 5 in order to sign messages of $n = 1$ group elements. The public key is

$$PK_{\mathsf{SPS}} = \Big(g,\ h,\ \hat{g},\ (v, w),\ \Omega = h^\omega,\ \mathsf{pk}_{pots},\ \mathsf{pk}_{hsps},\ \{(z_j, r_j)\}_{j=1}^3\Big),$$

   where $\mathsf{pk}_{hsps} := \big(\hat{g}_z, \hat{g}_r, \{\hat{g}_i\}_{i=1}^6\big)$ and

$$\mathsf{pk}_{pots} := \big(G_z, G_r, G_t, H_z, H_u, \hat{G}_z, \hat{G}_r, \hat{G}_t, \hat{H}_z, \hat{H}_u, \{\hat{G}_1, \hat{H}_1\}\big).$$

   The private key is $SK_{\mathsf{SPS}} = (\omega, \mathsf{sk}_{pots})$, where $\mathsf{sk}_{pots} := \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$.
3. Generate a key pair $(\mathsf{sk}_{tbe}, \mathsf{pk}_{tbe})$ for the TBE scheme of Section 6. The public key consists of

$$\mathsf{pk}_{tbe} := \Big(g,\ h,\ X_1,\ X_2,\ S,\ W,\ T,\ V,\ \mathsf{pk}'_{hsig},\ \{(Z_i^{(0)}, R_i^{(0)})\}_{i=1}^4\Big)$$

   and the secret key is $\mathsf{sk}_{tbe} := x$. For simplicity, the generator g of pk_{tbe} can be recycled from step 1.
4. Choose a vector $\hat{\boldsymbol{u}}_1 = (\hat{u}_{11}, \hat{u}_{12}) \xleftarrow{R} \hat{\mathbb{G}}^2$ and set $\hat{\boldsymbol{u}}_2 = \hat{\boldsymbol{u}}_1{}^\xi$, where $\xi \xleftarrow{R} \mathbb{Z}_p$. Also, define the vectors $\boldsymbol{u}_1 = (g, X_1)$ and $\boldsymbol{u}_2 = (h, X_2)$. These vectors will form Groth-Sahai CRSes $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ in the perfectly binding setting. The private key sk_{tbe} of the TBE scheme will serve as an extraction trapdoor for commitments generated on the CRS $(\boldsymbol{u}_1, \boldsymbol{u}_2)$.
5. Choose a chameleon hash function $\mathsf{CMH} = (\mathsf{CMKg}, \mathsf{CMhash}, \mathsf{CMswitch})$ with a key pair (hk, tk) and randomness space $\mathcal{R}_{hash}$.
6. Choose vectors $\hat{\mathbf{h}} = (\hat{h}_0, \ldots, \hat{h}_\ell) \in \hat{\mathbb{G}}^\ell$ and $\mathbf{h} = (h_0, \ldots, h_\ell) \in \mathbb{G}^\ell$, where $\ell \in \mathsf{poly}(\lambda)$, such that $\log_g(h_i) = \log_{\hat{g}}(\hat{h}_i)$ for each $i \in \{0, \ldots, \ell\}$. These will be used by group members to generate Waters signatures.
7. Set $\mathcal{S}_{\mathsf{GM}} := SK_{\mathsf{SPS}}$, $\mathcal{S}_{\mathsf{OA}} := \mathsf{sk}_{tbe}$ as authorities' private keys and the group public key is

$$\mathcal{Y} := \Big((\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T),\ g,\ \hat{g},\ f,\ \hat{f},\ \mathbf{h},\ \hat{\mathbf{h}},\ PK_{\mathsf{SPS}},\ \mathsf{pk}_{tbe},\ \mathsf{CMH},\ hk,\ (\boldsymbol{u}_1, \boldsymbol{u}_2),\ (\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)\Big).$$

Join$^{(\mathrm{GM}, \mathcal{U}_i)}$**:** the group manager and the prospective user $\mathcal{U}_i$ run the following interactive protocol $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$:

1. $\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y})$ draws $\phi \xleftarrow{R} \mathbb{Z}_p$ and computes $\Phi = g^\phi$, which is sent to J_{GM} with a signature $sig_i = \mathrm{Sign}_{\mathsf{usk}[i]}(\Phi)$ to J_{GM}. J_{GM} checks that $\Phi \in \mathbb{G}$ and $\mathrm{Verify}_{\mathsf{upk}[i]}\big(\Phi, sig_i\big) = 1$. If not J_{GM} aborts and returns $\perp$. If $\Phi \in \mathbb{G}$ already appears in some entry $\mathsf{transcript}_j$ of the database St_{trans}, J_{GM} halts and returns $\perp$ to $\mathsf{J}_{\mathsf{user}}$.
2. J_{GM} uses SK_{SPS} to certify $\mathcal{U}_i$ as a group member by generating a structure-preserving signature

$$\sigma_\Phi = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \hat{\sigma_6}, z, r, Z, R, U) \in \mathbb{G}^5 \times \hat{\mathbb{G}} \times \mathbb{G}^5$$

   on $\Phi \in \mathbb{G}$ with a randomization token $\sigma_{\Phi,\mathsf{r}} = (\sigma_{2,\mathsf{r}}, \sigma_{4,\mathsf{r}}, \sigma_{1,\mathsf{r}}, z_\mathsf{r}, r_\mathsf{r}) \in \mathbb{G}^5$. The first part of σ_Φ is of the form

$$\begin{array}{llll} \sigma_1 = g^\omega \cdot (v^\tau \cdot w)^s, & \sigma_2 = g^{s\cdot\tau}, & \sigma_3 = g^s, & z = z_1^\omega \cdot (z_2^\tau \cdot z_3)^s \\ \sigma_4 = h^{s\cdot\tau} & \sigma_5 = h^s, & \tilde{\sigma_6} = \hat{g}^\tau, & r = r_1^\omega \cdot (r_2^\tau \cdot r_3)^s, \end{array}$$

   for some $s, \tau \xleftarrow{R} \mathbb{Z}_p$, and the randomization token is

$$\sigma_{\Phi,\mathsf{r}} = (\sigma_{2,\mathsf{r}}, \sigma_{4,\mathsf{r}}, \sigma_{1,\mathsf{r}}, z_\mathsf{r}, r_\mathsf{r}) = (g^\tau, h^\tau, v^\tau, z_2^\tau, r_2^\tau).$$

3. $\mathsf{J_{GM}}$ stores $\mathsf{transcript}_i = (\Phi, \sigma_\Phi, \sigma_{\Phi,\mathsf{r}}, i, \mathsf{upk}[i], sig_i)$ in the database St_{trans}, sends $(\sigma_\Phi, \sigma_{\Phi,\mathsf{r}})$ to $\mathsf{J_{user}}$. $\mathsf{J_{user}}$ halts if σ_Φ is an invalid SPS or if $\sigma_{\Phi,\mathsf{r}}$ is not consistent with $\tilde{\sigma_6} = \hat{g}^\tau$ (this requires to test that $e(\sigma_{2,\mathsf{r}}, \hat{g}) = e(g, \hat{\sigma_6})$ and similarly for other components of $\sigma_{\Phi,\mathsf{r}}$). Otherwise, $\mathsf{J_{user}}$ defines the membership certificate as $\mathsf{cert}_i = \big(\Phi, \sigma_\Phi, \sigma_{\Phi,\mathsf{r}}\big) \in \mathbb{G} \times (\mathbb{G}^5 \times \hat{\mathbb{G}} \times \mathbb{G}^5) \times \mathbb{G}^5$, where Φ will serve as the tag identifying $\mathcal{U}_i$. The membership secret sec_i is defined as $\mathsf{sec}_i = \phi \in \mathbb{Z}_p$.

Sign$(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, M)$**:** To sign $M \in \{0,1\}^\ell$, parse the membership certificate cert_i as $\big(\Phi, \sigma_\Phi, \sigma_{\Phi,\mathsf{r}}\big)$, where $\sigma_\Phi = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \hat{\sigma_6}, z, r, Z, R, U) \in \mathbb{G}^5 \times \hat{\mathbb{G}} \times \mathbb{G}^5$ and $\sigma_{\Phi,\mathsf{r}} = (\sigma_{2,\mathsf{r}}, \sigma_{4,\mathsf{r}}, \sigma_{1,\mathsf{r}}, z_\mathsf{r}, r_\mathsf{r}) \in \mathbb{G}^5$. Parse the membership secret sec_i as $\phi \in \mathbb{Z}_p$ and do the following.

1. Re-randomize $\sigma_\Phi = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \hat{\sigma_6}, z, r, Z, R, U)$ by randomly choosing $s'' \xleftarrow{R} \mathbb{Z}_p$ and computing

$$(\sigma_1', \sigma_2', \sigma_3', \sigma_4', \sigma_5', \hat{\sigma_6}, z', r', Z, R, U)$$
$$= (\sigma_1 \cdot \sigma_{1,\mathsf{r}}^{s''} \cdot w^{s''}, \sigma_2 \cdot \sigma_{2,\mathsf{r}}^{s''}, \sigma_3 \cdot g^{s''}, \sigma_4 \cdot \sigma_{4,\mathsf{r}}^{s''}, \sigma_5 \cdot h^{s''}, \hat{\sigma_6}, z \cdot z_\mathsf{r}^{s''}, r \cdot r_\mathsf{r}^{s''}, Z, R, U)$$

   which satisfies

$$\sigma_1' = g^\omega \cdot (v^\tau \cdot w)^{s'}, \quad \sigma_2' = g^{s' \cdot \tau}, \quad \sigma_3 = g^{s'}, \quad z' = z_1^\omega \cdot (z_2^\tau \cdot z_3)^{s'}$$
$$\sigma_4' = h^{s' \cdot \tau} \quad \sigma_5' = h^{s'}, \quad \tilde{\sigma_6} = \hat{g}^\tau, \quad r' = r_1^\omega \cdot (r_2^\tau \cdot r_3)^{s'}$$

   where $s' = s + s''$.
2. Using $\mathsf{sec}_i = \phi \in \mathbb{Z}_p$, generate a Waters signature

$$(\sigma_{W,1}, \sigma_{W,2}) = \big(f^\phi \cdot H_\mathbb{G}(M)^\rho, g^\rho\big) \in \mathbb{G}^2$$

   on the message $M \in \{0,1\}^\ell$, where $\rho \xleftarrow{R} \mathbb{Z}_p$ and $H_\mathbb{G}(M) = h_0 \cdot \prod_{i=1}^\ell h_i^{M[i]}$.
3. Using the CRSes $(\boldsymbol{u_1}, \boldsymbol{u_2})$ and $(\hat{\boldsymbol{u_1}}, \hat{\boldsymbol{u_2}})$, compute Groth-Sahai commitments $\boldsymbol{C}_{\sigma_1}$, $\boldsymbol{C}_{\sigma_2}$, $\boldsymbol{C}_{\sigma_4}$, $\boldsymbol{C}_{\hat{\sigma_6}}$, $\boldsymbol{C}_z$, $\boldsymbol{C}_r$, $\boldsymbol{C}_Z$ $\boldsymbol{C}_R$ and $\boldsymbol{C}_U$ to the variables $(\sigma_1', \sigma_2', \sigma_4', \hat{\sigma_6}, z', r', Z, R, U)$. Also, compute Groth-Sahai commitments $\boldsymbol{C}_\Phi$ and $\boldsymbol{C}_{\sigma_{W,1}}$ to $\Phi \in \mathbb{G}$ and $\sigma_{W,1} \in \mathbb{G}$. Note that $\boldsymbol{C}_\Phi$ can be written as $(C_1, C_0) = (g^{\theta_{\Phi,1}} \cdot h^{\theta_{\Phi_2}}, \Phi \cdot X_1^{\theta_{\Phi,1}} \cdot X_2^{\theta_{\Phi,2}})$ for some $\theta_{\Phi,1}, \theta_{\Phi,2} \in_R \mathbb{Z}_p$.
4. Generate Groth-Sahai NIWI proofs $\boldsymbol{\pi}_1 \in \hat{\mathbb{G}}^2$, $\boldsymbol{\pi}_2, \boldsymbol{\pi}_3, \boldsymbol{\pi}_4 \in \mathbb{G}^2 \times \hat{\mathbb{G}}^2$ and $\boldsymbol{\pi}_5 \in \hat{\mathbb{G}}^2$ that committed variables $(z', r', \sigma_1', \sigma_2', \sigma_4', \hat{\sigma_6}, Z, R, U, \Phi)$ satisfy the pairing product equations

$$e(\boxed{z}', \hat{g_z}) \cdot e(\boxed{r}', \hat{g_r}) \cdot \prod_{i \in \{1,2,4\}} e(\boxed{\sigma_i'}, \hat{g_i}) = \prod_{i \in \{3,5\}} e(\sigma_i', \hat{g_i})^{-1} \cdot e(\Omega, \hat{g_6})^{-1}$$
$$e(\boxed{\sigma_2'}, \hat{g}) = e(\sigma_3', \boxed{\hat{\sigma_6}}), \qquad e(\boxed{\sigma_4'}, \hat{g}) = e(\sigma_5', \boxed{\hat{\sigma_6}}).$$

   and

$$e(G_t, \boxed{\hat{\sigma_6}}) = e(\boxed{Z}, \hat{G_z}) \cdot e(\boxed{R}, \hat{G_r}) \cdot e(\boxed{\Phi}, \hat{G_1})$$
$$1_{\mathbb{G}_T} = e(\boxed{Z}, \hat{H_z}) \cdot e(\boxed{U}, \hat{H_u}) \cdot e(\boxed{\Phi}, \hat{H_1}).$$

5. Generate a NIWI proof $\boldsymbol{\pi}_6 \in \hat{\mathbb{G}}^2$ that variables $(\Phi, \sigma_{W,1}) \in \mathbb{G} \times \mathbb{G}$ satisfy

$$e(\boxed{\sigma_{W,1}}, \hat{g}) = e(\boxed{\Phi}, \hat{f}) \cdot e(\sigma_{W,2}, H_{\hat{\mathbb{G}}}(M)),$$

   where $H_{\hat{\mathbb{G}}}(M) = \hat{h}_0 \cdot \prod_{i=1}^\ell \hat{h}_i^{M[i]}$.

6. Compute a chameleon hash value

$$\tau = \mathsf{CMhash}(hk, (\boldsymbol{C}_{\sigma_1}, \boldsymbol{C}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_Z, \boldsymbol{C}_R, \boldsymbol{C}_U, \boldsymbol{C}_\Phi, \boldsymbol{C}_{\sigma_{W,1}}, \sigma_{W,2}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, \boldsymbol{\pi}_4, \boldsymbol{\pi}_5, \boldsymbol{\pi}_6), r_{hash}), \quad (31)$$

where $r_{hash} \xleftarrow{R} \mathcal{R}_{hash}$. Then, using the tag τ and the random coins $(\theta_{\Phi,1}, \theta_{\Phi,2}) \in \mathbb{Z}_p^2$ of $\boldsymbol{C}_\Phi$, compute $C_2 = (S^\tau \cdot T)^{\theta_{\Phi,1}} \cdot (W^\tau \cdot V)^{\theta_{\Phi,2}}$. Using pk'_{hsig} as a CRS, generate a QA-NIZK argument

$$(Z^{(0)}, R^{(0)}) = \big((Z_3^{(0)^\tau} \cdot Z_1^{(0)})^{\theta_{\Phi,1}} \cdot (Z_4^{(0)^\tau} \cdot Z_2^{(0)})^{\theta_{\Phi,2}}, (R_3^{(0)^\tau} \cdot R_1^{(0)})^{\theta_{\Phi,1}} \cdot (R_4^{(0)^\tau} \cdot R_2^{(0)})^{\theta_{\Phi,2}}\big)$$

that the vector $(C_1, C_1^\tau, C_2) \in \mathbb{G}^3$ is in the row space of $\mathbf{L}$. This allows transforming $\boldsymbol{C}_\Phi$ into a TBE ciphertext $\tilde{\boldsymbol{C}}_\Phi = (C_0, C_1, C_2, Z^{(0)}, R^{(0)})$ as

$$\tilde{\boldsymbol{C}}_\Phi = \big(\Phi \cdot X_1^{\theta_{\Phi,1}} \cdot X_2^{\theta_{\Phi,2}},\ g^{\theta_{\Phi,1}} \cdot h^{\theta_{\Phi,2}},\ (S^\tau \cdot T)^{\theta_{\Phi,1}} \cdot (W^\tau \cdot V)^{\theta_{\Phi,2}},\ (Z_3^{(0)^\tau} Z_1^{(0)})^{\theta_{\Phi,1}} (Z_4^{(0)^\tau} Z_2^{(0)})^{\theta_{\Phi,2}},\ (R_3^{(0)^\tau} R_1^{(0)})^{\theta_{\Phi,1}} (R_4^{(0)^\tau} R_2^{(0)})^{\theta_{\Phi,2}}\big)$$

for the tag τ, which contains the original commitment $\boldsymbol{C}_\Phi$ in its first two coordinates.

Return the signature

$$\sigma = \big(\boldsymbol{C}_{\sigma_1}, \boldsymbol{C}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_Z, \boldsymbol{C}_R, \boldsymbol{C}_U, \tilde{\boldsymbol{C}}_\Phi, \boldsymbol{C}_{\sigma_{W,1}}, \sigma_{W,2}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, \boldsymbol{\pi}_4, \boldsymbol{\pi}_5, \boldsymbol{\pi}_6, r_{hash}\big) \quad (32)$$

Verify$(\sigma, M, \mathcal{Y})$**:** Parse σ as above. Return 1 if and only if: (i) $\tilde{\boldsymbol{C}}_\Phi$ is a valid TBE ciphertext (i.e., (13) holds) for the tag (31); (ii) The NIWI proofs $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, \boldsymbol{\pi}_4, \boldsymbol{\pi}_5, \boldsymbol{\pi}_6$ verify.

Open$(M, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, St)$**:** parse the signature σ as in (32). If $\mathsf{Verify}(\sigma, M, \mathcal{Y}) = 0$, return $\perp$. Otherwise, use $\mathcal{S}_{\mathsf{OA}} = \mathsf{sk}_{tbe}$ to decrypt the Elgamal ciphertext $\boldsymbol{C}_\Phi \in \mathbb{G}^2$ contained in $\tilde{\boldsymbol{C}}_\Phi$. Then, check if the resulting plaintext Φ appears in a record $\mathsf{transcript}_i = (\Phi, \sigma_\Phi, \sigma_{\Phi,\mathrm{r}}, i, \mathsf{upk}[i], sig_i)$ of the user database. If so, return the corresponding i. Otherwise, return $\perp$.

The signature consists of 32 elements of $\mathbb{G}_1$, 14 elements of $\mathbb{G}_2$ and one element of $\mathbb{Z}_p$. In comparison, combining the Abe *et al.* [1] signatures with previous approaches for achieving full anonymity would require at least 40 elements of $\mathbb{G}$ and 26 elements of $\hat{\mathbb{G}}$. If each element of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$) has a 256-bit (resp. 512-bit) representation, our total signature length is 15616 bits (or 1.9 kB) which saves 33% w.r.t. [1]. If we use the Jutla-Roy technique [44] to optimize our structure-preserving signature, we can eliminate $\boldsymbol{C}_r$ from the signature so as to save 2 elements of $\mathbb{G}_1$. The Boyen-Mei-Waters technique [19] further allows dispensing with the randomness r_{hash} of the chameleon hash function. In this case, our signature size drops to 14848 bits (1.81 kB), or 63% of the length enabled by the structure-preserving signatures of Abe *et al.* [1].

Table 2. Comparison between Type-III pairing-based group signatures

Schemes static/dynamic	Group public key size	Signature $(\mathbb{G}, \hat{\mathbb{G}})$-size*	Signature bit-size†	Simple assumptions?
Boyen-Waters [21]‡ (CPA)	$\mathcal{O}(\lambda)$	$(10, 8)$	6 656	✗
Section 7	$(28, 20)^\diamond$	$(20, 8)$	9 216	✓
Section 7 + [44,19]	$\mathcal{O}(\lambda)$	$(17, 8)$	8 448	✓
Groth07 [38]‡	$\mathcal{O}(1)$	$(27, 12)$	13 056	✗
(Known results) [1] + [53,44]	$\mathcal{O}(\lambda)$	$(40, 26)$	23 552	✓
Appendix J	$\mathcal{O}(\lambda)$	$(33, 14)$	15 616	✓
Appendix J + [44,19]	$\mathcal{O}(\lambda)$	$(30, 14)$	14 848	✓

∗ We assume that $\mathbb{Z}_p$ elements are as long as elements of $\mathbb{G}$ for simplicity.
◇ The key size of the chameleon hash function should be added here.
† At a 256-bit (resp. 512-bit) representation of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$) and counting scalars.
‡ Adapted from type-I to type-III pairings.

For the same security level, adapting Groth's efficient construction [38] using in asymmetric pairings would require 27 elements of $\mathbb{G}$ and 12 elements of $\hat{\mathbb{G}}$. Its signature size would amount to 13056 bits (or 1.59 kB). However, it relies on the non-standard q-U assumption [38, Section 2] and, unlike constructions based on structure-preserving signatures [8], it does not support round-optimal concurrent joins [45]. Our optimized construction is thus almost as efficient as the state-of-the-art standard model realization [38] with the benefit of relying on well-established constant-size assumptions.

The security of the scheme is proved in the model of Kiayias and Yung [46], which is recalled in Appendix H.2. Note that, while the model of [46] does not require the opening authority to prove that it correctly opens signatures, our construction readily extends to provide proofs of correct opening as in the model of Bellare *et al.* [15]. By combining the encryption scheme of Section 6 and NIZK proofs for multi-exponentiation equations, the opening authority can convince a judge that ciphertexts are properly decrypted. This can be achieved by having the OA publicize a Groth-Sahai commitment to sk_{tbe} and a NIZK proof that the decrypted value C_0/C_1^x is consistent with the commitment and pk_{tbe}. This approach makes it possible to obtain the opening soundness property of Sakai *et al.* [58].

Theorem 7 (Security against Misidentification attacks). *The scheme is secure against misidentification attacks assuming that the SXDH and XDLIN$_2$ assumption hold in* $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$.

The proof of Theorem 7 relies on the unforgeability of the structure-preserving signature of Section 5 in a standard manner. The proof is rather straightforward and omitted here.

The non-frameability of the scheme relies on the unforgeability of Waters signatures in asymmetric bilinear groups. More precisely, we use a variant of Waters signature where the public key is of the form

$$pk_{\mathsf{W}} = \big(g,\ \hat{g},\ \Phi = g^{\phi},\ f = g^{\alpha_f},\ \hat{f} = \hat{g}^{\alpha_f},\ \boldsymbol{h} = (h_0, \ldots, h_\ell) = g^{\boldsymbol{\beta}},\ \hat{\boldsymbol{h}} = (\hat{h}_0, \ldots, \hat{h}_\ell) = \hat{g}^{\boldsymbol{\beta}}\big),$$

for random $\phi, \alpha_f \in_R \mathbb{Z}_p$, $\boldsymbol{\beta} \in_R \mathbb{Z}_p^{\ell+1}$, and the signature consists of

$$(\sigma_{W,1}, \sigma_{W,2}) = \big(f^{\phi} \cdot (h_0 \cdot \prod_{i=1}^{\ell} h_i^{M[i]})^{\rho}, g^{\rho}\big).$$

The security of this variant relies on a variant of the Computational Diffie-Hellman assumption (CDH) which asserts the hardness of computing $g^{ab} \in \mathbb{G}$ given $(g, \hat{g}, g^a, g^b, \hat{g}^b)$. This assumption is a natural variant of CDH that is implied by the XDLIN$_2$ assumption.

Theorem 8 (Non-frameability). *The scheme is secure against framing attacks assuming that the XDLIN$_2$ assumption holds in* $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$.

Proof. Let us assume that a PPT adversary $\mathcal{A}$ can create a forgery $(M^\star, \sigma^\star)$ that opens to some honest user $i \in U^b$ who did not sign $M^\star$. We give a simple reduction $\mathcal{B}$ that uses $\mathcal{A}$ to break the unforgeability of Waters signature.

Algorithm $\mathcal{B}$ takes as input a public key

$$pk_{\mathsf{W}} = \big((\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T),\ g,\ \hat{g},\ \Phi^{\dagger},\ f,\ \hat{f} = \hat{g},\ \boldsymbol{h} = (h_0, \ldots, h_\ell),\ \hat{\boldsymbol{h}} = (\hat{h}_0, \ldots, \hat{h}_\ell)\big),$$

for the Waters signature and interacts with the adversary $\mathcal{A}$ to mount a chosen-message attack. To generate the group public key $\mathcal{Y}$, $\mathcal{B}$ runs steps 2-5 of the real setup algorithm. As a result, $\mathcal{B}$ knows $\mathcal{S}_{\mathsf{GM}} = SK_{\mathsf{SPS}}$, $\mathcal{S}_{\mathsf{OA}} = \mathsf{sk}_{tbe}$ and the extraction trapdoor $\log_{\hat{u}_{11}}(\hat{u}_{12})$ of the Groth-Sahai CRS $(\hat{\boldsymbol{u}}_{\boldsymbol{1}}, \hat{\boldsymbol{u}}_{\boldsymbol{2}})$. The adversary $\mathcal{B}$ is run on input of the group public key

$$\mathcal{Y} := \Big((\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T),\ g,\ \hat{g},\ f,\ \hat{f},\ \mathbf{h},\ \hat{\mathbf{h}},\ PK_{\mathsf{SPS}},\ \mathsf{pk}_{tbe},\ \mathsf{CMH},\ hk,\ (\boldsymbol{u}_{\boldsymbol{1}}, \boldsymbol{u}_{\boldsymbol{2}}),\ (\hat{\boldsymbol{u}}_{\boldsymbol{1}}, \hat{\boldsymbol{u}}_{\boldsymbol{2}})\Big).$$

If the adversary $\mathcal{A}$ decides to corrupt the group manager or the opening authority during the game, $\mathcal{B}$ can reveal $\mathcal{S}_{\mathsf{GM}} = SK_{\mathsf{SPS}}$ and $\mathcal{S}_{\mathsf{OA}} = \mathsf{sk}_{tbe}$. At the outset of the game, $\mathcal{B}$ picks a random $j^\star \xleftarrow{R} \{1, \ldots, q_b\}$ and interacts with $\mathcal{A}$ as follows.

- Q_{keyGM}-queries: if $\mathcal{A}$ decides to corrupt the group manager, $\mathcal{B}$ surrenders $\mathcal{S}_{\mathsf{GM}} = (sk_{\mathsf{W}}^{(0)}, sk_{\mathsf{W}}^{(1)})$.
- $Q_{\mathsf{b\text{-}join}}$-queries: At any time $\mathcal{A}$ can act as a corrupted group manager and introduce a new honest user i in the group by invoking the $Q_{\mathsf{b\text{-}join}}$ oracle. Then, $\mathcal{B}$ runs $\mathsf{J}_{\mathsf{user}}$ on behalf of the honest user in an execution of Join protocol. The actions taken by $\mathcal{B}$ are dictated by the index $j \in \{1, \ldots, q_b\}$ of the $Q_{\mathsf{b\text{-}join}}$-query.
  - If $j \neq j^\star$, $\mathcal{B}$ follows the exact specification of $\mathsf{J}_{\mathsf{user}}$.
  - If $j = j^\star$, $\mathcal{B}$ sends the value $\Phi^\dagger$ (which it received as part of pk_{W}) to J_{GM} at step 1 of Join. User $j^\star$'s membership secret is thus defined to be the unknown underlying $\mathsf{sec}_{j^\star} = \phi \in \mathbb{Z}_p$. In the rest of the join protocol, $\mathcal{B}$ proceeds like the actual $\mathsf{J}_{\mathsf{user}}$ algorithm and obtains a membership certificate $\mathsf{cert}_{j^\star} = \big(\Phi^\dagger, \sigma_\Phi^\dagger, \sigma_{\Phi,\mathsf{r}}^\dagger\big)$.
- Q_{pub}-queries: These can be answered as in the real game, by having the simulator return $\mathcal{Y}$.
- Q_{sig}-queries: When the adversary $\mathcal{A}$ requests user $i \in U^b$ to sign a message M, $\mathcal{B}$ can answer the query by faithfully running the actual signing algorithm if $i \neq j^\star$. Otherwise (namely, if $i = j^\star$), $\mathcal{B}$ invokes its own challenger to obtain a Waters signature $(\sigma_{W,1}, \sigma_{W,2}) \in \mathbb{G}^2$ on the message M. It also recalls user $j^\star$'s membership certificate $\mathsf{cert}_{j^\star} = \big(\Phi^\dagger, \sigma_\Phi^\dagger, \sigma_{\Phi,\mathsf{r}}^\dagger\big)$ that it obtained from the adversary at the $j^\star$-th $Q_{\mathsf{b\text{-}join}}$-query. Using $(\sigma_{W,1}, \sigma_{W,2})$ and $\mathsf{cert}_{j^\star}$, it can easily run steps 1 and 3-6 of the signing algorithm to generate a valid group signature for M on behalf of user $j^\star$.

When $\mathcal{A}$ halts, it presumably frames some honest user $i^\star \in U^b$ by outputting a signature

$$\sigma^\star = \big(\boldsymbol{C}_{\sigma_1}^\star, \boldsymbol{C}_{\sigma_2}^\star, \sigma_3^\star, \boldsymbol{C}_{\sigma_4}^\star, \sigma_5^\star, \boldsymbol{C}_{\hat{\sigma}_6}^\star, \boldsymbol{C}_z^\star, \boldsymbol{C}_r^\star, \boldsymbol{C}_Z^\star, \boldsymbol{C}_R^\star, \boldsymbol{C}_U^\star, \tilde{\boldsymbol{C}}_\Phi^\star, \boldsymbol{C}_{\sigma_{W,1}}^\star, \sigma_{W,2}^\star, \boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star, \boldsymbol{\pi}_3^\star, \boldsymbol{\pi}_4^\star, \boldsymbol{\pi}_5^\star, \boldsymbol{\pi}_6^\star, r_{hash}^\star\big),$$

for some message $M^\star$, that opens to $i^\star \in U^b$ although user $i^\star$ was never requested to sign $M^\star$. At this point, $\mathcal{B}$ halts and declares failure if $\tilde{\boldsymbol{C}}_\Phi^\star$ does not decrypt to $\Phi^\dagger$ under sk_{tbe} since, in this case, $\mathcal{B}$ was unlucky when choosing $j^\star \in_R \{1, \ldots, q_b\}$. However, with probability $1/q_b$, $\sigma^\star$ does open to the user introduced at the $j^\star$-th $Q_{\mathsf{b\text{-}join}}$-query. In this case, the perfect soundness of Groth-Sahai proofs guarantees that $\boldsymbol{C}_{\sigma_{W,1}}^\star$ is a commitment to a group element $\sigma_{W,1}^\star$ such that

$$e(\sigma_{W,1}^\star, \hat{g}) = e(\Phi^\dagger, \hat{f}) \cdot e(\sigma_{W,2}^\star, H_{\hat{\mathbb{G}}}(M)),$$

which means that $(M^\star, (\sigma_{W,1}^\star, \sigma_{W,2}^\star))$ is a valid forgery for the Waters signature. The result of [61] tells us that, if $\mathcal{A}$ has advantage ε as a framing adversary making at most q_b $Q_{\mathsf{b\text{-}join}}$-queries and q_s signing queries, then $\mathcal{B}$ implies an algorithm solving the aforementioned variant of the CDH problem with advantage $\epsilon/(8 \cdot q_b \cdot q_s \cdot (\ell+1))$. In turn, the latter algorithm implies an efficient XDLIN_2 distinguisher with the same advantage. □

Theorem 9. *The scheme provides full anonymity assuming that: (i) The SXDH assumption holds in $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$; (ii)* CMhash *is a collision-resistant chameleon hash function.*

The proof of the above theorem is very similar to the proof of Theorem 6 except that we need to additionally rely on the security of the homomorphic signature (which does not introduce any other assumption) of Section 2.2 to eliminate an annoying case. Specifically, given that σ_3 and σ_5 appear in the clear in each group signature, we must worry about the event that the adversary chooses to be challenged on two membership certificate $\mathsf{cert}_0^\star, \mathsf{cert}_1^\star$ such that exactly one of these contains a maliciously formed structure-preserving signature $\sigma_\Phi = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \hat{\sigma_6}, z, r, Z, R, U)$ where $\log_g(\sigma_3) \neq \log_h(\sigma_5)$.

Proof. The result is proved via a sequence of games that begins with the real anonymity game and ends with a game where no advantage is left to the adversary. In each game, we define S_i to be the event that the adversary wins.

Game_0**:** This is the real game. Namely, the challenger generates the group public key $\mathcal{Y}$ as well as secret keys $\mathcal{S}_{\mathsf{GM}}$, $\mathcal{S}_{\mathsf{OA}}$ for the group manager and the opening authority. The adversary is run on input of $\mathcal{Y}$ and $\mathcal{A}$ and is granted access to the same oracles as in the real game. In the challenge phase, $\mathcal{A}$ outputs two pairs $(\mathsf{sec}_0^\star, \mathsf{cert}_0^\star), (\mathsf{sec}_1^\star, \mathsf{cert}_1^\star)$ and a message $M^\star$. Recall that $\mathcal{A}$ is able to create valid such pairs on its own since it can obtain $\mathcal{S}_{\mathsf{GM}}$ by invoking the Q_{keyGM} oracle. If $\neg(\mathsf{cert}_b^\star \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_b^\star)$ for each $b \in \{0,1\}$, the challenger $\mathcal{B}$ flips a coin $d \xleftarrow{R} \{0,1\}$ and returns a challenge $\sigma^\star \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{cert}_b^\star, \mathsf{sec}_b^\star, M^\star)$ which we denote as

$$\sigma^\star = \big(\boldsymbol{C}^\star_{\sigma_1}, \boldsymbol{C}^\star_{\sigma_2}, \sigma_3^\star, \boldsymbol{C}^\star_{\sigma_4}, \sigma_5^\star, \boldsymbol{C}^\star_{\hat{\sigma_6}}, \boldsymbol{C}^\star_z, \boldsymbol{C}^\star_r, \boldsymbol{C}^\star_Z, \boldsymbol{C}^\star_R, \boldsymbol{C}^\star_U, \tilde{\boldsymbol{C}}^\star_\Phi, \boldsymbol{C}^\star_{\sigma_{W,1}}, \sigma^\star_{W,2}, \boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star, \boldsymbol{\pi}_3^\star, \boldsymbol{\pi}_4^\star, \boldsymbol{\pi}_5^\star, \boldsymbol{\pi}_6^\star, r^\star_{hash}\big).$$

The adversary is allowed further access to the opening oracle for arbitrary signatures but $\sigma^\star$. When $\mathcal{A}$ halts, it outputs a bit $b' \in \{0,1\}$ and wins if $b' = b$. We call S_0 the latter event.

Game_1**:** This game is like Game_0 except that the challenger $\mathcal{B}$ aborts the experiment if a certain event F_1 comes about. This event F_1 is defined to be the event that, in the challenge phase, the adversary $\mathcal{A}$ chooses two membership certificates $\mathsf{cert}_0^\star, \mathsf{cert}_1^\star$ for which at least one of the underlying structure-preserving signatures $\sigma^\star_{\Phi,0}, \sigma^\star_{\Phi,1}$ – say

$$\sigma^\star_{\Phi,d} = (\sigma^\star_{1,d}, \sigma^\star_{2,d}, \sigma^\star_{3,d}, \sigma^\star_{4,d}, \sigma^\star_{5,d}, \sigma_{\hat{6},d}{}^\star, z^\star_d, r^\star_d, Z^\star_d, R^\star_d, U^\star_d) \in \mathbb{G}^5 \times \hat{\mathbb{G}} \times \mathbb{G}^5$$

for some $d \in \{0,1\}$ – is such that $\log_g(\sigma^\star_{3,d}) \neq \log_h(\sigma^\star_{5,d})$. Note that this implies that the vector $(\sigma^\star_{1,d}, \sigma^\star_{2,d}, \sigma^\star_{3,d}, \sigma^\star_{4,d}, \sigma^\star_{5,d}, \Omega)$ is outside the row space of the matrix $\mathbf{M}$ in (9). For this reason, event F_1 would imply a breach in the security of the instance of the linearly homomorphic signature included in the public key of the SPS scheme. Recall that the private key $\mathcal{S}_{\mathsf{GM}} = SK_{\mathsf{SPS}}$ consists of $(\omega, \mathsf{sk}_{pots})$ and does not include the LHSPS private key sk_{hsps} chosen at step a.3 of the key generation algorithm of the SPS scheme. Hence, even if the adversary obtains $\mathcal{S}_{\mathsf{GM}} = SK_{\mathsf{SPS}}$, it can only create structure-preserving signatures $\sigma^\star_{\Phi,b}$ where $z_b^\star, r_b^\star$ are obtained by homomorphically deriving signatures from the pairs $\{(z_j, r_j)\}_{j=1}^3$ included in PK_{SPS} (in which case, we always have $\log_g(\sigma^\star_{3,d}) = \log_h(\sigma^\star_{5,d})$).

More formally, assuming that F_1 occurs with non-negligible probability, we build a LHSPS forger $\mathcal{B}$ that receives as input a public key pk_{hsps}. It generates the SPS public key by faithfully running steps a.1 and a.2 of the key generation algorithm. It then invokes its own challenger to obtain homomorphic signatures $\{(z_j, r_j)\}_{j=1}^3$ on the rows of $\mathbf{M} \in \mathbb{G}^{3\times 6}$. It then conducts step b of the real SPS key generation algorithm to obtain PK_{SPS} and faithfully runs steps 3-7 of the setup algorithm to obtain a group public key $\mathcal{Y}$. Since $\mathcal{B}$ knows $\mathcal{S}_{\mathsf{OA}} = \mathsf{sk}_{tbe}$ can perfectly simulate the opening oracle as well as all other oracle. By hypothesis, one of the two membership certificates $\mathsf{cert}_d^\star$ of the challenge phase must contain a structure-preserving signature $\sigma^\star_{\Phi,d}$ such that $\log_g(\sigma^\star_{3,d}) \neq \log_h(\sigma^\star_{5,d})$. At this point, $\mathcal{B}$ can win the game against its own challenger by outputting the vector $(\sigma^\star_{1,d}, \sigma^\star_{2,d}, \sigma^\star_{3,d}, \sigma^\star_{4,d}, \sigma^\star_{5,d}, \Omega)$ and the homomorphic signature $(z_d^\star, r_d^\star)$. Since the LHSPS scheme is secure under the DDH assumption in $\hat{\mathbb{G}}$, we thus obtain the inequality $\Pr[F_1] \leq \mathbf{Adv}^{\mathrm{DDH}}_{\hat{\mathbb{G}}}(\lambda)$, so that $|\Pr[S_1] - \Pr[S_0]| \leq \Pr[F_1] \leq \mathbf{Adv}^{\mathrm{DDH}}_{\hat{\mathbb{G}}}(\lambda)$.

Game_2**:** We modify the opening oracle. When the adversary queries the opening of a signature

$$\sigma = \big(\boldsymbol{C}_{\sigma_1}, \boldsymbol{C}_{\sigma_2}, \sigma_3, \boldsymbol{C}_{\sigma_4}, \sigma_5, \boldsymbol{C}_{\hat{\sigma_6}}, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_Z, \boldsymbol{C}_R, \boldsymbol{C}_U, \tilde{\boldsymbol{C}}_\Phi, \boldsymbol{C}_{\sigma_{W,1}}, \sigma_{W,2}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, \boldsymbol{\pi}_4, \boldsymbol{\pi}_5, \boldsymbol{\pi}_6, r_{hash}\big),$$

$\mathcal{B}$ parses the commitment $\tilde{\boldsymbol{C}}_\Phi$ as $(C_0, C_1, C_2, Z^{(0)}, R^{(0)})$ and aborts the game in the event that C_1 coincides with the $C_1^\star$ component of $\tilde{\boldsymbol{C}}^\star_\Phi$ in the challenge signature $\sigma^\star$ (we assume w.l.o.g. that $C_1^\star$ is chosen at the outset of the game). Since $C_1^\star$ is independent of $\mathcal{A}$'s view until the challenge

phase, the probability of this failure event F_2 is at most q/p, where q is the number of queries to the opening oracle. We have $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2] \leq q/p$.

Game$_3$: We modify again the opening oracle and introduce a new failure event F_3 which also causes the challenger $\mathcal{B}$ to halt and output 0. The latter is defined to be the event that $\mathcal{A}$ queries the opening of a signature such that

$$\tau = \mathsf{CMhash}(hk, (\boldsymbol{C}_{\sigma_1}, \ldots, \boldsymbol{\pi}_6), r_{hash}) = \mathsf{CMhash}(hk, (\boldsymbol{C}^{\star}_{\sigma_1}, \ldots, \boldsymbol{\pi}^{\star}_6), r^{\star}_{hash}) = \tau^{\star}$$

We have $|\Pr[S_3] - \Pr[S_2]| \leq \mathbf{Adv}^{\text{CR-CMhash}}(\lambda)$ since F_3 would imply a collision for the chameleon hash function.

From here on, we are free to use tk in the following games since we will not rely on the collision-resistance of CMH anymore.

Game$_4$: We further modify the opening oracle. At each opening query, the challenger $\mathcal{B}$ parses $\tilde{\boldsymbol{C}}_{\Phi}$ as $(C_0, C_1, C_2, Z^{(0)}, R^{(0)})$. The difference with Game$_3$ is that $\mathcal{B}$ does not only return $\perp$ when

$$e(Z^{(0)}, \hat{G}_z) \cdot e(R^{(0)}, \hat{G}_r) \neq e(C_1, \hat{G_1}^{\tau} \cdot \hat{G}_2)^{-1} \cdot e(C_2, \hat{G}_2)^{-1},$$

but also returns $\perp$ if the equalities

$$Z^{(0)} = {C_1}^{-\varphi_1} \cdot (C_1^{\tau})^{-\varphi_2} \cdot {C_2}^{-\varphi_3}, \qquad R^{(0)} = {C_1}^{-\vartheta_1} \cdot (C_1^{\tau})^{-\vartheta_2} \cdot {C_2}^{-\vartheta_3} \tag{33}$$

are not satisfied. Otherwise, it computes $\Phi' = C_0/{C_1}^{x}$ and checks if $\Phi' = \Phi$ for some registered group member's public value $\Phi \in \mathbb{G}$. If so, $\mathcal{B}$ outputs the corresponding index j. Otherwise, it outputs $\perp$.

Clearly, Game$_4$ and Game$_3$ proceed identically until the event F_4 that $\mathcal{A}$ queries the opening of a signature where $\tilde{\boldsymbol{C}}_{\Phi}$ passes the verification test of Game$_3$ but fails the test of Game$_4$. This means that the TBE ciphertext $\tilde{\boldsymbol{C}}_{\Phi} = (C_0, C_1, C_2, Z^{(0)}, R^{(0)})$ satisfies (13) but not (33). We claim that event F_4 would contradict the DDH assumption in $\mathbb{G}$. Indeed, if this event occurs, $\mathcal{B}$ can compute its own linearly homomorphic signature

$$Z^{\dagger} = {C_1}^{-\varphi_1} \cdot (C_1^{\tau})^{-\varphi_2} \cdot {C_2}^{-\varphi_3}, \qquad R^{\dagger} = {C_1}^{-\vartheta_1} \cdot (C_1^{\tau})^{-\vartheta_2} \cdot {C_2}^{-\vartheta_3},$$

on the vector (C_1, C_1^{τ}, C_2) which necessarily satisfies

$$e(Z^{\dagger}, \hat{G}_z) \cdot e(R^{\dagger}, \hat{G}_r) = e(C_1, \hat{G_1}^{\tau} \cdot \hat{G}_2)^{-1} \cdot e(C_2, \hat{G}_2)^{-1}$$

and $(Z^{\dagger}, R^{\dagger}) \neq (Z^{(0)}, R^{(0)})$. This provides $\mathcal{B}$ with two distinct homomorphic signatures on the vector (C_1, C_1^{τ}, C_2), which in turn yield

$$e(Z^{\dagger}/Z^{(0)}, \hat{G}_z) \cdot e(R^{\dagger}/R^{(0)}, \hat{G}_r) = 1_{\mathbb{G}_T}.$$

If F_4 occurs with noticeable probability, $\mathcal{B}$ can solve an instance $(\hat{G}_z, \hat{G}_r)$ of the Double Pairing problem and also defeat the DDH assumption in $\hat{\mathbb{G}}$.

We thus have the inequality $|\Pr[S_4] - \Pr[S_3]| \leq \Pr[F_4] \leq \mathbf{Adv}^{\text{DDH}}_{\hat{\mathbb{G}}}(\lambda)$. Note that event F_4 also covers the event that the adversary $\mathcal{A}$ manages to re-randomize the $(Z^{(0)\star}, R^{(0)\star})$ components of $\tilde{\boldsymbol{C}}^{\star}_{\Phi}$ in the challenge $\sigma^{\star}$.

Game$_5$: The generation of $\mathsf{pk}_{tbe} = \big(g, h, X_1, X_2, S, W, T, V, \mathsf{pk}'_{hsig}, \{(Z_i, R_i)\}_{i=1}^{4}\big)$ is modified in the group public key. Namely, $\mathcal{B}$ defines

$$X_1 = g^x, \qquad X_2 = h^x$$

for a randomly drawn $x \xleftarrow{R} \mathbb{Z}_p$. Then, it picks $\alpha_s, \beta_s, \alpha_t \xleftarrow{R} \mathbb{Z}_p$ and sets

$$\begin{aligned} S &= g^{\alpha_s} \cdot X_1^{\beta_s}, & T &= X_1^{-\beta_s \cdot \tau^{\star}} \cdot g^{\alpha_t} \\ W &= h^{\alpha_s} \cdot X_2^{\beta_s}, & V &= X_2^{-\beta_s \cdot \tau^{\star}} \cdot h^{\alpha_t}, \end{aligned} \tag{34}$$

where $\tau^\star$ is a random element in the range of the chameleon hash function CMhash. Note that (X_1, X_2, S, T, W, V) have the same distribution as in Game_4 since, in the TBE scheme, we are implicitly defining $\alpha = \alpha_s + \beta_s \cdot x$ and $\beta = -\beta_s \cdot x \cdot \tau^\star + \alpha_t$.

When the commitment $\tilde{\boldsymbol{C}}^\star_\Phi = (C^\star_0, C^\star_1, C^\star_2, Z^{(0)\star}, R^{(0)\star})$ is computed in the challenge phase, $\mathcal{B}$ picks $\theta_{\Phi,1}, \theta_{\Phi,2} \xleftarrow{R} \mathbb{Z}_p$ and sets

$$(C^\star_0, C^\star_1, C^\star_2, Z^{(0)\star}, R^{(0)\star}) = \big(\Phi^\star_b \cdot X_1^{\theta_{\Phi,1}} \cdot X_2^{\theta_{\Phi,2}},\ g^{\theta_{\Phi,1}} \cdot h^{\theta_{\Phi,2}},\ (g^{\theta_{\Phi,1}} \cdot h^{\theta_{\Phi,2}})^{\alpha_s \cdot \tau^\star + \alpha_t},\ Z^{(0)\star},\ R^{(0)\star}\big),$$

where $\Phi^\star_b$ is part of $\mathsf{cert}^\star_b$, and

$$Z^{(0)\star} = {C_1}^{\star -\varphi_1} \cdot (C_1^{\star\tau^\star})^{-\varphi_2} \cdot {C_2}^{\star\ -\varphi_3}, \qquad R^{(0)\star} = {C_1}^{\star -\vartheta_1} \cdot (C_1^{\star\tau^\star})^{-\vartheta_2} \cdot {C_2}^{\star\ -\vartheta_3}.$$

Note that the pair $(Z^{(0)\star}, R^{(0)\star})$ is computed using the simulation trapdoor $\mathsf{sk}'_{hsig} = \{(\varphi_i, \vartheta_i)\}_{i=1}^3$ as a simulated QA-NIZK proof that $(C^\star_1, C_1^{\star\tau^\star}, C^\star_2)$ belongs to the row space of $\mathbf{L}$. However, it is a simulated proof for a true statement and, by the quasi-adaptive zero-knowledge property, $(Z^{(0)\star}, R^{(0)\star})$ has the same distribution as if it were computed using the witnesses (θ_3, θ_4). Next, $\mathcal{B}$ computes $\boldsymbol{C}^\star_{\sigma_1}, \sigma^\star_3, \boldsymbol{C}^\star_{\sigma_4}, \sigma^\star_5, \boldsymbol{C}^\star_{\hat{\sigma}_6}, \boldsymbol{C}^\star_{\tilde{z}}, \boldsymbol{C}^\star_{\tilde{r}}$ as well as $\boldsymbol{C}^\star_{\sigma_{W,1}}, \sigma^\star_{W,2}$ and the NIWI proofs $\boldsymbol{\pi}^\star_1, \boldsymbol{\pi}^\star_2, \boldsymbol{\pi}^\star_3$ and uses the trapdoor tk of the chameleon hash function to determine $r^\star_{hash} \in \mathcal{R}_{hash}$ such that

$$\tau^\star = \mathsf{CMhash}(hk, (\boldsymbol{C}^\star_{\sigma_1}, \boldsymbol{C}^\star_{\sigma_2}, \sigma^\star_3, \boldsymbol{C}^\star_{\sigma_4}, \sigma^\star_5, \boldsymbol{C}^\star_{\hat{\sigma}_6}, \boldsymbol{C}^\star_z, \boldsymbol{C}^\star_r, \boldsymbol{C}^\star_Z, \boldsymbol{C}^\star_R, \boldsymbol{C}^\star_U, \boldsymbol{C}^\star_\Phi, \boldsymbol{C}^\star_{\sigma_{W,1}}, \sigma^\star_{W,2}, \boldsymbol{\pi}^\star_1, \boldsymbol{\pi}^\star_2, \boldsymbol{\pi}^\star_3, \boldsymbol{\pi}^\star_4, \boldsymbol{\pi}^\star_5, \boldsymbol{\pi}^\star_6), r^\star_{hash}),$$

where $\boldsymbol{C}^\star_\Phi = (C^\star_0, C^\star_1)$. The challenge signature

$$\sigma^\star = \big(\boldsymbol{C}^\star_{\sigma_1}, \boldsymbol{C}^\star_{\sigma_2}, \sigma^\star_3, \boldsymbol{C}^\star_{\sigma_4}, \sigma^\star_5, \boldsymbol{C}^\star_{\hat{\sigma}_6}, \boldsymbol{C}^\star_z, \boldsymbol{C}^\star_r, \boldsymbol{C}^\star_Z, \boldsymbol{C}^\star_R, \boldsymbol{C}^\star_U, \tilde{\boldsymbol{C}}^\star_\Phi, \boldsymbol{C}^\star_{\sigma_{W,1}}, \sigma^\star_{W,2}, \boldsymbol{\pi}^\star_1, \boldsymbol{\pi}^\star_2, \boldsymbol{\pi}^\star_3, \boldsymbol{\pi}^\star_4, \boldsymbol{\pi}^\star_5, \boldsymbol{\pi}^\star_6, r^\star_{hash}\big).$$

is thus distributed as in Game_4. It comes that $\Pr[S_5] = \Pr[S_4]$.

Game_6**:** We modify again the opening oracle. When $\mathcal{A}$ queries the opening of a signature, $\mathcal{B}$ parses $\tilde{\boldsymbol{C}}_\Phi$ as $(C_0, C_1, C_2, Z^{(0)}, R^{(0)}) \in \mathbb{G}^5$. If the latter tuple satisfies the test (33), $\mathcal{B}$ computes

$$\Phi' = C_0 \cdot \big(C_2 / C_1^{\alpha_s \cdot \tau + \alpha_t}\big)^{-\frac{1}{\beta_s \cdot (\tau - \tau^\star)}},$$

which is well-defined unless the failure event of Game_2 occurs, and checks if $\Phi' = \Phi$ for one of the registered members' identifiers $\Phi \in \mathbb{G}$. If so, $\mathcal{B}$ returns the corresponding user index j. Otherwise, $\mathcal{B}$ outputs $\perp$.

The adversary's view remains identical to its view in Game_5 until the event F_6 that the opening oracle gives a different result than the opening oracle of Game_5. This only happens if $\tilde{\boldsymbol{C}}_\Phi$ is such that $C_2 \neq C_1^{\alpha \cdot \tau + \beta}$ (so that (C_1, C_1^τ, C_2) is outside the row space of $\mathbf{L}$) but still satisfies the test (33). We claim that this only occurs with negligible probability $\Pr[F_6] \leq q/(p-q)$.

To see this, let us consider what an unbounded adversary $\mathcal{A}$ can observe about $\mathsf{sk}'_{hsig} = \{(\varphi_1, \varphi_2, \varphi_3, \vartheta_1, \vartheta_2, \vartheta_3)\}$. In the public key pk_{tbe} of the TBE scheme, the discrete logarithms of $\{\hat{G}_i = \hat{G}_z^{\ \varphi_i} \cdot \hat{G}_r^{\ \vartheta_i}\}_{i=1}^3$ provide 3 linear equations and those of $\{(Z_i^{(0)}, R_i^{(0)})\}_{i=1}^4$ only provide $\mathcal{A}$ with two more independent equations. Indeed, since $\mathbf{L}$ has rank 2, the information supplied by $(Z_2^{(0)}, R_2^{(0)})$ and $(Z_4^{(0)}, R_4^{(0)})$ is redundant with that revealed by $(Z_1^{(0)}, R_1^{(0)})$ and $(Z_3^{(0)}, R_3^{(0)})$. Furthermore, $\{R_i^{(0)}\}_{i=1}^4$ do not reveal any more information than $\{Z_i^{(0)}\}_{i=1}^4$ since they are uniquely determined by $\{Z_i^{(0)}\}_{i=1}^4$. From $\mathcal{A}$'s view, the vector $(\varphi_1, \varphi_2, \varphi_3, \vartheta_1, \vartheta_2, \vartheta_3)$ is uniformly distributed in a one-dimensional subspace. This implies that, at the first opening query such that (C_1, C_1^τ, C_2) is outside the row space of $\mathbf{L}$, the equalities

$$Z^{(0)} = {C_1}^{-\varphi_1} \cdot (C_1^\tau)^{-\varphi_2} \cdot {C_2}^{-\varphi_3}, \qquad R^{(0)} = {C_1}^{-\vartheta_1} \cdot (C_1^\tau)^{-\vartheta_2} \cdot {C_2}^{-\vartheta_3} \tag{35}$$

can only hold with probability $1/p$. However, each opening query where $\mathcal{B}$ returns $\perp$ potentially allows $\mathcal{A}$ to rule out one candidate for the vector $(\varphi_1, \varphi_2, \varphi_3, \vartheta_1, \vartheta_2, \vartheta_3)$. At the k-th query, the equalities (35) thus hold with probability $\leq 1/(q-k)$. We thus find $|\Pr[S_6] - \Pr[S_5]| \leq q/(p-q)$.

Game_7**:** This game is identical to Game_6 with one final change in the generation of the public key $\mathsf{pk}_{tbe} = \big(g, h, X_1, X_2, S, \hat{W}, T, V, \mathsf{pk}'_{hsig}, \{(Z_i^{(0)}, R_i^{(0)})\}_{i=1}^4\big)$ of the TBE scheme. Namely, $\mathcal{B}$ defines

$$X_1 = g^x, \qquad\qquad X_2 = h^{x'}$$

for randomly chosen $x, x' \xleftarrow{R} \mathbb{Z}_p$. All remaining components are computed as previously. In particular, $\mathcal{B}$ still computes (S, W, T, V) as per (29). Any significant change in $\mathcal{A}$'s behavior would imply a DDH distinguisher in $\mathbb{G}$. It follows that $|\Pr[S_7] - \Pr[S_6]| \leq \mathbf{Adv}_{\mathbb{G}}^{\mathrm{DDH}}(\lambda)$.

As a side effect of this modified distribution of pk_{tbe}, we remark that the opening oracle always gives the correct answer since, for any TBE ciphertext, $\tilde{\boldsymbol{C}}_{\sigma_2} = (C_0, C_1, C_2, Z^{(0)}, R^{(0)})$, there always exist exponents $\theta_{\Phi,1}, \theta_{\Phi,2} \in \mathbb{Z}_p$ such that

$$(C_1, C_2) = \big(g^{\theta_{\Phi,1}} \cdot h^{\theta_{\Phi,2}},\ (S^{\tau} \cdot T)^{\theta_{\Phi,1}} \cdot (W^{\tau} \cdot V)^{\theta_{\Phi,2}}\big),$$

so that the opening oracle always computes $\Phi' = C_0 \cdot X_1^{-\theta_{\Phi,1}} \cdot X_2^{-\theta_{\Phi,2}}$.

Game_8**:** We modify the distribution of the group public key. At step 3 of the group key generation phase, we replace $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ by a perfectly hiding Groth-Sahai CRS, where $\hat{\boldsymbol{u}}_2$ is random in $\hat{\mathbb{G}}^2$ instead of being linearly dependent with $\hat{\boldsymbol{u}}_1$. Clearly, under the DDH assumption in $\mathbb{G}$, $\mathcal{A}$'s view should not be significantly affected by this change and we have $|\Pr[S_8] - \Pr[S_7]| \leq \mathbf{Adv}_{\hat{\mathbb{G}}}^{\mathrm{DDH}}(\lambda)$.

In Game_8, we claim that $\Pr[S_8] = 1/2$, so that the adversary's advantage is zero. Indeed, $(\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2)$ is a perfectly hiding Groth-Sahai CRS and the same holds for $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ since $\boldsymbol{u}_2 = (X_1, X_2)$ is now linearly independent of $\boldsymbol{u}_1 = (g, h)$. Also, unless the failure event F_1 of Game_1 occurs, the distribution of $(\sigma_3^\star, \sigma_5^\star)$ (which are given in the clear in the challenge signature $\sigma^\star$) does not depend on the challenge bit $b \in \{0,1\}$. Moreover, $\tilde{\boldsymbol{C}}_{\Phi}^\star$ is computed as

$$\begin{aligned}\tilde{\boldsymbol{C}}_{\Phi}^\star &= (C_0^\star, C_1^\star, C_2^\star, Z^{(0)\star}, R^{(0)\star}) \\ &= \big(\Phi_b^\star \cdot X_1^{\theta_{\Phi,1}} \cdot X_2^{\theta_{\Phi,2}},\ g^{\theta_{\Phi,1}} \cdot h^{\theta_{\Phi,2}},\ (g^{\theta_{\Phi,1}} \cdot h^{\theta_{\Phi,2}})^{\alpha_s \cdot \tau^\star + \alpha_t},\ Z^\star,\ R^\star\big)\end{aligned}$$

where

$$Z^{(0)\star} = {C_1^\star}^{-\varphi_1} \cdot (C_1^{\star\tau^\star})^{-\varphi_2} \cdot {C_2^\star}^{-\varphi_3}, \qquad R^{(0)\star} = {C_1^\star}^{-\vartheta_1} \cdot (C_1^{\star\tau^\star})^{-\vartheta_2} \cdot {C_2^\star}^{-\vartheta_3},$$

which means that $(C_2^\star, Z^{(0)\star}, R^{(0)\star})$ do not reveal any more information about $(\theta_{\Phi,1}, \theta_{\Phi,2})$ than $C_1^\star$ does. Hence, even if the information $(C_2^\star, Z^{(0)\star}, R^{(0)\star})$ is publicized, $\boldsymbol{C}_{\Phi}^\star = (C_0^\star, C_1^\star)$ remains a perfectly hiding commitment to $\Phi_b^\star$ and $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$ and $\boldsymbol{\pi}_3$ remain perfectly NIWI Groth-Sahai proofs.

When combining the above, the adversary's advantage is at most

$$\mathbf{Adv}(\lambda) \leq \mathbf{Adv}_{\mathbb{G}}^{\text{CR-CMhash}}(\lambda) + \mathbf{Adv}_{\mathbb{G}}^{\mathrm{DDH}}(\lambda) + 3 \cdot \mathbf{Adv}_{\hat{\mathbb{G}}}^{\mathrm{DDH}}(\lambda) + \frac{2q}{p-q},$$

which is negligible under the stated assumptions. □

We note that the transition from Game_0 to Game_1 still works in the variant of the scheme where the SPS scheme is optimized via the QA-NIZK proof of Jutla and Roy [44]. Indeed, we can simply rely on the soundness of the QA-NIZK argument and exploit the fact that, in the reduction, $\mathcal{B}$ is allowed to know the discrete logarithms of all entries of the matrix $\mathbf{M}$ w.r.t. the base element $g \in \mathbb{G}$.