Highly-Efficient Fully-Anonymous Dynamic Group Signatures

David Derler[1] and Daniel Slamanig[2,‡]

[1] IAIK, Graz University of Technology, Graz, Austria
david.derler@tugraz.at
[2] AIT Austrian Institute of Technology, Vienna, Austria
daniel.slamanig@ait.ac.at

Abstract. Group signatures are a central tool in privacy-enhancing cryptography, which allow members of a group to anonymously produce signatures on behalf of the group. Consequently, they are an attractive means to implement privacy-friendly authentication mechanisms. Ideally, group signatures are dynamic and thus allow to dynamically and concurrently enroll new members to a group. For such schemes, Bellare et al. (CT-RSA'05) proposed the currently strongest security model (BSZ model). This model, in particular, ensures desirable anonymity guarantees. Given the prevalence of the resource asymmetry in current computing scenarios, i.e., a multitude of (highly) resource-constrained devices are communicating with powerful (cloud-powered) services, it is of utmost importance to have group signatures that are highly-efficient and can be deployed in such scenarios. Satisfying these requirements in particular means that the signing (client) operations are lightweight.

We propose a novel, generic approach to construct dynamic group signature schemes, being provably secure in the BSZ model and particularly suitable for resource-constrained devices. Our results are interesting for various reasons: We can prove our construction secure without requiring random oracles. Moreover, when opting for an instantiation in the random oracle model (ROM) the so obtained scheme is extremely efficient and outperforms the fastest constructions providing anonymity in the BSZ model—which also rely on the ROM—known to date. Regarding constructions providing a weaker anonymity notion than BSZ, we surprisingly outperform the popular short BBS group signature scheme (Crypto'04; also proven secure in the ROM) and thereby even obtain shorter signatures. We provide a rigorous comparison with existing schemes that highlights the benefits of our scheme. On a more theoretical side, we provide the first construction following the "without encryption" paradigm introduced by Bichsel et al. (SCN'10) in the strong BSZ model.

Keywords: group signatures ⋄ BSZ model ⋄ CCA2-full anonymity ⋄ efficiency ⋄ structure-preserving signatures on equivalence classes

---

This is the full version of a paper to appear in ASIA CCS'18: 2018 ACM Asia Conference on Computer and Communication Security, June 4-8, 2018, Incheon, Republic of Korea. ACM, New York, NY, USA. https://doi.org/10.1145/3196494.3196507.

‡ Work done while affiliated with IAIK, Graz University of Technology, Graz, Austria.

1 Introduction

Group signatures, initially introduced by Chaum and van Heyst [CvH91], are an important privacy-enhancing cryptographic tool which allow a group manager to set up a group so that every member of this group can later anonymously sign messages on behalf of the group. Thereby, a dedicated authority (called opening authority) can open a given group signature to determine the identity of the actual signer. They allow authentication without revealing the individuals' identity and are thus an ideal means to implement privacy-friendly authentication. Group signatures have received significant attention from the cryptographic community beginning in the early 90ies [CvH91] and gain increasing practical relevance due to technological innovations in intelligent transportation systems (e.g., floating car data, toll systems, parking) as well as public transportation systems (i.e., smart ticketing), where user privacy is considered to play an important role (cf. [PPF+12]). In particular, operators must not trace and profile passengers, but in case of a court order there is a party that can re-identify the otherwise anonymous passengers. Such privacy aspects are for instance highlighted in the EU Directive 2010/40/EU on intelligent transport systems and more broadly in the upcoming EU General Data Protection Regulation (GPDR), latter becoming effective in May 2018. These developments and in particular considering that users' devices in most use-cases are rather resource constrained, make it important to have particularly efficient group signature candidates at hand.

Motivation. As an illustrative example for the importance of very fast signature generation and verification times, as well as re-identification functionality as provided by group signatures, consider public transportation system where every user needs to sign on passing a gate. Here, a large number of tickets (signatures) need to be processed in the shortest possible time and the more tickets can be processed the more effective and competitive such solutions are. Also for the subjective comfort of users these times are extremely critical. For instance, it makes a significant difference between processing/waiting time of one second compared to 500 or even 250 ms in terms of efficiency of the system as well as user satisfaction. While former is at the edge of acceptability, the latter two values significantly increase the passengers' comfort and will not influence the passenger in his behavior (i.e., requirement to stop and wait at a gate). Moreover it doubles and quadruples the number of processed passengers respectively, which can make a huge difference during peak hours.

In such a scenario, we observe a significant resource asymmetry between user (signer) and verifier devices. While former are typically resource-constrained, latter are much more powerful. Consequently, while verification can be performed by powerful devices, an efficient signing routine running on the resource-constrained clients is the most important characteristic of group signatures.

We also observe that especially the re-identification feature is highly valuable and indeed typically required feature in public-transport, as law-enforcement requires to re-identify users within ongoing investigations, say after terroristic attack. This is a gap which is not covered by the existing body of work on privacy for public transportation [HCDF06, HZB+13, RHBP13, MDND15, RBHP15], as they all build upon privacy-friendly cryptography which *does not* provide a re-identification feature, i.e., single-show attribute-based credentials or e-cash.

Goal. Despite their increasing practical importance, no progress has been made with respect to computational efficiency improvements of group signature schemes providing the desirable notions of CPA- as well as CCA2-full anonymity within the last decade. The most efficient schemes known to date are the BBS group signature scheme [BBS04] (which achieves weaker CPA-full anonymity) and the XSGS group signature scheme [DP06] (which achieves stronger CCA2-full anonymity as required by BSZ). In this paper we set the goal to further push the computational efficiency limits of group signature schemes providing those desirable anonymity notions. Before we discuss our contributions, we take a look at previous work on group signatures to put our contributions in context.

1.1 Previous Work on Group Signatures

Group signatures were first rigorously formalized for static groups by Bellare et al. [BMW03]. In this setting, all members are fixed at setup and also receive their honestly generated keys at setup from the group manager. This model was later extended to the dynamic case by Bellare et al.[BSZ05] (henceforth denoted as BSZ model), where new group members can be dynamically and concurrently enrolled to the group. Further, it separates the role of the issuer and the opener so that they can operate independently. The BSZ model requires a strong anonymity notion, where anonymity of a group signature is preserved even if the adversary can see arbitrary key exposures and openings of other group signatures. A slightly weaker model, which is used to prove the security (and in particular anonymity) of the popular BBS group signature scheme was introduced by Boneh et al. [BBS04]. This model is a relaxation of the BSZ model, and in particular weakens anonymity so that the adversary can not request openings for signatures. As it is common, we refer to this anonymity notion as CPA-full anonymity, whereas we use CCA2-full anonymity to refer to anonymity in the sense of BSZ.

Over the years, two main construction paradigms for group signatures have been established. The first one is the widely used sign-encrypt-prove (SEP) paradigm [CS97]. Here, a signature is an encrypted membership certificate together with a signature of knowledge, where the signer demonstrates knowledge of some signed value in the ciphertext [ACJT00, BBS04, NS04, BSZ05, KY05, DP06, BW07, BW06, Gro07, LPY15, LLM$^+$16, LMPY16]. As an alternative to this paradigm, Bichsel et al. in [BCN$^+$10] proposed an elegant design paradigm for group signatures which does not require to encrypt the membership certificate to produce signatures. Henceforth we call this paradigm sign-randomize-proof (SRP). Essentially, they use a signature scheme which supports (1) randomization of signatures so that multiple randomized versions of the same signature are unlinkable, and (2) efficiently proving knowledge of a signed value. In their construction, on joining the group, the issuer uses such a signature scheme to sign a commitment to the user's secret key. The user can then produce a group signature for a message by randomizing the signature and computing a signature of knowledge on the message, which demonstrates knowledge of the signed secret key. To open signatures, in contrast to constructions following SEP, which support constant time opening by means of decrypting the ciphertext in the signature, constructions in this paradigm require a linear scan, i.e., to check a given signature against each potential user. We, however, want to stress that opening is an infrequent operation typically run on a powerful machine and thus such a linear scan does not impact the practicality of such schemes. Bichsel et al. proposed an instantiation based on the randomizable pairing-based Camenisch-Lysyanskaya (CL) signature scheme [CL04] (whose EUF-CMA security is based on the interactive LRSW assumption). Recently, Pointcheval and Sanders [PS16] proposed another randomizable signature scheme (whose EUF-CMA security is proven in the generic group model), which allows to instantiate the approach due to Bichsel et al. more efficiently.

The main drawback of existing constructions following SRP is that they rely on a security model that is weaker than the BSZ model [BSZ05]: anonymity only holds for users whose keys do not leak (we will henceforth use CCA^- to refer to anonymity in this sense). This essentially means that once a user key leaks, all previous signatures of this user can potentially be attributed to this user. Furthermore, the model used for SRP constructions assumes that the opening authority and the issuing authority are one entity, meaning that the issuer can identify all signers when seeing group signatures. Both aforementioned weakenings can be highly problematic in practical applications of group signatures. It is thus a natural question to ask whether it is possible to prove that constructions following the SRP paradigm provide CPA- or even CCA2-full anonymity. Unfortunately, for existing constructions we have to answer this negatively. Even when allowing to modify the existing constructions in [BCN+10, PS16] to allow the explicit use of encryption upon joining the group (which might solve the separability issue regarding issuer and opener), it is easy to see that knowledge of the user secret key breaks CCA2- as well as CPA-full anonymity for both constructions.[3] Since CCA2-full anonymity straight forwardly implies anonymity in the SRP model, this example confirms that CCA2-full anonymity is a strictly stronger notion. The weaker notion of CPA-full anonymity is somewhat orthogonal to the anonymity notion used by the SRP model: it appropriately models the leakage of user secret keys, but restricts the open oracle access. Yet, in practice it seems that the risk that a user secret key leaks is extremely hard to quantify, which is why we deem CPA-full anonymity to be more desirable. This is also underpinned by the fact that—to the best of our knowledge—no attacks arising from the restriction of the open oracle access in CPA-full anonymity are known.

[3] Each valid group signature contains a valid randomizable signature on the secret key of the user. While group signatures only contain a proof of knowledge of the signed secret key, being in possession of secret key candidates allows to simply test them using the verification algorithm of the randomizable signature scheme. This clearly provides a distinguisher against CCA2- as well as CPA-full anonymity.

1.2 Contribution

We tackle the following open questions, which are of highly practical relevance but also of theoretical interest:

- *Is it possible to further push the computational efficiency limits of group signature schemes providing the more desirable* CPA*-full and* CCA2*-full anonymity notions?*
- *Is it possible to construct schemes providing those more desirable anonymity notions, where compelling efficiency is reached by relying on the SRP paradigm?*

We answer both questions posed above to the affirmative by contributing a novel approach to construct group signatures. Our approach is a composition of structure preserving signatures on equivalence classes (SPS-EQ) [HS14, FHS18], conventional digital signatures, public key encryption, non-interactive zero-knowledge proofs, and signatures of knowledge. Although these tools may sound quite heavy, we obtain conceptually simple and surprisingly efficient group signatures, which provably provide CCA2-full anonymity in the strongest model for dynamic group signatures, i.e., the BSZ model. In doing so, we obtain the first construction which achieves this strong security notion following the SRP paradigm. In addition to that, we introduce an even more efficient CPA-fully anonymous variant of our scheme.

We proceed in showing how to instantiate our constructions in the random oracle model (ROM) to obtain particularly efficient schemes. We are thereby able to further push the long standing computational efficiency limits for both CPA- and CCA2-fully anonymous group signature schemes regarding signature generation and verification. When comparing to the celebrated and popular BBS group signature scheme [BBS04] (which achieves CPA-full anonymity in the ROM), besides being more efficient we surprisingly even obtain shorter signatures. Ultimately, when comparing to instantiations in the vein of Bichsel et al. (which provide a less desirable anonymity notion), our instantiations provide comparable computational efficiency.

A Remark on SRP. While existing constructions following the SRP paradigm (sometimes also called "without encryption" paradigm) do not explicitly use public key encryption, they all rely on assumptions which imply public key encryption. In general, their goal is not to avoid public key encryption, but to make efficiency gains by constructing schemes which do not make explicit use of public key encryption in the sense of SEP, i.e., upon signature generation. We, thus, may henceforth refer to such schemes as "without explicit encryption". Our scheme is in the same spirit and also makes its efficiency gains by avoiding the explicit use of encryption upon signature generation (note that we explicitly use encryption upon joining and opening).

2 Preliminaries

In this section, we provide some preliminaries and recall the required primitives.

Notation. Let $x \xleftarrow{R} X$ denote the operation that picks an element uniformly at random from a finite set X and assigns it to x. We assume that all algorithms run in polynomial time and use $y \leftarrow \mathsf{A}(x)$ to denote that y is assigned the output of the potentially probabilistic algorithm A on input x and fresh random coins and write $y \leftarrow \mathsf{A}(x;\ r)$ to make the random coins r of A explicit. We assume that every algorithm outputs a special symbol $\perp$ on error. We write $\Pr[\Omega : \mathcal{E}]$ to denote the probability of an event $\mathcal{E}$ over the probability space Ω. A function $\epsilon : \mathbb{N} \to \mathbb{R}^+$ is called negligible if for all $c > 0$ there is a k_0 such that $\epsilon(k) < 1/k^c$ for all $k > k_0$. In the remainder of this paper, we use ϵ to denote such a negligible function. We use the $[\cdot]$ operator to access list entries, i.e., let $L = (a, b, \ldots, z)$ then $L[1]$ refers to a.

Let $\mathbb{G}_1 = \langle P \rangle$, $\mathbb{G}_2 = \langle \hat{P} \rangle$, and $\mathbb{G}_T$ be groups of prime order p. A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a map, where it holds for all $(P, \hat{Q}, a, b) \in \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{Z}_p^2$ that $e(aP, b\hat{Q}) = e(P, \hat{Q})^{ab}$, and $e(P, \hat{P}) \neq 1$, and e is efficiently computable. We assume the Type-3 setting, where $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ is known.

Definition 1 (Bilinear Group Generator). *Let* BGGen *be an algorithm which takes a security parameter* κ *and generates a bilinear group* $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ *in the Type-3 setting, where the common group order* p *of the groups* $\mathbb{G}_1, \mathbb{G}_2$ *and* $\mathbb{G}_T$ *is a prime of bitlength* κ*,* e *is a pairing and* P *and* $\hat{P}$ *are generators of* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, respectively.*

Definition 2 (Decisional Diffie-Hellman Assumption (DDH)). *The DDH assumption relative to a prime-order* p *group* $\mathbb{G} = \langle P \rangle$ *with* $\log_2 p = \kappa$ *states that for all PPT adversaries* $\mathcal{A}$ *there exists a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{matrix} b \xleftarrow{R} \{0,1\},\ r, s, t \xleftarrow{R} \mathbb{Z}_p, \\ b^* \leftarrow \mathcal{A}(P, rP, sP, (b \cdot (rs) + (1-b) \cdot t)P \end{matrix} \ :\ b = b^*\right] \leq 1/2 + \epsilon(\kappa).$$

Definition 3 (Symmetric External Diffie-Hellman Assumption (SXDH)). *Let* BG *be a bilinear group generated by* BGGen*. Then, the SXDH assumption states that the DDH assumption holds in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*.*

Additionally, we introduce a natural assumption in the Type-3 bilinear group setting. We justify its plausibility in Appendix A.

Definition 4 (Computational co-Diffie-Hellman Inversion Assumption (co-CDHI)). *The co-CDHI assumption states that for all PPT adversaries* $\mathcal{A}$ *there exists a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{matrix} \mathsf{BG} \leftarrow \mathsf{BGGen}(1^\kappa), \\ a \xleftarrow{R} \mathbb{Z}_p,\ C \leftarrow \mathcal{A}(\mathsf{BG}, aP, 1/a\hat{P}) \end{matrix} \ :\ C = 1/aP\right] \leq \epsilon(\kappa).$$

Structure Preserving Signatures on EQ Classes. Subsequently, we briefly recall structure-preserving signatures on equivalence classes ($\mathsf{SPS\text{-}EQ}$) as presented in [HS14, FHS18]. Therefore, let p be a prime and $\ell > 1$; then $\mathbb{Z}_p^\ell$ is a vector space and one can define a projective equivalence relation on it, which propagates to $\mathbb{G}_i^\ell$ and partitions $\mathbb{G}_i^\ell$ into equivalence classes. Let $\sim_\mathcal{R}$ be this relation, i.e., for $M, N \in \mathbb{G}_i^\ell : M \sim_\mathcal{R} N \Leftrightarrow \exists\, s \in \mathbb{Z}_p^* : M = sN$. An $\mathsf{SPS\text{-}EQ}$ scheme now signs an equivalence class $[M]_\mathcal{R}$ for $M \in (\mathbb{G}_i^*)^\ell$ by signing a representative M of $[M]_\mathcal{R}$. One of the design goals of $\mathsf{SPS\text{-}EQ}$ is to guarantee that two message-signature pairs from the same equivalence class cannot be linked. Below, we formally recall the definition of an $\mathsf{SPS\text{-}EQ}$ scheme.

Definition 5 ($\mathsf{SPS\text{-}EQ}$)**.** *An* $\mathsf{SPS\text{-}EQ}$ *on* $\mathbb{G}_i^*$ *(for* $i \in \{1, 2\}$*) consists of the following PPT algorithms:*

$\mathsf{BGGen}_\mathcal{R}(1^\kappa)$**:** *On input of a security parameter* κ *outputs a bilinear group* BG*.*

$\mathsf{KGen}_\mathcal{R}(\mathsf{BG}, \ell)$**:** *On input of a bilinear group* BG *and a vector length* $\ell > 1$ *outputs a key pair* $(\mathsf{sk}, \mathsf{pk})$*.*

$\mathsf{Sign}_\mathcal{R}(M, \mathsf{sk})$**:** *On input a representative* $M \in (\mathbb{G}_i^*)^\ell$ *and a secret key* sk *outputs a signature* σ *for the equivalence class* $[M]_\mathcal{R}$*.*

$\mathsf{ChgRep}_\mathcal{R}(M, \sigma, \rho, \mathsf{pk})$**:** *On input of a representative* $M \in (\mathbb{G}_i^*)^\ell$ *of class* $[M]_\mathcal{R}$*, a signature* σ *for* M*, a scalar* ρ *and a public key* pk *returns an updated message-signature pair* (M', σ')*, where* $M' = \rho \cdot M$ *is the new representative and* σ' *its updated signature.*

$\mathsf{Vrf}_\mathcal{R}(M, \sigma, \mathsf{pk})$**:** *On input of a representative* $M \in (\mathbb{G}_i^*)^\ell$*, a signature* σ *and a public key* pk *outputs a bit* $b \in \{0, 1\}$*.*

$\mathsf{VKey}_\mathcal{R}(\mathsf{sk}, \mathsf{pk})$**:** *This algorithm on input a secret key* sk *and a public key* pk *outputs a bit* $b \in \{0, 1\}$*.*

For security, one requires the following properties.

Definition 6 (Correctness). *An* $\mathsf{SPS\text{-}EQ}$ *scheme on* $(\mathbb{G}_i^*)^\ell$ *is called* correct *if for all security parameters* $\kappa \in \mathbb{N}$*,* $\ell > 1$*,* $\mathsf{BG} \leftarrow \mathsf{BGGen}_\mathcal{R}(1^\kappa)$*,* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}_\mathcal{R}(\mathsf{BG}, \ell)$*,* $M \in (\mathbb{G}_i^*)^\ell$ *and* $\rho \in \mathbb{Z}_p^*$*:*

$$\mathsf{VKey}_\mathcal{R}(\mathsf{sk}, \mathsf{pk}) = 1 \quad \wedge \quad \Pr\left[\mathsf{Vrf}_\mathcal{R}(M, \mathsf{Sign}_\mathcal{R}(M, \mathsf{sk}), \mathsf{pk}) = 1\right] = 1 \quad \wedge$$
$$\Pr\left[\mathsf{Vrf}_\mathcal{R}(\mathsf{ChgRep}_\mathcal{R}(M, \mathsf{Sign}_\mathcal{R}(M, \mathsf{sk}), \rho, \mathsf{pk}), \mathsf{pk}) = 1\right] = 1.$$

For $\mathsf{EUF\text{-}CMA}$ security, a valid message-signature pair, corresponding to an unqueried equivalence class, is considered a forgery.

Definition 7 ($\mathsf{EUF\text{-}CMA}$)**.** *An* $\mathsf{SPS\text{-}EQ}$ *over* $(\mathbb{G}_i^*)^\ell$ *is* existentially unforgeable under adaptively chosen-message attacks*, if for all PPT adversaries* $\mathcal{A}$ *with access to a signing oracle* $\mathcal{O}^{\mathsf{Sign}_\mathcal{R}}$*, there is a negligible function* $\epsilon(\cdot)$ *such that:*

$$\Pr\left[\begin{array}{l} \mathsf{BG} \leftarrow \mathsf{BGGen}_\mathcal{R}(1^\kappa), \\ (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}_\mathcal{R}(\mathsf{BG}, \ell), \\ (M^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{Sign}_\mathcal{R}(\mathsf{sk}, \cdot)}}(\mathsf{pk}) \end{array} : \begin{array}{r} [M^*]_\mathcal{R} \neq [M]_\mathcal{R} \quad \forall M \in Q^{\mathsf{Sign}_\mathcal{R}} \quad \wedge \\ \mathsf{Vrf}_\mathcal{R}(M^*, \sigma^*, \mathsf{pk}) = 1 \end{array}\right] \leq \epsilon(\kappa),$$

where $Q^{\mathsf{Sign}_\mathcal{R}}$ *is the set of queries that* $\mathcal{A}$ *has issued to the signing oracle* $\mathcal{O}^{\mathsf{Sign}_\mathcal{R}}$*.*

Besides $\mathsf{EUF\text{-}CMA}$ security, an additional security property for $\mathsf{SPS\text{-}EQ}$ was introduced in [FHS15] (cf. Definition 8).

Definition 8 (Perfect Adaption of Signatures). *An* $\mathsf{SPS\text{-}EQ}$ *scheme on* $(\mathbb{G}_i^*)^\ell$ perfectly adapts signatures *if for all tuples* $(\mathsf{sk}, \mathsf{pk}, M, \sigma, \rho)$ *where it holds that* $\mathsf{VKey}_\mathcal{R}(\mathsf{sk}, \mathsf{pk}) = 1$*,* $\mathsf{Vrf}_\mathcal{R}(M, \sigma, \mathsf{pk}) = 1$*,* $M \in (\mathbb{G}_i^*)^\ell$*, and* $\rho \in \mathbb{Z}_p^*$*, the distributions* $(\rho M, \mathsf{Sign}_\mathcal{R}(\rho M, \mathsf{sk}))$ *and* $\mathsf{ChgRep}_\mathcal{R}(M, \sigma, \rho, \mathsf{pk})$ *are identical.*

Digital Signature Schemes. Subsequently, we recall a definition of digital signature schemes.

Definition 9 (Digital Signatures). *A digital signature scheme* Σ *consists of the following PPT algorithms:*

$\mathsf{KGen}(1^\kappa)$: *Takes a security parameter* κ *as input and outputs a secret (signing) key* sk *and a public (verification) key* pk *with associated message space* $\mathcal{M}$ *(we may omit to mention the message space* $\mathcal{M}$*).*

$\mathsf{Sign}(\mathsf{sk}, m)$: *Takes a secret key* sk *and a message* $m \in \mathcal{M}$ *as input and outputs a signature* σ*.*

$\mathsf{Vrf}(\mathsf{pk}, m, \sigma)$: *Takes a public key* pk*, a message* $m \in \mathcal{M}$ *and a signature* σ *as input and outputs a bit* $b \in \{0, 1\}$*.*

Besides correctness we require existential unforgeability under adaptively chosen message attacks ($\mathsf{EUF\text{-}CMA}$) [GMR88]. Below, we recall formal definitions of these properties.

Definition 10 (Correctness). *A digital signature scheme* Σ *is correct, if for all* κ*, all* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa)$*, and all* $m \in \mathcal{M}$ *it holds that* $\Pr[\mathsf{Vrf}(\mathsf{pk}, m, \mathsf{Sign}(\mathsf{sk}, m)) = 1] = 1$.

Definition 11 ($\mathsf{EUF\text{-}CMA}$). *A digital signature scheme* Σ *is* $\mathsf{EUF\text{-}CMA}$ *secure, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that*

$$\Pr\left[\begin{array}{l}(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa), \\ (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{S}(\mathsf{sk}, \cdot)}}(\mathsf{pk})\end{array} : \begin{array}{r}\mathsf{Vrf}(\mathsf{pk}, m^*, \sigma^*) = 1, \ \wedge \\ m^* \notin Q^{\mathsf{S}}\end{array}\right] \leq \epsilon(\kappa) \ ,$$

where $\mathcal{A}$ *has access to an oracle* $\mathcal{O}^{\mathsf{S}}$ *that allows to execute the* Sign *algorithm and the environment keeps track of all message queried to* $\mathcal{O}^{\mathsf{S}}$ *via* Q^{S}*.*

Public Key Encryption. We also require public key encryption, which we recall below.

Definition 12 (Public Key Encryption). *A public key encryption scheme* Ω *consists of the following PPT algorithms:*

$\mathsf{KGen}(1^\kappa)$: *Takes a security parameter* κ *as input and outputs a secret decryption key* sk *and a public encryption key* pk *(and we assume that the message space* $\mathcal{M}$ *is implicit in* pk*).*

$\mathsf{Enc}(\mathsf{pk}, m)$: *Takes a public key* pk *and a message* $m \in \mathcal{M}$ *as input and outputs a ciphertext* c*.*

$\mathsf{Dec}(\mathsf{sk}, c)$: *Takes a secret key* sk *and a ciphertext* c *as input and outputs a message* $m \in \mathcal{M} \cup \{\perp\}$*.*

We require a public key encryption scheme to be correct and $\mathsf{IND\text{-}T}$ secure. Below we formally recall those notions.

Definition 13 (Correctness). *A public key encryption scheme Ω is correct if it holds for all κ, for all $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa)$, and for all messages $m \in \mathcal{M}$ that*

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m] = 1.$$

Definition 14 ($\mathsf{IND\text{-}T}$ Security). *Let $\mathsf{T} \in \{\mathsf{CPA}, \mathsf{CCA2}\}$. A public key encryption scheme Ω is $\mathsf{IND\text{-}T}$ secure, if for all PPT adversaries $\mathcal{A}$ there exists a negligible function $\epsilon(\cdot)$ such that*

$$\Pr\left[\begin{array}{l}(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\kappa),\\ (m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}^{\mathcal{O}_\mathsf{T}}(\mathsf{pk}),\\ \ b \xleftarrow{R} \{0,1\},\ c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b),\\ b^* \leftarrow \mathcal{A}^{\mathcal{O}_\mathsf{T}}(c, \mathsf{st})\end{array} : \begin{array}{r} b = b^* \ \wedge \\ c \notin Q^{\mathsf{Dec}} \ \wedge \ |m_0| = |m_1|\end{array}\right] \leq 1/2 + \epsilon(\kappa),$$

where the adversary runs in two stages, $\mathcal{O}_\mathsf{T} \leftarrow \emptyset$ if $\mathsf{T} = \mathsf{CPA}$, and $\mathcal{O}_\mathsf{T} \leftarrow \{\mathcal{O}^{\mathsf{Dec}}(\mathsf{sk}, \cdot)\}$ if $\mathsf{T} = \mathsf{CCA2}$. Q^{Dec} denotes the list of queries to $\mathcal{O}^{\mathsf{Dec}}$ and we set $Q^{\mathsf{Dec}} \leftarrow \emptyset$ if $\mathsf{T} = \mathsf{CPA}$.

Non-Interactive Zero-Knowledge Proof Systems. Now, we recall a standard definition of non-interactive zero-knowledge proof systems. Therefore, let L_R be an **NP**-language with witness relation $R : L_R = \{x \mid \exists\, w : R(x, w) = 1\}$.

Definition 15 (Non-Interactive Zero-Knowledge Proof System). *A non-interactive proof system Π consists of the following PPT algorithms:*

$\mathsf{Setup}(1^\kappa)$: *Takes a security parameter κ as input, and outputs a common reference string crs.*

$\mathsf{Proof}(\mathsf{crs}, x, w)$: *Takes a common reference string crs, a statement x, and a witness w as input, and outputs a proof π.*

$\mathsf{Vrf}(\mathsf{crs}, x, \pi)$: *Takes a common reference string crs, a statement x, and a proof π as input, and outputs a bit $b \in \{0, 1\}$.*

We require proof systems to be complete, sound, and zero-knowledge. Below, we recall formal definition of those properties (adapted from [BGI14]).

Definition 16 (Completeness). *A non-interactive proof system Π is* complete*, if for every adversary $\mathcal{A}$ it holds that*

$$\Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa),\ (x, w) \leftarrow \mathcal{A}(\mathsf{crs}),\\ \pi \leftarrow \mathsf{Proof}(\mathsf{crs}, x, w)\end{array} : \begin{array}{r}\mathsf{Vrf}(\mathsf{crs}, x, \pi) = 1\\ \vee\ (x, w) \notin R\end{array}\right] \approx 1.$$

Definition 17 (Soundness). *A non-interactive proof system Π is* sound*, if for every PPT adversary $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that*

$$\Pr\left[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa),\ (x, \pi) \leftarrow \mathcal{A}(\mathsf{crs}) : \mathsf{Vrf}(\mathsf{crs}, x, \pi) = 1 \ \wedge \ x \notin L_R\right] \leq \epsilon(\kappa).$$

If we quantify over all adversaries $\mathcal{A}$ and require $\epsilon = 0$, we have perfect soundness, but we present the definition for computationally sound proofs (arguments).

Definition 18 (Adaptive Zero-Knowledge). *A non-interactive proof system* Π *is* adaptively zero-knowledge*, if there exists a PPT simulator* $S = (\mathcal{S}_1, \mathcal{S}_2)$ *such that for every PPT adversary* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that*

$$\left| \begin{array}{l} \Pr\left[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa) \ : \ \mathcal{A}^{\mathcal{P}(\mathsf{crs},\cdot,\cdot)}(\mathsf{crs}) = 1\right] \ - \\ \Pr\left[(\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\kappa) \ : \ \mathcal{A}^{\mathcal{S}(\mathsf{crs},\tau,\cdot,\cdot)}(\mathsf{crs}) = 1\right] \end{array} \right| \leq \epsilon(\kappa),$$

where, τ *denotes a simulation trapdoor. Thereby,* $\mathcal{P}$ *and* $\mathcal{S}$ *return* $\perp$ *if* $(x, w) \notin R$ *or* $\pi \leftarrow \mathsf{Proof}(\mathsf{crs}, x, w)$ *and* $\pi \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, x)$*, respectively, otherwise.*

Signatures of Knowledge. Below we recall signatures of knowledge (SoKs) [CL06], where L_R is as above. For the formal notions we follow [BCC+15] and use a stronger generalization of the original extraction property termed f-extractability. A signature of knowledge (SoK) for L_R is defined as follows.

Definition 19 (Signatures of Knowledge). *A* SoK *consists of the following PPT algorithms:*

$\mathsf{Setup}(1^\kappa)$: *Takes a security parameter* κ *as input and outputs a common reference string* crs*. The message space* $\mathcal{M}$ *is implicitly defined by* crs*.*
$\mathsf{Sign}(\mathsf{crs}, x, w, m)$: *Takes a common reference string* crs*, a word* x*, a witness* w*, and a message* m *as input and outputs a signature* σ*.*
$\mathsf{Vrf}(\mathsf{crs}, x, m, \sigma)$: *Takes a common reference string* crs*, a word* x*, a message* m*, and a signature* σ *as input and outputs a bit* $b \in \{0, 1\}$*.*

We require signatures of knowledge to be correct, simulatable and f-extractable. We formally recall those notions below.

Definition 20 (Correctness). *A* SoK *w.r.t.* L_R *is correct, if there exists a negligible function* $\epsilon(\cdot)$ *such that for all* $x \in L_R$*, for all* w *such that* $(x, w) \in R$*, and for all* $m \in \mathcal{M}$ *it holds that*

$$\Pr\left[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa),\ \sigma \leftarrow \mathsf{Sign}(\mathsf{crs}, x, w, m) \ : \ \mathsf{Vrf}(\mathsf{crs}, x, m, \sigma) = 1\right] \geq 1 - \epsilon(\kappa).$$

Definition 21 (Simulatability). *A* SoK *w.r.t.* L_R *is simulatable, if there exists a PPT simulator* $\mathcal{S} = (\mathsf{SSetup}, \mathsf{SimSign})$ *such that for all PPT adversaries* $\mathcal{A}$ *there exists a negligible function* $\epsilon(\cdot)$ *such that it holds that*

$$\left| \begin{array}{l} \Pr\left[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa),\ \mathcal{A}^{\mathsf{Sign}(\mathsf{crs},\cdot,\cdot,\cdot)}(\mathsf{crs}) = 1\right] \ - \\ \Pr\left[(\mathsf{crs}, \tau) \leftarrow \mathsf{SSetup}(1^\kappa),\ \mathcal{A}^{\mathsf{Sim}(\mathsf{crs},\tau,\cdot,\cdot,\cdot)}(\mathsf{crs}) = 1\right] \end{array} \right| \leq \epsilon(\kappa),$$

where $\mathsf{Sim}(\mathsf{crs}, \tau, x, w, m) := \mathsf{SimSign}(\mathsf{crs}, \tau, x, m)$ *and* Sim *only responds if* $(x, w) \in R$*.*

Definition 22 (f-Extractability). *A* SoK *w.r.t. L_R is f-extractable, if in addition to $\mathcal{S}$ there exists a PPT extractor* Extract*, such that for all PPT adversaries $\mathcal{A}$ there exists a negligible function $\epsilon(\cdot)$ such that it holds that*

$$\Pr\left[\begin{array}{l}(\mathsf{crs},\tau) \leftarrow \mathsf{SSetup}(1^\kappa),\\ (x,m,\sigma) \leftarrow \mathcal{A}^{\mathsf{Sim}(\mathsf{crs},\tau,\cdot,\cdot,\cdot)}(\mathsf{crs}),\\ y \leftarrow \mathsf{Extract}(\mathsf{crs},\tau,x,m,\sigma)\end{array} : \begin{array}{r}\mathsf{Vrf}(\mathsf{crs},x,m,\sigma)=0\ \vee\\ (x,m,\sigma)\in Q^{\mathsf{Sim}}\ \vee\\ (\exists\ w:(x,w)\in R\ \wedge\\ y=f(w))\end{array}\right] \geq 1-\epsilon(\kappa),$$

where Q^{Sim} denotes the queries (resp. answers) of Sim*.*

We note that, as illustrated in [BCC+15], this notion is a generalization of the original extractability notion from [CL06] which implies the original extractability notion if f is the identity. In this case, we simply call the f-extractability property *extractability*. Analogous to [BCC+15], we require the used SoK to be at the same time extractable and *straight-line f-extractable* with respect to some f other than the identity, where straight-line as usual says that the extractor runs without rewinding the adversary [Fis05].

3 Dynamic Group Signatures

Subsequently, we recall the established model for dynamic group signatures. We follow Bellare et al. [BSZ05] (BSZ model), with the slight difference that we relax the perfect correctness to only require computational correctness. Furthermore, we also present the weaker anonymity notion of CPA-full anonymity from [BBS04] and the notion of opening soundness [SSE+12], which addresses issues regarding hijacking of signatures by malicious group members. In particular, we use the notion of weak opening soundness, where the opening authority is required to be honest, since we believe that this notion provides a good tradeoff between computational efficiency of potential instantiations and expected security guarantees (even the authors of [SSE+12] say that weak opening soundness already addresses the attacks they had in mind).

$\mathsf{GKGen}(1^\kappa)$: Takes a security parameter κ as input and outputs a triple $(\mathsf{gpk},\mathsf{ik},\mathsf{ok})$ containing the group public key gpk, the issuing key ik as well as the opening key ok.

$\mathsf{UKGen}(1^\kappa)$: Takes a security parameter κ as input and outputs a user key pair $(\mathsf{usk}_i,\mathsf{upk}_i)$.

$\mathsf{Join}(\mathsf{gpk},\mathsf{usk}_i,\mathsf{upk}_i)$: Takes the group public key gpk and the user's key pair $(\mathsf{usk}_i,\mathsf{upk}_i)$ as input. It interacts with the Issue algorithm and outputs the group signing key gsk_i of user i on success.

$\mathsf{Issue}(\mathsf{gpk},\mathsf{ik},i,\mathsf{upk}_i,\mathsf{reg})$: Takes the group public key gpk, the issuing key ik, the index i of a user, user i's public key upk_i, and the registration table reg as input. It interacts with the Join algorithm and adds an entry for user i in reg on success. In the end, it returns reg.

$\mathsf{Sign}(\mathsf{gpk},\mathsf{gsk}_i,m)$: Takes the group public key gpk, a group signing key gsk_i, and a message m as input and outputs a group signature σ.

$\mathsf{Vrf}(\mathsf{gpk}, m, \sigma)$: Takes the group public key gpk, a message m and a signature σ as input and outputs a bit $b \in \{0, 1\}$.

$\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathsf{reg}, m, \sigma)$: Takes the group public key gpk, the opening key ok, the registration table reg, a message m, and a valid signature σ on m under gpk as input. It extracts the identity of the signer and returns a pair (i, τ), where τ is a proof.

$\mathsf{Judge}(\mathsf{gpk}, m, \sigma, i, \mathsf{upk}_i, \tau)$: Takes the group public key gpk, a message m, a valid signature σ on m under gpk, an index i, user i's public key upk_i, and a proof τ. It returns a bit $b \in \{0, 1\}$.

Oracles. In the following we recall the definitions of the oracles required by the security model. We assume that the keys $(\mathsf{gpk}, \mathsf{ik}, \mathsf{ok})$ created in the experiments are implicitly available to the oracles. Furthermore, the environment maintains the sets $\mathtt{HU}, \mathtt{CU}$ of honest and corrupted users, the set $\mathtt{GS}$ of message-signature tuples returned by the challenge oracle, the lists $\mathsf{upk}, \mathsf{usk}, \mathsf{gsk}$ of user public keys, user private keys, and group signing keys. The list upk is publicly readable and the environment also maintains the registration table reg. Finally, $\mathtt{SI}$ represents a list that ensures the consistency of subsequent calls to CrptU and SndToI. All sets are initially empty and all list entries are initially set to $\perp$. In the context of lists, we use $\mathsf{upk}_i, \mathsf{usk}_i$, etc. as shorthand for $\mathsf{upk}[i], \mathsf{usk}[i]$, etc.

$\mathsf{AddU}(i)$: Takes an index i as input. If $i \in \mathtt{CU} \cup \mathtt{HU}$ it returns $\perp$. Otherwise it runs $(\mathsf{usk}_i, \mathsf{upk}_i) \leftarrow \mathsf{UKGen}(1^\kappa)$ and $(\mathsf{reg}, \mathsf{gsk}_i) \leftarrow$ $\langle \mathsf{Issue}(\mathsf{gpk}, \mathsf{ik}, i, \mathsf{upk}_i, \mathsf{reg}) \leftrightarrow \mathsf{Join}(\mathsf{gpk}, \mathsf{usk}_i, \mathsf{upk}_i) \rangle$.
Finally, it sets $\mathtt{HU} \leftarrow \mathtt{HU} \cup \{i\}$ and returns upk_i.

$\mathsf{CrptU}(i, \mathsf{upk}_j)$: Takes an index i and user public key upk_j as input. If $i \in \mathtt{CU} \cup \mathtt{HU}$ it returns $\perp$. Otherwise it sets $\mathtt{CU} \leftarrow \mathtt{CU} \cup \{i\}$, $\mathtt{SI}[i] \leftarrow \top$ and $\mathsf{upk}_i \leftarrow \mathsf{upk}_j$.

$\mathsf{SndToI}(i)$: Takes an index i as input. If $\mathtt{SI}[i] \neq \top$ it returns $\perp$. Otherwise, it plays the role of an honest issuer when interacting with the corrupted user i. More precisely, it runs $\mathsf{reg} \leftarrow \langle \mathsf{Issue}(\mathsf{gpk}, \mathsf{ik}, i, \mathsf{upk}_i, \mathsf{reg}) \leftrightarrow \mathcal{A} \rangle$. In the end it sets $\mathtt{SI}[i] \leftarrow \perp$.

$\mathsf{SndToU}(i)$: Takes an index i as input. If $i \notin \mathtt{HU}$ it sets $\mathtt{HU} \leftarrow \mathtt{HU} \cup \{i\}$, runs $(\mathsf{usk}_i, \mathsf{upk}_i) \leftarrow \mathsf{UKGen}(1^\kappa)$. Then it plays the role of the honest user i when interacting with a corrupted issuer. More precisely, it runs $\mathsf{gsk}_i \leftarrow \langle \mathcal{A} \leftrightarrow \mathsf{Join}(\mathsf{gpk}, \mathsf{usk}_i, \mathsf{upk}_i) \rangle$.

$\mathsf{USK}(i)$: Takes an index i as input and returns $(\mathsf{gsk}_i, \mathsf{usk}_i)$.

$\mathsf{RReg}(i)$: Takes an index i as input and returns reg_i.

$\mathsf{WReg}(i, \rho)$: Takes an index i and a registration table entry ρ as input and sets $\mathsf{reg}_i \leftarrow \rho$.

$\mathsf{GSig}(i, m)$: Takes an index i and a message m as input. If $i \notin \mathtt{HU}$ or $\mathsf{gsk}_i = \perp$ it returns $\perp$ and $\sigma \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}_i, m)$ otherwise.

$\mathsf{Ch}(b, i_0, i_1, m)$: Takes a bit b, two indexes i_0 and i_1, and a message m as input. If $\{i_0, i_1\} \not\subseteq \mathtt{HU} \vee \mathsf{gsk}_{i_0} = \perp \vee \mathsf{gsk}_{i_1} = \perp$ it returns $\perp$. Otherwise, it computes $\sigma \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}_{i_b}, m)$, sets $\mathtt{GS} \leftarrow \mathtt{GS} \cup \{(m, \sigma)\}$ and returns σ.

$\mathsf{Open}(m, \sigma)$: Takes a message m and a signature σ as input. If $(m, \sigma) \in \mathtt{GS}$ or $\mathsf{Vrf}(\mathsf{gpk}, m, \sigma) = 0$ it returns $\perp$. Otherwise, it returns $(i, \tau) \leftarrow \mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathsf{reg}, m, \sigma)$.

Security Notions. We require dynamic group signatures to be correct, anonymous, traceable, non-frameable, and weakly opening sound. Correctness, requires that everything works correctly if everyone behaves honestly. Note that we relax perfect correctness to computational correctness.

Definition 23 (Correctness). *A* GSS *is correct, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that*

$$\Pr\left[\begin{array}{l}(\mathsf{gpk},\mathsf{ik},\mathsf{ok}) \leftarrow \mathsf{GKGen}(1^\kappa),\\ \mathcal{O} \leftarrow \{\mathsf{AddU}(\cdot),\mathsf{RReg}(\cdot)\},\\ (i,m) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{gpk}),\\ \sigma \leftarrow \mathsf{Sign}(\mathsf{gpk},\mathsf{gsk}_i,m),\\ (j,\tau) \leftarrow \mathsf{Open}(\mathsf{gpk},\mathsf{ok},\mathsf{reg},m,\sigma)\end{array} : \begin{array}{r}\mathsf{Vrf}(\mathsf{gpk},m,\sigma)=1 \ \wedge \ i \in \mathtt{HU}\\ \wedge \ \mathsf{gsk}_i \neq \bot \ \wedge \ i=j \ \wedge\\ \mathsf{Judge}(\mathsf{gpk},m,\sigma,i,\mathsf{upk}_i,\tau)=1\end{array}\right] \geq 1-\epsilon(\kappa).$$

Anonymity captures the intuition that group signers remain anonymous for everyone except the opening authority. Thereby, the adversary can see arbitrary key exposures. Furthermore, in the $\mathsf{CCA2}$ case the adversary can even request arbitrary openings of other group signatures.

Definition 24 (T-Full Anonymity). *Let* $\mathsf{T} \in \{\mathsf{CPA},\mathsf{CCA2}\}$*. A* GSS *is* T*-fully anonymous, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that*

$$\Pr\left[\begin{array}{l}(\mathsf{gpk},\mathsf{ik},\mathsf{ok}) \leftarrow \mathsf{GKGen}(1^\kappa),\\ b \xleftarrow{R} \{0,1\}, \ b^* \leftarrow \mathcal{A}^{\mathcal{O}_\mathsf{T}}(\mathsf{gpk},\mathsf{ik})\end{array} : b=b^*\right] \leq 1/2+\epsilon(\kappa),$$

where

$$\mathcal{O}_\mathsf{T} \leftarrow \begin{cases}\left\{\begin{array}{l}\mathsf{Ch}(b,\cdot,\cdot,\cdot), \ \mathsf{SndToU}(\cdot),\mathsf{WReg}(\cdot,\cdot),\\ \mathsf{USK}(\cdot), \ \mathsf{CrptU}(\cdot,\cdot)\end{array}\right\} & \textit{if } \ \mathsf{T}=\mathsf{CPA}, \ \textit{and}\\[2ex] \left\{\begin{array}{l}\mathsf{Ch}(b,\cdot,\cdot,\cdot), \ \mathsf{Open}(\cdot,\cdot), \ \mathsf{SndToU}(\cdot),\\ \mathsf{WReg}(\cdot,\cdot), \ \mathsf{USK}(\cdot), \ \mathsf{CrptU}(\cdot,\cdot)\end{array}\right\} & \textit{if } \ \mathsf{T}=\mathsf{CCA2}.\end{cases}$$

Traceability models the requirement that, as long as the issuer behaves honestly and its secret key remains secret, every valid signature can be traced back to a user. This must even hold if the opening authority colludes with malicious users.

Definition 25 (Traceability). *A* GSS *is traceable, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that*

$$\Pr\left[\begin{array}{l}(\mathsf{gpk},\mathsf{ik},\mathsf{ok}) \leftarrow \mathsf{GKGen}(1^\kappa),\\ \mathcal{O} \leftarrow \{\mathsf{SndToI}(\cdot), \ \mathsf{AddU}(\cdot),\\ \mathsf{RReg}(\cdot), \ \mathsf{USK}(\cdot), \ \mathsf{CrptU}(\cdot)\},\\ (m,\sigma) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{gpk},\mathsf{ok}),\\ (i,\tau) \leftarrow \mathsf{Open}(\mathsf{gpk},\mathsf{ok},\mathsf{reg},m,\sigma)\end{array} : \begin{array}{r}\mathsf{Vrf}(\mathsf{gpk},m,\sigma)=1 \ \wedge\\ (i=\bot \ \vee\\ \mathsf{Judge}(\mathsf{gpk},m,\sigma,i,\mathsf{upk}_i,\tau)=0)\end{array}\right] \leq \epsilon(\kappa).$$

Non-frameability requires that no one can forge signatures for honest users. This must even hold if the issuing authority, the opening authority, and, other malicious users collude.

Definition 26 (Non-Frameability). *A* GSS *is non-frameable, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that*

$$\Pr\left[\begin{array}{l}(\mathsf{gpk},\mathsf{ik},\mathsf{ok}) \leftarrow \mathsf{GKGen}(1^\kappa),\\ \mathcal{O} \leftarrow \{\mathsf{SndToU}(\cdot),\ \mathsf{WReg}(\cdot,\cdot),\\ \mathsf{GSig}(\cdot,\cdot),\ \mathsf{USK}(\cdot),\ \mathsf{CrptU}(\cdot)\},\\ (m,\sigma,i,\tau) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{gpk},\mathsf{ok},\mathsf{ik})\end{array} : \begin{array}{r}\mathsf{Vrf}(\mathsf{gpk},m,\sigma)=1\ \wedge\\ i \in \mathtt{HU}\ \wedge\ \mathsf{gsk}_i \neq \perp\ \wedge\\ i \notin \mathtt{USK}\ \wedge\ (i,m) \notin \mathtt{SIG}\ \wedge\\ \mathsf{Judge}(\mathsf{gpk},m,\sigma,i,\mathsf{upk}_i,\tau)=1\end{array}\right] \leq \epsilon(\kappa),$$

where $\mathtt{USK}$ *and* $\mathtt{SIG}$ *denote the queries to the oracles* USK *and* Sign*, respectively.*

Weak opening soundness [SSE+12] essentially requires that no malicious user can claim ownership of a signature issued by an honest user, as long as the opening authority behaves honestly.

Definition 27 (Weak Opening Soundness). *A* GSS *is weakly opening sound, if for all PPT adversaries* $\mathcal{A}$ *there is a negligible function* $\epsilon(\cdot)$ *such that*

$$\Pr\left[\begin{array}{l}(\mathsf{gpk},\mathsf{ik},\mathsf{ok}) \leftarrow \mathsf{GKGen}(1^\kappa),\\ \mathcal{O} \leftarrow \{\mathsf{AddU}(\cdot)\},\\ (m,i,j,\mathsf{st}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{gpk}),\\ \sigma \leftarrow \mathsf{Sign}(\mathsf{gpk},\mathsf{gsk}_i,m),\\ \tau \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{st},\sigma,\mathsf{gsk}_j)\end{array} : \begin{array}{r} i \neq j\ \wedge\ \{i,j\} \subseteq \mathtt{HU}\ \wedge\\ \mathsf{Judge}(\mathsf{gpk},m,\sigma,j,\mathsf{upk}_j,\tau)=1\end{array}\right] \leq \epsilon(\kappa).$$

4 Construction

Our idea is inspired by [HS14], who use the "unlinkability" feature of $\mathsf{SPS\text{-}EQ}$ signatures to construct anonymous credentials. Essentially, a credential in their approach represents a signature for an equivalence class and to show a credential they always present a newly re-randomized signature to a random representative of this class. While, due to the intuitive relation of anonymous credentials and group signatures, it might seem straightforward to map this idea to group signatures, it turns out that there are various subtle, yet challenging issues which we need to solve.

First, the anonymity notion is much stronger than the one of anonymous credentials (see, e.g., [FHS18]) in that it does not put many restrictions on the Ch and the USK oracles. In particular, Ch can be called an arbitrary number of times and USK can be called for all users. Thus, the user secret keys must be of a form so that it is possible to embed decision problem instances into them upon simulation, while not influencing their distribution (as the adversary sees those keys and would be able to detect the simulation otherwise). More precisely, anonymity in our paradigm seems to require that the user keys contain no $\mathbb{Z}_p$ elements, which, in turn, renders the non-frameability proof more difficult. Second, if $\mathsf{CCA2}$-full anonymity is required, the simulatability of the open oracle needs to be ensured, while the reduction must not be aware of the opening information (as otherwise the reduction could trivially break anonymity on its own and would be meaningless). This seems to crucially require a proof system providing rather strong extractability properties. To maintain efficiency, it is important to find the mildest possible requirement which still allows the security proofs to work out. Third, the non-frameability adversary is given the issuing key as well as the opening key. Thus, the reduction must be able to simulate the whole join process without knowledge of a user secret key in a way that the distribution change is not even detectable with the knowledge of these keys.

Now, before we present our full construction, we briefly revisit our basic idea. In our scheme, each group member chooses a secret vector $(R, P) \in (\mathbb{G}_1^*)^2$ representing an equivalence class where the second component P is identical for all users. When joining the group, a blinded version $q \cdot (R, P)$ with $q \xleftarrow{R} \mathbb{Z}_p^*$ of this vector, i.e., another representative of the class, is signed by the issuer using an $\mathsf{SPS\text{-}EQ}$, and, by the re-randomization property of $\mathsf{SPS\text{-}EQ}$ and the feature to publicly change representatives of classes, the user thus obtains a signature on the unblinded key (R, P) using $\mathsf{ChgRep}_{\mathcal{R}}$ with q^{-1}. To provide a means to open signatures, a user additionally has to provide an encryption of a value $\hat{R} \in \mathbb{G}_2$ such that $e(R, \hat{P}) = e(P, \hat{R})$ on joining (and has to sign the ciphertext as an identity proof). The group signing key of the user is then the pair consisting of the vector (R, P) and the $\mathsf{SPS\text{-}EQ}$ signature on this vector. A group member can sign a message m on behalf of the group by randomizing its group signing key and computing a signature of knowledge (SoK) to the message m proving knowledge of the used randomizer.[4] The group signature is then the randomized group signing key and the SoK.

Very roughly, a signer remains anonymous since it is infeasible to distinguish two randomized user secret keys under DDH in $\mathbb{G}_1$. The unforgeability of $\mathsf{SPS\text{-}EQ}$ ensures that each valid signature can be opened. Furthermore, it is hard to forge signatures of honest group members since it is hard to unblind a user secret key under co-CDHI and the signature of knowledge essentially ensures that we can extract such an unblinded user secret key from a successful adversary.

Detailed Construction. We require zero-knowledge proofs upon Join and Open. The **NP** relation R_{J} corresponding to the proof carried out in Join is defined as

$$\begin{aligned}((U_i, Q, \hat{C}_{\mathsf{J}_i}, \mathsf{pk}_{\mathsf{O}}), (r, \omega)) \in R_{\mathsf{J}} \iff \\ \hat{C}_{\mathsf{J}_i} = \Omega.\mathsf{Enc}(\mathsf{pk}_{\mathsf{O}}, r\hat{P};\ \omega) \ \wedge\ U_i = r \cdot Q.\end{aligned}$$

Essentially, the NP language which is associated to this relation consists of all tuples where U_i and the group element $r\hat{P}$, encrypted within C_{J_i} share the same discrete logarithm r with respect to bases Q and $\hat{P}$, respectively. The **NP** relation R_{O} corresponding to the proof carried out upon Open is

$$\begin{aligned}((\hat{C}_{\mathsf{J}_i}, \mathsf{pk}_{\mathsf{O}}, \sigma), (\mathsf{sk}_{\mathsf{O}}, \hat{R})) \in R_{\mathsf{O}} \iff \hat{R} = \Omega.\mathsf{Dec}(\mathsf{sk}_{\mathsf{O}}, \\ \hat{C}_{\mathsf{J}_i}) \ \wedge\ \mathsf{pk}_{\mathsf{O}} \equiv \mathsf{sk}_{\mathsf{O}} \ \wedge\ e(\sigma_1[1][1], \hat{P}) = e(\sigma_1[1][2], \hat{R}).\end{aligned}$$

[4] For technical reasons and in particular for extractability, we actually require a signature of knowledge for message $m' = \sigma_1||m$, where σ_1 contains the re-randomized user secret key and $\mathsf{SPS\text{-}EQ}$ signature.

Thereby, $\mathsf{pk} \equiv \mathsf{sk}$ denotes the consistency of pk and sk. Note that σ_1 represents the randomized user secret key, i.e., is of the form $((\rho R, \rho P), \sigma')$ and consists of a randomized message vector and a corresponding randomized $\mathsf{SPS\text{-}EQ}$ signature. We use $\sigma_1[1][j]$ to refer to the jth element in the (randomized) message vector. Essentially, the NP language associated to this relation consists of all tuples where the element $\hat{R}$ being encrypted in C_{J_i} and the element $\sigma_1[1][1]$ share the same discrete logarithms with respect to bases $\hat{P}$ and $\sigma_2[1][2]$, respectively.

Furthermore, upon Sign we require a signature of knowledge which is with respect to the following **NP** relation R_{S}.

$$((P, Q), \rho) \in R_{\mathsf{S}} \iff Q = \rho \cdot P.$$

For the sake of compactness, we assume that the languages defined by R_{J}, R_{O}, R_{S} are implicit in the CRSs $\mathsf{crs}_{\mathsf{J}}$, $\mathsf{crs}_{\mathsf{O}}$, and $\mathsf{crs}_{\mathsf{S}}$, respectively. The full construction is presented as Scheme 1. Note that if multiple users collude and use the same value r upon $\mathsf{Join}^{(1)}$, we always return the first user who registered with this particular value r in Open. Then, Open always returns the signer who initiated the collusion by sharing the r value, which, we think, is the most reasonable choice. Note that this is in line with the BSZ model: traceability only requires that every valid signature can be opened, while not requiring that it opens to one particular user out of the set of colluding users; correctness and non-frameability are defined for honest users and are therefore clearly not influenced.

4.1 Security

First, note that our $\mathsf{Join} \leftrightarrow \mathsf{Issue}$ protocol is inherently concurrently secure: we only have two moves which means that interleaving different $\mathsf{Join} \leftrightarrow \mathsf{Issue}$ will not be accepted as valid $\mathsf{Join} \leftrightarrow \mathsf{Issue}$. Since the simplicity of a two-move protocol makes it rather hard to see from the proof that we actually consider concurrent security, we explicitly stress it here to make sure that this additional feature is not overlooked.

In our proofs, we omit to make the negligible distribution switches which arise when sampling uniformly random from $\mathbb{Z}_p$ instead of $\mathbb{Z}_p^*$ explicit and instead treat them as conceptual changes for the sake of compactness.

Theorem 1. *If* $\mathsf{SPS\text{-}EQ}$ *is correct,* SoK *is correct, and* Π *is sound, then Scheme 1 is correct.*

Proof (Proof (Sketch)). Correctness is straight forward to verify by inspection. We only have to take care of one detail: There is the possibility that two honest executions of AddU yield the same value r (which is chosen uniformly at random upon $\mathsf{Join}^{(1)}$). Thus, the probability of two colliding r is negligible.

Theorem 2. *If* Π *is adaptively zero-knowledge,* SoK *is simulatable,* Ω *is* $\mathsf{IND\text{-}CPA}$ *secure,* $\mathsf{SPS\text{-}EQ}$ *perfectly adapts signatures, and the DDH assumption holds in* $\mathbb{G}_1$*, then Scheme 1 is* CPA*-full anonymous.*

$\mathsf{GKGen}(1^\kappa)$: Run $\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$, $(\mathsf{sk}_{\mathcal{R}}, \mathsf{pk}_{\mathcal{R}}) \leftarrow \mathsf{KGen}_{\mathcal{R}}(\mathsf{BG}, 2)$, $(\mathsf{sk}_{\mathsf{O}}, \mathsf{pk}_{\mathsf{O}}) \leftarrow \Omega.\mathsf{KGen}(1^\kappa)$, $\mathsf{crs}_{\mathsf{J}} \leftarrow \Pi.\mathsf{Setup}(1^\kappa)$, $\mathsf{crs}_{\mathsf{O}} \leftarrow \Pi.\mathsf{Setup}(1^\kappa)$, $\mathsf{crs}_{\mathsf{S}} \leftarrow \mathsf{SoK}.\mathsf{Setup}(1^\kappa)$, set $\mathsf{gpk} \leftarrow (\mathsf{pk}_{\mathcal{R}}, \mathsf{pk}_{\mathsf{O}}, \mathsf{crs}_{\mathsf{J}}, \mathsf{crs}_{\mathsf{O}}, \mathsf{crs}_{\mathsf{S}})$, $\mathsf{ik} \leftarrow \mathsf{sk}_{\mathcal{R}}$, $\mathsf{ok} \leftarrow \mathsf{sk}_{\mathsf{O}}$ and return $(\mathsf{gpk}, \mathsf{ik}, \mathsf{ok})$.

$\mathsf{UKGen}(1^\kappa)$: Return $(\mathsf{usk}_i, \mathsf{upk}_i) \leftarrow \Sigma.\mathsf{KGen}(1^\kappa)$.

$\mathsf{Join}^{(1)}(\mathsf{gpk}, \mathsf{usk}_i, \mathsf{upk}_i)$: Choose $q, r \xleftarrow{R} \mathbb{Z}_p^*$, set $(U_i, Q) \leftarrow (r \cdot qP, qP)$, and output $M_{\mathsf{J}} \leftarrow ((U_i, Q), \hat{C}_{\mathsf{J}_i}, \sigma_{\mathsf{J}_i}, \pi_{\mathsf{J}_i})$ and $\mathsf{st} \leftarrow (\mathsf{gpk}, q, U_i, Q)$, where

$$\hat{C}_{\mathsf{J}_i} \leftarrow \Omega.\mathsf{Enc}(\mathsf{pk}_{\mathsf{O}}, r\hat{P};\ \omega),\ \ \sigma_{\mathsf{J}_i} \leftarrow \Sigma.\mathsf{Sign}(\mathsf{usk}_i, \hat{C}_{\mathsf{J}_i}),$$
$$\pi_{\mathsf{J}_i} \leftarrow \Pi.\mathsf{Proof}(\mathsf{crs}_{\mathsf{J}}, (U_i, Q, \hat{C}_{\mathsf{J}_i}, \mathsf{pk}_{\mathsf{O}}), (r, \omega)).$$

$\mathsf{Issue}(\mathsf{gpk}, \mathsf{ik}, i, \mathsf{upk}_i, \mathsf{reg})$: Receive $M_{\mathsf{J}} = ((U_i, Q), \hat{C}_{\mathsf{J}_i}, \sigma_{\mathsf{J}_i}, \pi_{\mathsf{J}_i})$, return reg and send σ' to user i, where

$$\mathsf{reg}_i \leftarrow (\hat{C}_{\mathsf{J}_i}, \sigma_{\mathsf{J}_i}),\ \ \sigma' \leftarrow \mathsf{Sign}_{\mathcal{R}}((U_i, Q), \mathsf{sk}_{\mathcal{R}}),$$

if $\Pi.\mathsf{Vrf}(\mathsf{crs}_{\mathsf{J}}, (U_i, Q, \hat{C}_{\mathsf{J}_i}, \mathsf{pk}_{\mathsf{O}}), \pi_{\mathsf{J}_i}) = 1 \ \wedge\ \Sigma.\mathsf{Vrf}(\mathsf{upk}_i, \hat{C}_{\mathsf{J}_i}, \sigma_{\mathsf{J}_i}) = 1$, and return $\perp$ otherwise.

$\mathsf{Join}^{(2)}(\mathsf{st}, \sigma')$: Parse st as $(\mathsf{gpk}, q, U_i, Q)$ and return gsk_i, where

$$\mathsf{gsk}_i = ((rP, P), \sigma) \leftarrow \mathsf{ChgRep}_{\mathcal{R}}((U_i, Q), \sigma', q^{-1}, \mathsf{pk}_{\mathcal{R}}),$$

if $\mathsf{Vrf}_{\mathcal{R}}((U_i, Q), \sigma', \mathsf{pk}_{\mathcal{R}}) = 1$, and return $\perp$ otherwise.

$\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}_i, m)$: Choose $\rho \xleftarrow{R} \mathbb{Z}_p^*$, and return $\sigma \leftarrow (\sigma_1, \sigma_2)$, where

$$\sigma_1 \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(\mathsf{gsk}_i, \rho, \mathsf{pk}_{\mathcal{R}}),^{a}\ \sigma_2 \leftarrow \mathsf{SoK}.\mathsf{Sign}(\mathsf{crs}_{\mathsf{S}}, (P, \sigma_1[1][2]), \rho, \sigma_1 || m).$$

$\mathsf{Vrf}(\mathsf{gpk}, m, \sigma)$: Return 1 if the following holds, and 0 otherwise:

$$\mathsf{Vrf}_{\mathcal{R}}(\sigma_1, \mathsf{pk}_{\mathcal{R}}) = 1 \quad \wedge \quad \mathsf{SoK}.\mathsf{Vrf}(\mathsf{crs}_{\mathsf{S}}, (P, \sigma_1[1][2]), \sigma_1 || m, \sigma_2) = 1.$$

$\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathsf{reg}, m, \sigma)$: Parse σ as (σ_1, σ_2), and ok as sk_{O}. Obtain the lowest index i,[b] so that it holds for $(\hat{C}_{\mathsf{J}_i}, \sigma_{\mathsf{J}_i}) \leftarrow \mathsf{reg}_i$ that $\hat{R} \leftarrow \Omega.\mathsf{Dec}(\mathsf{sk}_{\mathsf{O}}, \hat{C}_{\mathsf{J}_i})$ and $e(\sigma_1[1][1], \hat{P}) = e(\sigma_1[1][2], \hat{R})$. Return (i, τ) and $\perp$ if no such entry exists, where

$$\tau \leftarrow (\pi_{\mathsf{O}}, \hat{C}_{\mathsf{J}_i}, \sigma_{\mathsf{J}_i}),\ \text{and}\ \pi_{\mathsf{O}} \leftarrow \Pi.\mathsf{Proof}(\mathsf{crs}_{\mathsf{O}}, (\hat{C}_{\mathsf{J}_i}, \mathsf{pk}_{\mathsf{O}}, \sigma), (\mathsf{sk}_{\mathsf{O}}, \hat{R})).$$

$\mathsf{Judge}(\mathsf{gpk}, m, \sigma, i, \mathsf{upk}_i, \tau)$: Parse τ as $(\pi_{\mathsf{O}}, \hat{C}_{\mathsf{J}_i}, \sigma_{\mathsf{J}_i})$, and return 1 if the following holds and 0 otherwise:

$$\Sigma.\mathsf{Vrf}(\mathsf{upk}_i, \hat{C}_{\mathsf{J}_i}, \sigma_{\mathsf{J}_i}) = 1 \quad \wedge \quad \Pi.\mathsf{Vrf}(\mathsf{crs}_{\mathsf{O}}, (\hat{C}_{\mathsf{J}_i}, \mathsf{pk}_{\mathsf{O}}, \sigma), \pi_{\mathsf{O}}) = 1.$$

[a] Note that gsk_i is of the form $((R, P), \sigma)$ and σ_1 is a randomization of gsk_i. We slightly abuse the notation of $\mathsf{Vrf}_{\mathcal{R}}$ and $\mathsf{ChgRep}_{\mathcal{R}}$ and input message-signature tuples instead of separately inputting messages and signatures.

[b] We assume that the indexes are in ascending order w.r.t. the time of registration.

Scheme 1: Fully-Anonymous Dynamic Group Signature Scheme

Theorem 3. *If* Π *is adaptively zero-knowledge,* SoK *is simulatable and straight-line* f*-extractable, where* $f : \mathbb{Z}_p \to \mathbb{G}_2$ *is defined as* $r \mapsto r \cdot \hat{P}$*,* Ω *is* IND-CCA2 *secure,* SPS-EQ *perfectly adapts signatures, and the DDH assumption holds in* $\mathbb{G}_1$*, then Scheme 1 is* CCA2*-full anonymous.*

Proof. We prove Theorem 2 and 3 by showing that the output distributions of the Ch oracle are (computationally) independent of the bit b, where we highlight the parts of the proof which are specific to Theorem 3 and can be omitted to prove Theorem 2. Therefore, let $q_{\mathsf{Ch}} \leq \mathsf{poly}(\kappa)$ be the number of queries to Ch, $q_{\mathsf{O}} \leq \mathsf{poly}(\kappa)$ be the number of queries to Open, and $q_{\mathsf{SndToU}} \leq \mathsf{poly}(\kappa)$ be the number of queries to SndToU.

Game 0: The original anonymity game.

Game 1: As Game 0, but we run $(\mathsf{crs_J}, \tau_{\mathsf{J}}) \leftarrow \Pi.\mathcal{S}_1(1^\kappa)$ instead of $\mathsf{crs_J} \leftarrow \Pi.\mathsf{Setup}(1^\kappa)$ upon running GKGen and store the trapdoor τ_{J}. Then, we simulate all calls to Π.Proof executed in Join using the simulator (without a witness).

Transition - Game 0 → Game 1: A distinguisher $\mathcal{D}^{0\to1}$ is an adversary against adaptive zero-knowledge of Π, and, therefore, the probability to distinguish Game 0 and Game 1 is negligible, i.e., $|\Pr[S_1] - \Pr[S_0]| \leq \epsilon_{\mathsf{ZK_J}}(\kappa)$.

Game 2: As Game 1, but we run $(\mathsf{crs_O}, \tau_{\mathsf{O}}) \leftarrow \Pi.\mathcal{S}_1(1^\kappa)$ instead of $\mathsf{crs_O} \leftarrow \Pi.\mathsf{Setup}(1^\kappa)$ upon running GKGen and store the trapdoor τ_{O}. Then, we simulate all calls to Π.Proof in Open using the simulator (without a witness).

Transition - Game 1 → Game 2: A distinguisher $\mathcal{D}^{1\to2}$ is an adversary against adaptive zero-knowledge of Π, and, therefore, the probability to distinguish Game 1 and Game 2 is negligible, i.e., $|\Pr[S_2] - \Pr[S_1]| \leq \epsilon_{\mathsf{ZK_O}}(\kappa)$.

Game 3: As Game 2, but we run $(\mathsf{crs_S}, \tau_{\mathsf{S}}) \leftarrow \mathsf{SoK.SSetup}(1^\kappa)$ instead of $\mathsf{crs_S} \leftarrow \mathsf{SoK.Setup}(1^\kappa)$ upon running GKGen and store the trapdoor τ_{S}. Then we simulate all calls to SoK.Sign using the simulator (without a witness).

Transition - Game 2 → Game 3: A distinguisher $\mathcal{D}^{2\to3}$ is an adversary against simulatability of SoK. Therefore, the distinguishing probability is negligible, i.e., $|\Pr[S_3] - \Pr[S_2]| \leq \epsilon_{\mathsf{SIM}}(\kappa)$.

Game 4: As Game 3, but instead of $(\mathsf{sk_O}, \mathsf{pk_O}) \leftarrow \Omega.\mathsf{KGen}(1^\kappa)$ in GKGen, we obtain $\mathsf{pk_O}$ from an IND-CPA (resp. IND-CCA2) challenger and set $\mathsf{sk_O} \leftarrow \perp$.

In the CCA2 case, we additionally maintain secret lists AU and OI, and upon each call to the SndToU oracle we store $\mathtt{AU}[i] \leftarrow (\mathsf{gsk}_i, \hat{C}_{\mathsf{J}_i}) = (((R, P), \sigma), \hat{C}_{\mathsf{J}_i})$. Then, we simulate the WReg oracle as follows

$\mathsf{WReg}(i, \rho)$: As the original oracle, but we additionally parse ρ as $(\hat{C}_{\mathsf{J}_i}, \sigma_{\mathsf{J}_i})$. If there exists an index j so that $\mathtt{AU}[j][2] = \hat{C}_{\mathsf{J}_i}$, we parse $\mathtt{AU}[j][1]$ as $((R, P), \sigma)$ and set $\mathtt{OI}[i] \leftarrow (R, \perp)$. If there exists no such index, we obtain $\hat{R}$ using the decryption oracle and set $\mathtt{OI}[i] \leftarrow (\perp, \hat{R})$.

Furthermore, we simulate the Open algorithm within the Open oracle as follows.

$\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathsf{reg}, m, \sigma)$: First, obtain $\hat{\Psi} = \rho\hat{P}$ using the straight-line f-extractor. Then, obtain the lowest index i where either $e(\sigma_1[1][1], \hat{P}) = e(\sigma_1[1][2], \mathtt{OI}[i][2])$ holds, or $e(\mathtt{OI}[i][1], \hat{\Psi}) = e(\sigma_1[1][1], \hat{P})$ holds. Compute a simulated proof τ and return (i, τ) and $\perp$ if no such index exists.

If the extractor fails at some point, we choose $b \xleftarrow{R} \{0,1\}$ and return b.

Transition - Game 3 → Game 4 (CPA): In the CPA case, we do not have to simulate the open oracle, and we only obtain the opening key from an IND-CPA challenger. Thus, this change is conceptual, i.e., $\Pr[S_3] = \Pr[S_4]$.

Transition - Game 3 → Game 4 (CCA2): By the straight-line f-extractability of the SoK, one can extract a witness ρ in every call to Open with overwhelming probability $1 - \epsilon_{\mathsf{EXT}}(\kappa)$. Thus, both games proceed identically unless the extraction fails, i.e., $|\Pr[S_3] - \Pr[S_4]| \leq q_{\mathsf{O}} \cdot \epsilon_{\mathsf{EXT}}(\kappa)$.

Game 5: As Game 4, but we compute the ciphertext $\hat{C}_{\mathsf{J}_i}$ in the Join algorithm (executed within the SndToU oracle) as $\hat{C}_{\mathsf{J}_i} \leftarrow \Omega.\mathsf{Enc}(\mathsf{pk}, \hat{P})$, i.e., with a constant message that is independent of the user.

Transition - Game 4 → Game 5: A distinguisher $\mathcal{D}^{4\rightarrow 5}$ is a distinguisher for the IND-CPA (resp. IND-CCA2) game of Ω. That is, $|\Pr[S_5] - \Pr[S_4]| \leq q_{\mathsf{SndToU}} \cdot \epsilon_{\mathsf{CPA}}(\kappa)$ (resp. $|\Pr[S_5] - \Pr[S_4]| \leq q_{\mathsf{SndToU}} \cdot \epsilon_{\mathsf{CCA2}}(\kappa)$).[5]

Game 6: As Game 5, but we re-add $\mathsf{sk_O}$, i.e., we again obtain $(\mathsf{sk_O}, \mathsf{pk_O}) \leftarrow \Omega.\mathsf{KGen}(1^\kappa)$. In the CCA2 case, we again decrypt ourselves with in the WReg simulation instead of using the decryption oracle.

Transition - Game 5 → Game 6: This change is conceptual. That is, $\Pr[S_5] = \Pr[S_6]$.

Game 7: As Game 6, but all calls to $\mathsf{ChgRep}_{\mathcal{R}}(M, \rho, \mathsf{pk}_{\mathcal{R}})$ are replaced by $\mathsf{Sign}_{\mathcal{R}}(\rho \cdot M, \mathsf{sk}_{\mathcal{R}})$.

Transition - Game 6 → Game 7: Under perfect adaption of signatures, the output distributions in Game 6 and Game 7 are identical, i.e., $\Pr[S_7] = \Pr[S_6]$.

Game 8: As Game 7, but we modify the Ch oracle as follows. Instead of running $\sigma_1 \leftarrow \mathsf{Sign}_{\mathcal{R}}(\rho \cdot \mathsf{gsk}_{i_b}[1], \mathsf{sk}_{\mathcal{R}})$, we choose $S, T \xleftarrow{R} \mathbb{G}_1$, and compute $\sigma_1 \leftarrow \mathsf{Sign}_{\mathcal{R}}((T, S), \mathsf{sk}_{\mathcal{R}})$.

Transition - Game 7 → Game 8: We claim $|\Pr[S_7] - \Pr[S_8]| \leq q_{\mathsf{Ch}} \cdot \epsilon_{\mathsf{DDH}}(\kappa)$. We prove this separately below.

In Game 8, the simulation is independent of the bit b, i.e., $\Pr[S_8] = 1/2$; what remains is to obtain a bound on the success probability in Game 0. In the CPA case, we have that $\Pr[S_0] \leq 1/2 + q_{\mathsf{SndToU}} \cdot \epsilon_{\mathsf{CPA}}(\kappa) + q_{\mathsf{Ch}} \cdot \epsilon_{\mathsf{DDH}}(\kappa) + \epsilon_{\mathsf{ZK_J}}(\kappa) + \epsilon_{\mathsf{ZK_O}}(\kappa) + \epsilon_{\mathsf{SIM}}(\kappa)$, which proves Theorem 2. In the CCA2 case, we have that $\Pr[S_0] \leq 1/2 + q_{\mathsf{SndToU}} \cdot \epsilon_{\mathsf{CCA2}}(\kappa) + q_{\mathsf{Ch}} \cdot \epsilon_{\mathsf{DDH}}(\kappa) + \epsilon_{\mathsf{ZK_J}}(\kappa) + \epsilon_{\mathsf{ZK_O}}(\kappa) + \epsilon_{\mathsf{SIM}}(\kappa) + q_{\mathsf{O}} \cdot \epsilon_{\mathsf{EXT}}(\kappa)$, which proves Theorem 3.

[5] For compactness, we collapsed the q_{SndToU} game changes into a single game change and note that one can straight forwardly unroll this to q_{SndToU} game changes where a single ciphertext is exchanged in each game.

Proof (Proof (of Claim)). Below we will show that Game 7 and Game 8 are indistinguishable by introducing further intermediate hybrid games.

Game 7_1: As Game 7, but we introduce a conceptual change which will make the subsequent distribution changes easier to follow. In particular upon each SndToU, we modify the simulation of Join so that we no longer choose $r \xleftarrow{R} \mathbb{Z}_p$ to obtain $(U_i, Q) \leftarrow (r \cdot qP, qP)$, but choose $R \xleftarrow{R} \mathbb{G}_1$ and obtain $(U_i, Q) \leftarrow (qR, qP)$.

Transition - Game 7 → Game 7_1: This is a conceptual change, i.e., $\Pr[S_7] = \Pr[S_{7_1}]$. Observe that we do not need to know r, as the proofs upon Join are simulated without a witness. Also the user secret keys $\mathsf{gsk}_i = ((R, P), \sigma)$ are exactly the same as honest secret keys.

Game 7_j $(2 \leq j \leq q_{\mathsf{Ch}} + 1)$: As Game 7_1, but we modify the Ch oracle as follows. For the first $j-1$ queries, instead of running $\sigma_1 \leftarrow \mathsf{Sign}_{\mathcal{R}}(\rho \cdot \mathsf{gsk}_{i_b}[1], \mathsf{sk}_{\mathcal{R}})$, we choose $S, T \xleftarrow{R} \mathbb{G}_1$, and compute $\sigma_1 \leftarrow \mathsf{Sign}_{\mathcal{R}}((T, S), \mathsf{sk}_{\mathcal{R}})$.

Transition - $7_j \to 7_{j+1}$ $(1 \leq j \leq q_{\mathsf{Ch}} + 1)$: For each transition, we present a hybrid game, which uses a DDH challenger to interpolate between Game 7_j and Game 7_{j+1}. First, we obtain a DDH instance $(aP, bP, cP) \in \mathbb{G}_1^3$ relative to BG. Then we proceed as follows:

- Upon each SndToU, we modify the simulation of Join as follows. Let i be the index of the user to join. We use the random self reducibility of DDH to obtain an independent DDH instance $(R_i, S_i, T_i) \xleftarrow{RSR} (aP, bP, cP)$ and set $\mathtt{CH}[i] \leftarrow (R_i, S_i, T_i)$. Then, we let $(U_i, Q) \leftarrow (qR_i, qP)$.
- Up to the $j-1$th query to Ch (i.e., for all queries where the answers are already random in Game 7_j), we compute σ_1 by choosing $S, T \xleftarrow{R} \mathbb{G}_1$, and compute $\sigma_1 \leftarrow \mathsf{Sign}_{\mathcal{R}}((T, S), \mathsf{sk}_{\mathcal{R}})$.
- Upon the jth query to Ch, we obtain $(\cdot, S_{i_b}, T_{i_b}) \leftarrow \mathtt{CH}[i_b]$ and set $\sigma_1 \leftarrow \mathsf{Sign}_{\mathcal{R}}((T_{i_b}, S_{i_b}), \mathsf{sk}_{\mathcal{R}})$.
- Starting from the $j+1$th query to Ch (i.e., for all queries where the answers are still honest in Game 7_j), we obtain $(R_{i_b}, \cdot, \cdot) \leftarrow \mathtt{CH}[i_b]$, choose $\rho \xleftarrow{R} \mathbb{Z}_p$ and set $\sigma_1 \leftarrow \mathsf{Sign}_{\mathcal{R}}((\rho R_{i_b}, \rho P), \mathsf{sk}_{\mathcal{R}})$.

In Game 7_j the first $j-1$ answers are already random due to the previous switches. Furthermore, the validity of the DDH instance (aP, bP, cP) provided by the challenger determines whether the answer of Ch for the jth query are for user i_b or random, i.e., if we are in Game j or in Game $j+1$. That is, $|\Pr[S_j] - \Pr[S_{j+1}]| \leq \epsilon_{\mathsf{DDH}}(\kappa)$.

In Game $7_{q_{\mathsf{Ch}}+1}$ all answers of Ch are random, i.e., this Game is equal to Game 8, i.e., $\Pr[S_8] = \Pr[7_{q_{\mathsf{Ch}}+1}]$. We can conclude the proof by summing over the distinguishing probabilities of all game changes which yields $|\Pr[S_7] - \Pr[S_8]| \leq q_{\mathsf{Ch}} \cdot \epsilon_{\mathsf{DDH}}(\kappa)$.

Theorem 4. *If* SPS-EQ *is* EUF-CMA *secure, and* Π *is sound, then Scheme 1 is traceable.*

Proof. We proceed using a sequence of games, where we let $q \leq \mathsf{poly}(\kappa)$ be the number of queries to the SndToI oracle.

Game 0: The original traceability game.

Game 1: As Game 0, but we obtain $\mathsf{crs_J}$ from a soundness challenger of Π.

Transition - Game 0 → Game 1: This change is conceptual. That is $\Pr[S_0] = \Pr[S_1]$.

Game 2: As Game 1, but after every successful execution of SndToI, we obtain $\hat{R} \leftarrow \Omega.\mathsf{Dec}(\mathsf{sk_O}, C_{\mathsf{J}_i})$ and abort if $e(U_i, \hat{P}) \neq e(Q, \hat{R})$.

Transition - Game 0 → Game 1: If we abort we have a valid proof π_{J_i} attesting that $(U_i, Q, \hat{C}_{\mathsf{J}_i}, \mathsf{pk_O}) \in L_{\mathsf{R_J}}$, but by the perfect correctness of Ω there exists no ω such that $C_{\mathsf{J}_i} = \Omega.\mathsf{Enc}(\mathsf{pk_O}, r \cdot \hat{P};\ \omega)\ \wedge\ U_i = r \cdot Q$, i.e., $(U_i, Q, \hat{C}_{\mathsf{J}_i}, \mathsf{pk_O})$ is actually not in L_{R_J}. Thus, both games proceed identically unless the adversary breaks the soundness of Π in one oracle query, i.e., $|\Pr[S_1] = \Pr[S_2]| \leq q \cdot \epsilon_{\mathsf{S}}(\kappa)$.

Game 3: As Game 2, but we obtain BG and a public key $\mathsf{pk}_{\mathcal{R}}$ from an $\mathsf{EUF\text{-}CMA}$ challenger of the $\mathsf{SPS\text{-}EQ}$. Whenever an $\mathsf{SPS\text{-}EQ}$ signature is required, the message to be signed is forwarded to the signing oracle provided by the $\mathsf{EUF\text{-}CMA}$ challenger.

Transition - Game 2 → Game 3: This change is conceptual. That is, $\Pr[S_2] = \Pr[S_3]$.

If the adversary eventually outputs a valid forgery (m, σ), we know that σ contains an $\mathsf{SPS\text{-}EQ}$ signature σ_1 for some (rP, P) such that we have never seen a corresponding $r\hat{P}$, i.e., there is no entry i in the registration table where $\hat{C}_{\mathsf{J}_i}$ contains $r\hat{P}$ s.t. $e(\sigma_1[1][1], \hat{P}) = e(\sigma_1[1][2], r\hat{P})$ holds. Consequently, σ_1 is a valid $\mathsf{SPS\text{-}EQ}$ signature for an unqueried equivalence class and we have that $\Pr[S_3] \leq \epsilon_{\mathsf{F}}(\kappa)$. This yields $\Pr[S_0] \leq \epsilon_{\mathsf{F}}(\kappa) + q \cdot \epsilon_{\mathsf{S}}(\kappa)$, which proves the theorem.

Theorem 5. *If* Π *is sound and adaptively zero-knowledge,* SoK *is simulatable and extractable,* Σ *is* $\mathsf{EUF\text{-}CMA}$ *secure,* Ω *is perfectly correct, and the co-CDHI assumption holds, then Scheme 1 is non-frameable.*

Proof. We proceed using a sequence of games. Thereby we let the number of users in the system be $q \leq \mathsf{poly}(\kappa)$.

Game 0: The original non-frameability game.

Game 1: As Game 0, but we guess the index i^* that will be attacked by the adversary. If the adversary attacks another index, we abort.

Transition - Game 0 → Game 1: The winning probability in Game 1 is the same as in Game 0, unless an abort event happens, i.e., $\Pr[S_1] = \Pr[S_0] \cdot {}^1/_q$.

Game 2: As Game 1, but we run $(\mathsf{crs_J}, \tau_\mathsf{J}) \leftarrow \Pi.\mathcal{S}_1(1^\kappa)$ instead of $\mathsf{crs_J} \leftarrow \Pi.\mathsf{Setup}(1^\kappa)$ upon running GKGen and store the trapdoor τ_J. Then, we simulate all calls to $\Pi.\mathsf{Proof}$ in Join using the simulator (without a witness).

Transition - Game 1 → Game 2: A distinguisher $\mathcal{D}^{1\to2}$ is an adversary against adaptive zero-knowledge of Π, and, therefore, the probability to distinguish Game 1 and Game 2 is negligible, i.e., $|\Pr[S_2] - \Pr[S_1]| \leq \epsilon_{\mathsf{ZK_J}}(\kappa)$.

Game 3: As Game 2, but we obtain $\mathsf{crs_O}$ from a soundness challenger upon running GKGen.

Transition - Game 2 → Game 3: This change is conceptual. That is, $\Pr[S_3] = \Pr[S_2]$.

Game 4: As Game 3, but we setup the SoK in simulation mode, i.e., we run $(\mathsf{crs_S}, \tau_\mathsf{S}) \leftarrow \mathsf{SoK.SSetup}(1^\kappa)$ instead of $\mathsf{crs_S} \leftarrow \mathsf{SoK.Setup}(1^\kappa)$ upon running GKGen and store the trapdoor τ_S. Then, we simulate all calls to $\mathsf{SoK.Sign}$ using the simulator, i.e., without a witness.

Transition - Game 3 → Game 4: A distinguisher $\mathcal{D}^{3\to 4}$ is an adversary against simulatability of SoK. Therefore, the distinguishing probability is negligible, i.e., $|\Pr[S_4] - \Pr[S_3]| \leq \epsilon_{\mathsf{SIM}}(\kappa)$.

Game 5: As Game 4, but we choose the values $r, q \xleftarrow{R} \mathbb{Z}_p$ used in the Join algorithm (executed within the SndToU oracle) when queried for user with index i^* beforehand and let (U_{i^*}, Q_{i^*}) denote (rqP, qP). Then, on every Join (within SndToU) for a user $i \neq i^*$ we check whether we have incidentally chosen the same class as for user i^*. This check is implemented as follows: with r_i being the value for r chosen upon Join for user i, we check whether $U_{i^*} = r_i \cdot Q_{i^*}$ (note that this check does not require to know the discrete logarithms q and r for user i^*).

Transition - Game 4 → Game 5: Both games proceed identically unless we have to abort. We abort with probability $\epsilon_{\mathsf{guess}}(\kappa) = {}^{q}/_{p-1}$ and we have that $|\Pr[S_4] - \Pr[S_5]| \leq \epsilon_{\mathsf{guess}}(\kappa)$.

Game 6: As Game 5, but we obtain a co-CDHI instance $(aP, {}^1/_a\hat{P})$ relative to BG and choose $\tau \xleftarrow{R} \mathbb{Z}_p$. Then, we modify the Join algorithm (executed within the SndToU oracle) when queried for user with index i^* as follows. We set $(U_{i^*}, Q_{i^*}) \leftarrow (\tau \cdot P, aP)$, and compute $\hat{C}_{\mathsf{J}_{i^*}} \leftarrow \Omega.\mathsf{Enc}(\mathsf{pk_O}, \tau \cdot {}^1/_a\hat{P})$ and store τ. On successful execution we set $\mathsf{gsk}_{i^*} \leftarrow ((U_{i^*}, Q_{i^*}), \sigma')$ (note that $\pi_{\mathsf{J}_{i^*}}$ as well as the signatures in the GSig oracle are already simulated, i.e., the discrete log of no Q_i value is required to be known to the environment).

Transition - Game 5 → Game 6: Since τ is uniformly random, we can write it as $\tau = ra$ for some $r \in \mathbb{Z}_p$. Then it is easy to see that the game change is conceptual, i.e., $\Pr[S_5] = \Pr[S_4]$.

Game 7: As Game 6, but for every forgery output by the $\mathcal{A}$, we extract $\rho \leftarrow \mathsf{SoK.Extract}(\mathsf{crs_S}, \tau_\mathsf{S}, (P, \sigma_1[1][2]), \sigma_1||m, \sigma_2)$ and abort if the extraction fails.

Transition - Game 6 → Game 7: By the extractability of the SoK, one can extract a witness ρ with overwhelming probability $1 - \epsilon_{\mathsf{EXT}}(\kappa)$. Thus, both games proceed identically unless the extractor fails $|\Pr[S_6] - \Pr[S_7]| \leq \epsilon_{\mathsf{EXT}}(\kappa)$.

Game 8: As Game 7, but we further modify the Join algorithm when queried for user with index i^* (executed within the SndToU oracle) as follows. Instead of choosing $(\mathsf{usk}_{i^*}, \mathsf{upk}_{i^*}) \leftarrow \mathsf{UKGen}(1^\kappa)$, we engage with an EUF-CMA challenger, obtain upk_{i^*} and set $\mathsf{usk}_{i^*} \leftarrow \emptyset$. If any signature is required, we obtain it using the oracle provided by the EUF-CMA challenger.

Transition Game 7 → Game 8: This change is conceptual. That is, $\Pr[S_7] = \Pr[S_6]$.

Now we have three possibilities if $\mathcal{A}$ outputs a valid forgery.

1. If a signature for $\hat{C}_{\mathsf{J}_{i^*}}$ was never requested, $\mathcal{A}$ is an EUF-CMA forger for Σ and the forgery is $(\hat{C}_{\mathsf{J}_{i^*}}, \sigma_{\mathsf{J}_{i^*}})$. The probability for this event is upper bounded by $\epsilon_{\mathsf{f}}(\kappa)$.

2. Otherwise, we know that $\hat{C}_{\mathsf{J}_{i^*}}$ is honestly computed by the environment and—by the perfect correctness of Ω—thus contains ${}^{\tau}\!/\!{}_{a}\hat{P}$, which leaves us two possibilities:
   (a) If $e(\sigma[1][1], \hat{P}) = e(\sigma[1][2], {}^{\tau}\!/\!{}_{a}\hat{P})$, $\mathcal{A}$ is an adversary against co-CDHI, since we can obtain $(((\tau \cdot {}^{1}\!/\!{}_{a}P, P), \sigma')) \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(\sigma_1, \rho^{-1}, \mathsf{pk}_{\mathcal{R}})$ and use τ to output $\tau^{-1} \cdot (\tau \cdot {}^{1}\!/\!{}_{a}P) = {}^{1}\!/\!{}_{a}P$. The probability for this to happen is upper bounded by $\epsilon_{\mathsf{co\text{-}CDHI}}(\kappa)$.
   (b) If $e(\sigma[1][1], \hat{P}) \neq e(\sigma[1][2], {}^{\tau}\!/\!{}_{a}\hat{P})$, $\mathcal{A}$ has produced an opening proof for a statement which is actually not in $L_{R_{\mathsf{O}}}$. The probability for this to happen is upper bounded by $\epsilon_{\mathsf{S}}(\kappa)$.

Taking the union bound we obtain $\epsilon_{\mathsf{nf8}}(\kappa) \leq \epsilon_{\mathsf{f}}(\kappa) + \epsilon_{\mathsf{co\text{-}CDHI}}(\kappa) + \epsilon_{\mathsf{S}}(\kappa)$, which yields the following bound for the success probability in Game 1: $\Pr[S_0] \leq q \cdot (\epsilon_{\mathsf{nf8}}(\kappa) + \epsilon_{\mathsf{ZK_J}}(\kappa) + \epsilon_{\mathsf{SIM}}(\kappa) + \epsilon_{\mathsf{guess}}(\kappa) + \epsilon_{\mathsf{EXT}}(\kappa))$, which is negligible.[6]

Theorem 6. *If Ω is perfectly correct, and Σ is* EUF-CMA *secure, then Scheme 1 is weakly opening sound.*

Proof (Proof (Sketch)). Upon honestly executing Join for users i and j, the probability that their r (resp. $\hat{R}$) values collide is negligible. The perfect correctness of Ω and the EUF-CMA security of Σ thus uniquely determine user i as the signer of σ with overwhelming probability. Then, it is easy to see that an adversary against weak opening soundness is an adversary against soundness of Π.

5 Instantiation in the ROM

To compare our approach to existing schemes regarding signature size and computational effort upon signature generation and verification, we present the sign and verification algorithms for an instantiation of our scheme with the SPS-EQ from [FHS18, FHS15]. We instantiate SoKs in the ROM by applying the transformation from [FKMV12] to Fiat-Shamir (FS) transformed Σ-protocols.

We note that the proofs performed within Join and Open, i.e., proving membership in the languages associated to NP relations $\mathsf{R_J}$ and $\mathsf{R_O}$, respectively, can straight forwardly be instantiated using standard techniques. Therefore, and since they are neither required within Sign nor Vrf, we do not discuss instantiations here.

The SPS-EQ Scheme from [FHS18, FHS15]. Before we introduce the approaches to obtain CPA-fully (resp. CCA2-fully) anonymous instantiations, we recall the SPS-EQ scheme from [FHS18, FHS15], which provides all required security properties, in Scheme 2 (we omit BGGen for brevity). Here, assuming the DDH assumption to hold on the message space yields that different message-signature pairs from the same equivalence class cannot be linked.

[6] We note that we could also write the three cases in the final step as three additional game changes where we abort upon the respective forgeries. However, we opted for this more compact presentation, which also gives us the same bound.

$\mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, \ell)$**:** On input a bilinear-group description BG and vector length $\ell > 1$ in unary, choose $(x_i)_{i\in[\ell]} \xleftarrow{R} (\mathbb{Z}_p)^\ell$, set secret key $\mathsf{sk} \leftarrow (x_i)_{i\in[\ell]}$, compute public key $\mathsf{pk} \leftarrow (\hat{X}_i)_{i\in[\ell]} = (x_i\hat{P})_{i\in[\ell]}$ and output $(\mathsf{sk}, \mathsf{pk})$. We assume that all other algorithms have implicit input BG.

$\mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk})$ **:** On input a representative $M = (M_i)_{i\in[\ell]}$ of equivalence class $[M]_{\mathcal{R}}$ and a secret key $\mathsf{sk} = (x_i)_{i\in[\ell]} \in (\mathbb{Z}_p)^\ell$, return $\perp$ if $M_i \notin \mathbb{G}_1^*$ for some $i \in [\ell]$. Else, choose $y \xleftarrow{R} \mathbb{Z}_p$ and output $\sigma \leftarrow (Z, Y, \hat{Y})$ with

$$Z \leftarrow y \textstyle\sum_{i\in[\ell]} x_i M_i \qquad\qquad Y \leftarrow \frac{1}{y}P \qquad\qquad \hat{Y} \leftarrow \frac{1}{y}\hat{P}.$$

$\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk})$**:** On input a representative $M = (M_i)_{i\in[\ell]}$ of equivalence class $[M]_{\mathcal{R}}$, a signature $\sigma = (Z, Y, \hat{Y})$ and public key $\mathsf{pk} = (\hat{X}_i)_{i\in[\ell]}$, output 0 if for some $i \in [\ell]$: $M_i \notin \mathbb{G}_1^*$ or $\hat{X}_i \notin \mathbb{G}_2^*$; or if $Z \notin \mathbb{G}_1$ or $Y \notin \mathbb{G}_1^*$ or $\hat{Y} \notin \mathbb{G}_2^*$. Return 1 if the following equations hold and 0 otherwise:

$$\textstyle\prod_{i\in[\ell]} e(M_i, \hat{X}_i) = e(Z, \hat{Y}) \quad \wedge \quad e(Y, \hat{P}) = e(P, \hat{Y}).$$

$\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \mu, \mathsf{pk})$**:** On input a representative $M = (M_i)_{i\in[\ell]}$ of equivalence class $[M]_{\mathcal{R}}$, signature $\sigma = (Z, Y, \hat{Y})$, $\mu \in \mathbb{Z}_p$ and public key pk, return $\perp$ if $\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk}) = 0$. Otherwise pick $\psi \xleftarrow{R} \mathbb{Z}_p$ and return $\hat{\sigma} \leftarrow (\psi\mu Z, \frac{1}{\psi}Y, \frac{1}{\psi}\hat{Y})$.

$\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk})$**:** On input $\mathsf{sk} = (x_i)_{i\in[\ell]}$ and $\mathsf{pk} = (\hat{X}_i)_{i\in[\ell]}$, output 1 if for all $i \in [\ell]$: $x_i \in \mathbb{Z}_p$ and $\hat{X}_i \in \mathbb{G}_2^*$ and $x_i\hat{P} = \hat{X}_i$; return 0 otherwise.

Scheme 2: The SPS-EQ scheme from [FHS18, FHS15]

The group signing key gsk_i consists of a vector of two group elements $(R, P) \in (\mathbb{G}_1^*)^2$ and an $\mathsf{SPS\text{-}EQ}$ signature $\sigma \in \mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2^*$ on this vector. Randomization of gsk_i with a random value $\rho \in \mathbb{Z}_p^*$, i.e., $\mathsf{ChgRep}_{\mathcal{R}}$, requires 4 multiplications in $\mathbb{G}_1$ and 1 multiplication in $\mathbb{G}_2$. Verification of the signature in gsk_i requires 5 pairings.

5.1 CPA-Full Anonymity

Subsequently, we show how Sign and Vrf are instantiated in the CPA-full anonymity setting. Therefore, let $H : \{0,1\}^* \to \mathbb{Z}_p$ be a random oracle and let x be the proven statement (which is implicitly defined by the scheme):

$\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}_i, m)$: Parse gsk_i as $((R, P), \sigma)$, choose $\rho \xleftarrow{R} \mathbb{Z}_p$, compute $\sigma_1 = ((R', P'), \sigma') \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(\mathsf{gsk}_i, \rho, \mathsf{pk}_{\mathcal{R}})$. Choose $\nu \xleftarrow{R} \mathbb{Z}_p$, compute $N \leftarrow \nu P$, $c \leftarrow H(N||\sigma_1||m||x)$, $z \leftarrow \nu + c \cdot \rho$, set $\sigma_2 \leftarrow (c, z)$, and return $\sigma \leftarrow (\sigma_1, \sigma_2)$.

$\mathsf{Vrf}(\mathsf{gpk}, m, \sigma)$: Parse σ as $(\sigma_1, \sigma_2) = (((R', P'), \sigma), (c, z))$, return 0 if $\mathsf{Vrf}_{\mathcal{R}}(\sigma_1, \mathsf{pk}_{\mathcal{R}}) = 0$. Otherwise let $N \leftarrow zP - cP'$ and check if $c = H(N||\sigma_1||m||x)$ holds. If so return 1 and 0 otherwise.

The used Σ-protocol is a standard proof of knowledge of the discrete logarithm $\log_P P'$, and it is easy to see that applying the transformations from [FKMV12] yields a SoK in the ROM with the properties we require. Group signatures contain 4 elements in $\mathbb{G}_1$, 1 element in $\mathbb{G}_2$ and 2 elements in $\mathbb{Z}_p$. Counting only the expensive operations, signing costs 5 multiplications in $\mathbb{G}_1$ and 1 multiplication in $\mathbb{G}_2$, and verification costs 2 multiplications in $\mathbb{G}_1$ and 5 pairings.

5.2 CCA2-Full Anonymity

CCA2-full anonymity requires straight-line extractable SoKs, as standard rewinding would lead to an exponential blowup in the reduction (cf. [BFW15]). One possibility would be to rely on the rather inefficient approach to straight-line extraction due to Fischlin [Fis05]. However, as we do not need to straight-line extract the full witness w, but it is sufficient to straight-line extract an image of w under a one-way function $f : \rho \mapsto \rho \cdot \hat{P}$, we can use the notion of straight-line f-extractable SoKs as recently proposed by Cerulli et al. [BCC+15]. This allows us to still use the FS paradigm with good efficiency. The construction uses the generic conversion in [FKMV12, BPW12]. The generic trick in [BCC+15] to obtain straight-line f-extractability is by computing an extractable commitment to the image of the witness w under a function f with respect to an extraction key in the CRS and proving consistency with the witness.[7]

For straight-line extractability, we let $\hat{Y}$ be a public key for the ElGamal variant in $\mathbb{G}_2$ from [BCC+15], which is generated upon $\mathsf{SoK.Setup}$ and represents the CRS of SoK. $\mathsf{SoK.SSetup}$ additionally returns τ such that $\hat{Y} = \tau \cdot \hat{P}$. Furthermore, let x be the proven statement (implicitly defined by the scheme and the generic compiler). Subsequently, we show how Sign and Vrf are instantiated in this setting, where $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$ is modelled as a random oracle:

$\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}_i, m)$: Parse gsk_i as $((R, P), \sigma)$, choose $\rho \xleftarrow{R} \mathbb{Z}_p$, compute $\sigma_1 = ((R', P'), \sigma') \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(\mathsf{gsk}_i, \rho, \mathsf{pk}_{\mathcal{R}})$. Choose $u, \nu, \eta \xleftarrow{R} \mathbb{Z}_p$, compute $(\hat{C}_1, \hat{C}_2) = (u\hat{Y}, \rho\hat{P} + u\hat{P})$, $N \leftarrow \nu P$, $\hat{M}_1 \leftarrow \eta\hat{Y}$, $\hat{M}_2 \leftarrow (\nu + \eta)\hat{P}$, $c \leftarrow H(N||\hat{M}_1||\hat{M}_2||\sigma_1||m||x)$, $z_1 \leftarrow \nu + c \cdot \rho$, $z_2 \leftarrow \eta + c \cdot u$, set $\sigma_2 \leftarrow (\hat{C}_1, \hat{C}_2, c, z_1, z_2)$, and return $\sigma \leftarrow (\sigma_1, \sigma_2)$.

$\mathsf{Vrf}(\mathsf{gpk}, m, \sigma)$: Parse σ as $(\sigma_1, \sigma_2) = (((R', P'), \sigma), (c, z_1, z_2))$, return 0 if $\mathsf{Verify}_{\mathcal{R}}(\sigma_1, \mathsf{pk}_{\mathcal{R}}) = 0$. Otherwise let $N \leftarrow z_1 P - cP'$, $\hat{M}_1 \leftarrow z_2 \cdot \hat{Y} - c \cdot \hat{C}_1$, $\hat{M}_2 \leftarrow (z_1 + z_2) \cdot \hat{P} - c \cdot \hat{C}_2$, and check if $c = H(N||\hat{M}_1||\hat{M}_2||\sigma_1||m||x)$ holds. If so return 1 and 0 otherwise.

Perfect completeness is easy to verify. Below, we prove SHVZK, i.e., that there is an efficient simulator, and special soundenss, i.e., and that there exists an extractor. Note that we additionally require the Σ-protocol to provide quasi-unique responses [Fis05], i.e., given an accepting proof it should be infeasible to find a new valid response for that proof, in order for the compiler in [BCC+15] to apply.

Lemma 1. *The above Σ-protocol is perfectly complete, SHVZK, special-sound and has quasi-unique responses.*

[7] Note that one can still obtain the full witness w using a rewinding extractor.

Proof. We investigate all properties, but omit perfect completeness as it is straight forward to verify.

SHVZK. We describe a simulator which outputs transcripts being indistinguishable from real transcripts. First, it chooses $P' \xleftarrow{R} \mathbb{G}_1, \hat{C}_1 \xleftarrow{R} \mathbb{G}_2, \hat{C}_2 \xleftarrow{R} \mathbb{G}_2$. While P' and $\hat{C}_1$ are identically distributed as in a real transcript, the random choice of $\hat{C}_2$ is not detectable under DDH in $\mathbb{G}_2$ which holds in the SXDH setting (more generally under $\mathsf{IND\text{-}CPA}$ of the used encryption scheme). Then, the simulator chooses $z_1, z_2, c \xleftarrow{R} \mathbb{Z}_p$ and computes $N \leftarrow z_1 \cdot P - c \cdot P'$, $\hat{M}_1 \leftarrow z_2 \cdot \hat{Y} - c \cdot \hat{C}_1$, $\hat{M}_2 \leftarrow (z_1 + z_2) \cdot \hat{P} - c \cdot \hat{C}_2$. It is easy to see that the transcript $(P', \hat{C}_1, \hat{C}_2, N, \hat{M}_1, \hat{M}_2, z_1, z_2, c)$ represents a valid transcript and its distribution is computationally indistinguishable from a real transcript.

Special soundness. Let us consider that we have two accepting answers (z_1, z_2, c) and (z_1', z_2', c') from the prover for distinct challenges $c \neq c'$. Then we have that

$$z_1 - c \cdot \rho = z_1' - c' \cdot \rho \text{ and } z_2 - c \cdot u = z_2' - c' \cdot u,$$

and extract a witness as $\rho \leftarrow \frac{z_1 - z_1'}{c - c'}$, $u \leftarrow \frac{z_2 - z_2'}{c - c'}$.

Quasi-unique responses. The answers z_1 and z_2 are uniquely determined by the word $\hat{Y}$, P', $\hat{C}_1$, $\hat{C}_2$, the commitments N, $\hat{M}_1$, $\hat{M}_2$ as well as the challenge c (and thus the verification equation).

Lemma 2. *Applying the generic conversions from [FKMV12] to the Fiat-Shamir transformed version of the above Σ-protocol with the setup* $\mathsf{SoK.Setup}$ *as described in Section 5.2 produces a signature of knowledge in the random oracle model, that is extractable and straight-line f-extractable.*

Proof. The proof is analogous to [BCC+15], but we re-state it for completeness: For simulatability, we observe that the CRS output by $\mathsf{SoK.SSetup}$ is identical to the CRS output by $\mathsf{SoK.Setup}$ and $\mathsf{SoK.SimSign}$ programs the random oracle to simulate proofs. Simulatability then follows from SHVZK. For extractability we rely on rewinding, special soundness and quasi-unique responses, using the results from [FKMV12]. For straight-line f-extractability, we use the trapdoor τ to decrypt $(\hat{C}_1, \hat{C}_2)$ in the proof transcript and obtain $\rho\hat{P} = f(\rho)$.

Switching Groups. The above protocol requires more operations in the more expensive group $\mathbb{G}_2$ than in $\mathbb{G}_1$. As we work in the SXDH setting, we can switch the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$ and thus all elements in $\mathbb{G}_1$ to $\mathbb{G}_2$ and vice versa, which trades computational efficiency for signature size.

6 Evaluation and Discussion

We now discuss our work in the light of recent concurrent, independent work. Then, we provide a performance evaluation targeting resource-constrained devices.

The [BCC+16] Model. In independent and concurrent work, a new model for fully-dynamic group signatures was proposed by Bootle et al. in [BCC+16]. Bootle et al. address maliciously generated issuer and opener keys, include the notion of opening soundness from [SSE+12] and formally model revocation by means of epochs. Although we target security in a different model, we want to briefly put our construction in context of their recent model.

We could easily incorporate the requirement to support maliciously generated keys in the fashion of [BCC+16] by extending the actual public keys of issuer and opener by a (straight-line extractable) zero-knowledge proof of knowledge of the respective secret keys.

For a practical revocation approach, it seems to be reasonable to choose a re-issuing based approach, i.e., to set up a new group after every epoch, as also used in [BCC+16]. Their group signature construction being secure in their model builds upon accountable ring signatures [BCC+15]. It comes at the cost of a group public key size linear in the number of group members and a signature size logarithmic in the number of group members, and the revocation related re-issuing requires every group member to obtain the new group public key.[8] Applying the same revocation approach to our scheme yields public keys as well as signatures of constant size, and re-issuing requires each remaining group member to re-join the new group.

While our scheme provides weak opening soundness, achieving the stronger notion for our scheme (where the opening authority may be malicious) would require the opening authority to additionally prove that the opened index i corresponds to the lowest index in reg so that the respective entry together with the signature in question satisfies the relation R_{O}. Such a proof could efficiently be instantiated using non-interactive plaintext in-equality proofs [BDSS16]. Nevertheless, we opted to stick with weak opening soundness because: (1) The only benefit of strong opening soundness would be to also cover dishonest opening authorities, while we believe that assuming the opening authority's honesty—given its power to deanonymize every user—is a crucial and very reasonable assumption. (2) Even [SSE+12], who introduced the notion of opening soundness emphasize that already weak opening soundness addresses all the attacks that motivated opening soundness in the first place. (3) Strong opening soundness would unnecessarily degrade the simplicity of our scheme.

6.1 Performance Evaluation and Comparison

To underline the practical efficiency of our approach, we provide a comparison of our ROM instantiation with other schemes in the ROM. Although in recent time we have seen increasing interest in group signatures schemes based on assumptions related to lattices and codes, existing schemes [LLNW16, LLM+16, ABCG17] are far from being competitive with regard to performance, let alone their suitability for current resource constrained devices. Consequently, we put

[8] There is a recent accountable ring signature scheme [LZCS16], which enables constant size signatures.

our focus on pairing based schemes. In particular we use two schemes who follow the approach of Bichsel et al., i.e., [BCN+10, PS16], which provide less desirable anonymity guarantees (denoted CCA^- henceforth), and the well known BBS scheme [BBS04] (with and without precomputations) providing CPA-full anonymity. We note that we use the plain BBS scheme for comparison, which does not even provide non-frameability and the non-frameable version would be even more expensive. Moreover, we use the group signature scheme with the shortest known signatures [DP06] (with and without precomputations) being secure in the strong BSZ model and thus providing CCA2-full anonymity. Finally, we also compare our scheme to the recent CCA2-fully anonymous scheme by Libert et al. [LMPY16] which is secure in the ROM under SXDH.

In Table 1 we provide a comparison of the estimated efficiency in a 254bit BN-pairing setting, where we highlight the values where our scheme is currently the best known scheme among other existing schemes providing the same security guarantees. Our estimations are based on performance values on an ARM-Cortex-M0+ with drop-in hardware accelerator [UW14]. This processor is small enough to be suited for smart cards or wireless sensor nodes [UW14]. Table 2 provides an abstract comparison regarding signature size, computational costs, and type of the underlying assumption.

Scheme	Anon.	Signature Size	Sign	Verify
[BCN+10]	CCA^-	1273bit	351ms	1105ms
[PS16]	CCA^-	1018bit	318ms	777ms
[BBS04]	CPA	2289bit	1545ms	2092ms
[BBS04] (prec.)	CPA	2289bit	1053ms	1600ms
Ours	CPA	**2037bit**	**266ms**	**886ms**
Ours	CCA2	3309bit	**771ms**	**1290ms**
Ours (switch)	CCA2	3563bit	**703ms**	**1154ms**
[DP06]	CCA2	2290bit	1380ms	2059ms
[DP06] (prec.)	CCA2	2290bit	1020ms	1353ms
[LMPY16]	CCA2	2547bit	1688ms	2299ms

Table 1. Estimations based on a BN-pairing implementation on an ARM-Cortex-M0+ with drop-in hardware accelerator, operating at 48MHz [UW14]. The performance figures using 254-bit curves are 33ms-101ms-252ms-164ms ($\mathbb{G}_1$-$\mathbb{G}_2$-$\mathbb{G}_T$-pairing). For the estimation of signature sizes, we use 255bit for elements in $\mathbb{G}_1$, 509bit for elements in $\mathbb{G}_2$ and 254bit for elements in $\mathbb{Z}_p$. The bold values highlight where our schemes are currently the fastest and have the shortest signatures.

Computational Efficiency. When comparing our CPA-fully ano-nymous scheme as well as our CCA2-fully anonymous scheme to other schemes providing the same anonymity guarantees, ours are the *by now fastest ones regarding signature generation and verification costs*. While there are schemes which use slightly less progressive assumptions, it seems that very good performance requires more

Scheme	Anon.	Signature Size	Sign	Verify	Open	Assumption Type
[BCN+10]	CCA^-	$3\mathbb{G}_1 + 2\mathbb{Z}_p$	$1\mathbb{G}_T + 3\mathbb{G}_1$	$5\mathrm{P} + 1\mathbb{G}_T + 1\mathbb{G}_1$	$\mathcal{O}(n)$	Interactive
[PS16]	CCA^-	$2\mathbb{G}_1 + 2\mathbb{Z}_p$	$1\mathbb{G}_T + 2\mathbb{G}_1$	$3\mathrm{P} + 1\mathbb{G}_T + 1\mathbb{G}_1$	$\mathcal{O}(n)$	GGM
[BBS04]	CPA	$3\mathbb{G}_1 + 6\mathbb{Z}_p$	$3\mathrm{P} + 3\mathbb{G}_T + 9\mathbb{G}_1$	$5\mathrm{P} + 4\mathbb{G}_T + 8\mathbb{G}_1$	$\mathcal{O}(1)$	q-Type (non-static) + DCR‡
[BBS04] (prec.)	CPA	$3\mathbb{G}_1 + 6\mathbb{Z}_p$	$3\mathbb{G}_T + 9\mathbb{G}_1$	$4\mathbb{G}_T + 8\mathbb{G}_1$	$\mathcal{O}(1)$	q-Type (non-static) + DCR‡
Ours	CPA	$1\mathbb{G}_2 + 4\mathbb{G}_1 + 2\mathbb{Z}_p$	$1\mathbb{G}_2 + 5\mathbb{G}_1$	$5\mathrm{P} + 2\mathbb{G}_1$	$\mathcal{O}(n)$	GGM
Ours	CCA2	$3\mathbb{G}_2 + 4\mathbb{G}_1 + 3\mathbb{Z}_p$	$6\mathbb{G}_2 + 5\mathbb{G}_1$	$5\mathrm{P} + 4\mathbb{G}_2 + 2\mathbb{G}_1$	$\mathcal{O}(n)$	GGM
Ours (switch)	CCA2	$4\mathbb{G}_2 + 3\mathbb{G}_1 + 3\mathbb{Z}_p$	$5\mathbb{G}_2 + 6\mathbb{G}_1$	$5\mathrm{P} + 2\mathbb{G}_2 + 4\mathbb{G}_1$	$\mathcal{O}(n)$	GGM
[DP06]	CCA2	$4\mathbb{G}_1 + 5\mathbb{Z}_p$	$3\mathrm{P} + 3\mathbb{G}_T + 4\mathbb{G}_1$	$5\mathrm{P} + 4\mathbb{G}_T + 7\ \mathbb{G}_1$	$\mathcal{O}(1)$	q-Type (non-static) + DCR‡
[DP06] (prec.)	CCA2	$4\mathbb{G}_1 + 5\mathbb{Z}_p$	$3\mathbb{G}_T + 8\mathbb{G}_1$	$1\mathrm{P} + 3\mathbb{G}_T + 2\mathbb{G}_2 + 7\mathbb{G}_1$	$\mathcal{O}(1)$	q-Type (non-static) + DCR‡
[LMPY16]	CCA2	$7\mathbb{G}_1 + 3\mathbb{Z}_p$	$4\mathrm{P} + 2\mathbb{G}_T + 16\mathbb{G}_1$	$8\mathrm{P} + 3\mathbb{G}_T + 7\mathbb{G}_1$	$\mathcal{O}(1)$	Standard

Table 2. Comparison of various schemes in the ROM regarding signature size, signing and verification cost, asymptotic opening cost, and required hardness assumptions, where, in terms of computational costs, we only count the expensive operations in $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ as well as the pairings. The values for [BCN+10] and [PS16] are taken from [PS16]. We use 'CCA^-' to denote anonymity in the sense of [BCN+10] and note that precomputation in [BBS04, DP06] requires to store extra elements in $\mathbb{G}_T$. We let n refer to the group size. ‡ . . . For [DP06], and also when [BBS04] would be turned into a dynamic scheme in the BSZ model, one requires an extractable commitment to a discrete log for the dynamic Join; [DP06] use Paillier encryption which requires the DCR assumption.

progressive ones. When looking for instance at the most compact CCA2-fully anonymous group signatures in the standard model under standard assumptions (SXDH and XDLIN) by Libert et al. [LPY15], signature sizes in the best case will have 30 $\mathbb{G}_1$ and 14 $\mathbb{G}_2$ elements ($\approx$ 15000 bit when taking the setting in Table 1), large public keys and computation times that are far from being feasible for resource constrained devices.

Regarding signature generation, we emphasize that our CPA-fully anonymous instantiation is the fastest among all schemes used for comparison (even among the ones providing CCA^- anonymity), and, to the best of our knowledge, *the fastest among all existing schemes*. This is of particular importance since signature generation is most likely to be executed on a constrained device. Regarding signature verification our CPA-fully anonymous instantiation is only outperformed by the CCA^- anonymous instantiation in [PS16].

Signature Size. Comparing schemes providing the same anonymity guarantees, our CPA-fully anonymous instantiation even provides shorter signature sizes than the popular BBS scheme [BBS04] and, to the best of our knowledge, the *shortest signature sizes among all* CPA-*fully anonymous schemes*. Regarding CCA2-fully anonymous schemes, it seems that gained efficiency in the "without explicit encryption" paradigm comes at the cost of larger signatures compared to instantiations following the SEP paradigm.

6.2 Implementation Results

To confirm the relative performance advantage of our scheme as presented in Table 1 (where we present estimations on a smart card like device using the performance figures from [UW14]), we benchmarked actual Java implementations of the scheme in [DP06] and our CCA2-fully anonymous scheme using the JMH benchmarking framework[9] and the BN-pairing implementation of the ECCelerate library.[10] The benchmarks were run on an Intel Core i7-4790 CPU with 16 GB RAM running Ubuntu 17.04. Clearly, on such a powerful platform signature generation and verification is much faster (signature generation and verification are not even noticable). However, it still gives us an insight in the accuracy of our estimations with respect to the relative computational costs of the schemes. We compare the absolute numbers of possible signing operations per second (i.e., the throughput) for both schemes. Our results show that our scheme allows *twice* as much signature generations per second as [DP06] (34.589 ops/s vs. 17.067ops/s), which shows that our relative performance advantage is even slightly better than suggested by the estimations. When increasing the bitlength of p, our scheme even gets more favorable compared to all other schemes, which is mainly due to the fact that we do not require operations in the computationally expensive target group $\mathbb{G}_T$. For example, when switching from 100 bit security (which is the current estimate for 256 bit BN-curves [BD18]) to 128 bit security (i.e.,

[9] `http://openjdk.java.net/projects/code-tools/jmh/`

[10] `https://jce.iaik.tugraz.at/sic/Products/Core_Crypto_Toolkits/ECCelerate`

462 bit BN-curves [BD18]), we can even compute ≈ 2.5 times more signatures than [DP06].

7 Conclusion

In this paper we further pushed the efficiency limits of CPA- as well as CCA2-fully anonymous group signature schemes with respect to signature generation and verification. We also pushed the limits with respect to signature size for CPA-fully anonymous group signature schemes. We observe that our construction is the *only* one which does neither require any pairing computations nor computations in the target group $\mathbb{G}_T$ upon signature creation, which makes it especially suitable for constrained devices. It is also interesting to note that the dynamic Join in [DP06] (and also for [BBS04] when turning it into a dynamic scheme) requires an extractable commitment to a discrete logarithm, which renders their Join procedure rather inefficient. The suggestion in [DP06] is to use Paillier encryption, which besides adding an additional hardness assumption (DCR) in the hidden order group (factoring-based) setting, induces a rather significant performance penalty due to an additional expensive equality proof of discrete logarithms in two different groups (see, e.g., [CM99]). In contrast, our scheme achieves this without any such performance penalty and without additionally requiring the hidden-order setting. Finally, our results affirmatively answer the theoretical question whether CPA- as well as CCA2-fully anonymous schemes following the "without explicit encryption" paradigm are possible at all. Finally, we want to mention that recently Fuchsbauer and Gay [FG18] have proposed a variant of SPS-EQ with a relaxed unforgeability notion under standard assumptions, i.e., Matrix-Diffie-Hellman assumptions. They show that their scheme can be plugged into the group signature construction in this paper. However, relying on standard assumption comes at the cost of decreased performance, making the resulting scheme less attractive regarding efficiency.

Acknowledgements. We thank Josef Schneider for providing us with the implementation of the group signature scheme and the benchmarking framework. We also thank Sherman Chow and various anonymous referees for their valuable comments. The authors have been supported by EU H2020 project PRISMACLOUD, grant agreement n°644962.

References

[ABCG17] Quentin Alamélou, Olivier Blazy, Stéphane Cauchie, and Philippe Gaborit. A code-based group signature scheme. *Des. Codes Cryptography*, 82(1-2):469–493, 2017.

[ACJT00] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO*, 2000.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO*, 2004.

[BCC+15] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on DDH. In *ESORICS*, 2015.

[BCC+16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, and Jens Groth. Foundations of fully dynamic group signatures. In *ACNS*, 2016.

[BCN+10] Patrik Bichsel, Jan Camenisch, Gregory Neven, Nigel P. Smart, and Bogdan Warinschi. Get shorty via group signatures without encryption. In *SCN*, 2010.

[BD18] Razvan Barbulescu and Sylvain Duquesne. Updating key size estimations for pairings. *Journal of Cryptology*, 2018.

[BDSS16] Olivier Blazy, David Derler, Daniel Slamanig, and Raphael Spreitzer. Non-interactive plaintext (in-)equality proofs and group signatures with verifiable controllable linkability. In *CT-RSA*, 2016.

[BFW15] David Bernhard, Marc Fischlin, and Bogdan Warinschi. Adaptive proofs of knowledge in the random oracle model. In *PKC*, 2015.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In *PKC*, 2014.

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT*, 2003.

[Boy08] Xavier Boyen. The uber-assumption family. In *Pairing*, 2008.

[BPW12] David Bernhard, Olivier Pereira, and Bogdan Warinschi. How not to prove yourself: Pitfalls of the fiat-shamir heuristic and applications to helios. In *ASIACRYPT*, 2012.

[BSZ05] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA*, 2005.

[BW06] Xavier Boyen and Brent Waters. Compact Group Signatures Without Random Oracles. In *EUROCRYPT*, 2006.

[BW07] Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In *PKC*, 2007.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO*, 2004.

[CL06] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In *CRYPTO*, 2006.

[CM99] Jan Camenisch and Markus Michels. Separability and efficiency for generic group signature schemes. In *CRYPTO*, 1999.

[CS97] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups (extended abstract). In *CRYPTO*, 1997.

[CvH91] David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT*, 1991.

[DP06] Cécile Delerablée and David Pointcheval. Dynamic fully anonymous short group signatures. In *VIETCRYPT*, 2006.

[FG18] Georg Fuchsbauer and Romain Gay. Weakly secure equivalence-class signatures from standard assumptions. In *Public-Key Cryptography - PKC 2018 - 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part II*, pages 153–183, 2018.

[FHS15] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model. In *CRYPTO*, 2015.

[FHS18] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. *Journal of Cryptology*, 2018.

[Fis05] Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In *CRYPTO*, 2005.

[FKMV12] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the fiat-shamir transform. In *INDOCRYPT*, 2012.

[GMR88] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM JoC*, 17(2), 1988.

[Gro07] Jens Groth. Fully anonymous group signatures without random oracles. In *ASIACRYPT*, 2007.

[HCDF06] Thomas S. Heydt-Benjamin, Hee-Jin Chae, Benessa Defend, and Kevin Fu. Privacy for public transportation. In *Privacy Enhancing Technologies, 6th International Workshop, PET 2006, Cambridge, UK, June 28-30, 2006, Revised Selected Papers*, pages 1–19, 2006.

[HS14] Christian Hanser and Daniel Slamanig. Structure-preserving signatures on equivalence classes and their application to anonymous credentials. In *ASIACRYPT*, 2014.

[HZB+13] Gesine Hinterwälder, Christian T. Zenger, Foteini Baldimtsi, Anna Lysyanskaya, Christof Paar, and Wayne P. Burleson. Efficient e-cash in practice: Nfc-based payments for public transportation systems. In *Privacy Enhancing Technologies - 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings*, pages 40–59, 2013.

[KY05] Aggelos Kiayias and Moti Yung. Group signatures with efficient concurrent join. In *EUROCRYPT*, 2005.

[LLM+16] Benoît Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, and Huaxiong Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *ASIACRYPT 2016*, 2016.

[LLNW16] Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT 2016*, 2016.

[LMPY16] Benoît Libert, Fabrice Mouhartem, Thomas Peters, and Moti Yung. Practical "signatures with efficient protocols" from simple assumptions. In *Asia CCS*, 2016.

[LPY15] Benoît Libert, Thomas Peters, and Moti Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In *CRYPTO*, 2015.

[LZCS16] Russell W. F. Lai, Tao Zhang, Sherman S. M. Chow, and Dominique Schröder. Efficient sanitizable signatures without random oracles. In *ESORICS*, 2016.

[MDND15] Milica Milutinovic, Koen Decroix, Vincent Naessens, and Bart De Decker. Privacy-preserving public transport ticketing system. In *Data and Applications Security and Privacy XXIX - 29th Annual IFIP WG 11.3 Working Conference, DBSec 2015, Fairfax, VA, USA, July 13-15, 2015, Proceedings*, pages 135–150, 2015.

[NS04] Lan Nguyen and Reihaneh Safavi-Naini. Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In *ASIACRYPT*, 2004.

[PPF+12] Macià Mut Puigserver, Magdalena Payeras-Capellà, Josep Lluís Ferrer-Gomila, Arnau Vives-Guasch, and Jordi Castellà-Roca. A survey of electronic ticketing applied to transport. *Computers & Security*, 31(8), 2012.

[PS16] David Pointcheval and Olivier Sanders. Short randomizable signatures. In *CT-RSA*, 2016.

[RBHP15] Andy Rupp, Foteini Baldimtsi, Gesine Hinterwälder, and Christof Paar. Cryptographic theory meets practice: Efficient and privacy-preserving payments for public transport. *ACM Trans. Inf. Syst. Secur.*, 17(3):10:1–10:31, 2015.

[RHBP13] Andy Rupp, Gesine Hinterwälder, Foteini Baldimtsi, and Christof Paar. P4R: privacy-preserving pre-payments with refunds for transportation systems. In *Financial Cryptography and Data Security - 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers*, pages 205–212, 2013.

[SSE+12] Yusuke Sakai, Jacob C. N. Schuldt, Keita Emura, Goichiro Hanaoka, and Kazuo Ohta. On the security of dynamic group signatures: Preventing signature hijacking. In *PKC*, 2012.

[UW14] Thomas Unterluggauer and Erich Wenger. Efficient pairings and ECC for embedded systems. In *CHES*, 2014.

A Plausibility of co-CDHI

To justify that co-CDHI is a plausible assumption, we state the following additional assumption in the Type-3 bilinear group setting, which falls into the Uber assumption family [Boy08] with $R = \langle 1, 1/b \rangle$, $S = \langle 1, b \rangle$, $T = \langle 1 \rangle$, and $f = b^2$.

Definition 28. *Relative to* BGGen *we have that for all PPT adversaries* $\mathcal{A}$ *there exists a negligible function* $\varepsilon(\cdot)$ *such that:*

$$\Pr\left[\mathsf{BG} \leftarrow \mathsf{BGGen}(1^\kappa),\ b \xleftarrow{R} \mathbb{Z}_p,\ \mathbf{c} \leftarrow \mathcal{A}(\mathsf{BG}, 1/bP, b\hat{P})\ :\ \mathbf{c} = e(P, \hat{P})^{b^2}\right] \leq \varepsilon(\kappa).$$

Lemma 3. *If the assumption in Definition 28 holds, then also the co-CDHI assumption holds.*

Proof. Assume a co-CDHI adversary $\mathcal{B}$. We obtain a problem instance $1/bP, b\hat{P}$ relative to BG for the problem underlying the assumption in Definition 28, start $\mathcal{B}(\mathsf{BG}, 1/bP, b\hat{P})$ to obtain $c = bP$, and output $e(P, \hat{P})^{b^2} \leftarrow e(c, b\hat{P})$ with the same probability as $\mathcal{B}$ outputs bP, i.e., breaks co-CDHI. □