J Cryptol (2019) 32:498–546
https://doi.org/10.1007/s00145-018-9281-4

Journal of
CRYPTOLOGY

Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials

Georg Fuchsbauer
Inria, Paris, France
École Normale Supérieure, CNRS, PSL Research University, Paris, France
georg.fuchsbauer@ens.fr

Christian Hanser
Infineon Technologies Austria AG, Graz, Austria
christian.hanser@infineon.com

Daniel Slamanig
AIT Austrian Institute of Technology, Vienna, Austria
daniel.slamanig@ait.ac.at

Communicated by Nigel Smart.

Received 16 March 2016 / Revised 2 February 2018
Online publication 15 February 2018

Abstract. Structure-preserving signatures (SPS) are a powerful building block for cryptographic protocols. We introduce SPS on equivalence classes (SPS-EQ), which allow joint randomization of messages and signatures. Messages are projective equivalence classes defined on group-element vectors, so multiplying a vector by a scalar yields a different representative of the same class. Our scheme lets one adapt a signature for one representative to a signature for another representative without knowledge of any secret. Moreover, given a signature, an adapted signature for a different representative is indistinguishable from a fresh signature on a random message. We propose a definitional framework for SPS-EQ and an efficient construction in Type-3 bilinear groups, which we prove secure against generic forgers. We also introduce set-commitment schemes that let one open subsets of the committed set. From this and SPS-EQ, we then build an efficient multi-show attribute-based anonymous credential system for an arbitrary number of attributes. Our ABC system avoids costly zero-knowledge proofs and only requires a short interactive proof to thwart replay attacks. It is the first credential system whose bandwidth required for credential showing is independent of the number of its attributes, i.e., constant-size. We propose strengthened game-based security definitions for ABC and prove our scheme anonymous against malicious organizations in the standard model; finally, we discuss a concurrently secure variant in the CRS model.

Keywords. Public-key cryptography, Pairing-based cryptography, Structure-preserving signatures, Attribute-based anonymous credentials, Set commitments.

1. Introduction

Digital signatures are an important cryptographic primitive that provide a means for integrity protection, non-repudiation and authenticity of messages in a publicly verifiable way. In most signature schemes, the message space consists of integers in $\mathbb{Z}_{\mathrm{ord}(\mathbb{G})}$ for some group $\mathbb{G}$, or of arbitrary strings mapped to either integers in $\mathbb{Z}_{\mathrm{ord}(\mathbb{G})}$ or elements of a group $\mathbb{G}$ via a cryptographic hash function. In the latter case, the hash function is often modeled as a random oracle (thus, one effectively signs random group elements).

In contrast, structure-preserving signature (SPS) schemes [2–8,20,58,61,72,73] sign group elements without requiring any prior encoding. SPS are defined over two groups $\mathbb{G}_1$ and $\mathbb{G}_2$, equipped with a bilinear map (pairing), and messages are vectors of group elements (from either $\mathbb{G}_1$ or $\mathbb{G}_2$ or both). Moreover, public keys and signatures also consist of group elements only and signatures are verified by deciding group membership of their elements and evaluating the pairing on elements from the public key, the message and the signature. Fully SPS schemes [9,66] also require the secret key to consist of group elements. The main reason for the introduction of SPS was their interoperability with the non-interactive zero-knowledge proof (NIZK) system by Groth and Sahai [67].

Randomization is a useful feature of signature schemes that lets anyone without knowledge of the secret key transform a signature into a new one that looks like a freshly generated signature on the same message. There have been various constructions of randomizable signatures [16,41,42,81,87] and SPS schemes supporting different types of randomization [3,6].

In this paper, we extend randomization by constructing SPS schemes that in addition to randomizing signatures also enable randomization of the signed *messages* in particular ways, and adaptation of the corresponding signatures. We show that such signature schemes are particularly interesting for applications in privacy-enhancing cryptographic protocols, as they allow avoiding costly zero-knowledge proofs.

1.1. *Contribution*

Our contributions can be broken down as follows: (1) introduction and instantiation of SPS on equivalence classes (SPS-EQ), which are defined on group-element vectors; (2) a randomizable set-commitment scheme that enables constant-size opening of subsets of the committed set; and, building on these primitives, (3) a new construction approach for multi-show attribute-based anonymous credentials, which we efficiently instantiate and analyze in a comprehensive security model we propose.

Structure-Preserving Signature Scheme on Equivalence Classes. Inspired by randomizable signatures, we introduce a variant of SPS. Instead of signing message vectors as in previous SPS schemes, our variant signs classes of a projective equivalence relation $\mathcal{R}$ defined over $\mathbb{G}^\ell$ with $\ell > 1$. These classes are lines going through the origin and are determined by the mutual ratios of the discrete logarithms of the vector components. By multiplying each component by the same scalar, a different representative of the same equivalence class is obtained. If the decisional Diffie–Hellman (DDH) assumption holds in group $\mathbb{G}$, then it is hard to decide whether two vectors belong to the same equivalence class.

In SPS-EQ, an equivalence class is signed by signing an arbitrary representative of the class. From this signature, one can then derive a signature for any other representative of the same class, without having access to the secret key. Unforgeability for SPS-EQ holds with respect to classes: after obtaining signatures on representatives of his choice, no adversary should be able to compute a signature on a representative of a class that is different from the ones signed. We also require that an adaptation of a signature is distributed like a freshly computed signature on the new representative. In combination with unlinkability of equivalence classes, this implies the following: given a representative and a signature on it, a random representative of the same class and the adaptation of the signature to it are indistinguishable from a completely random message and a fresh signature on it.

We present a definitional framework for SPS-EQ using game-based security definitions and give an efficient construction whose signatures are short and their length is independent of the message-vector length ℓ. We prove our construction secure in the generic-group model [84].

Set Commitments. We propose a new type of commitment scheme that lets one commit to sets and open arbitrary subsets. We first propose a model for this primitive and then give an efficient construction, which we prove secure in this model. It lets one commit to subsets of $\mathbb{Z}_p$ and a commitment and a subset-opening both consist of a single bilinear-group element. Our scheme is computationally binding, perfectly hiding, and computationally subset-sound, meaning that given a commitment to a set S it is infeasible to produce a subset-opening for some $T \not\subseteq S$. We prove our scheme secure under a generalization of the strong Diffie–Hellman assumption [14].

The scheme also supports commitment randomization, which is compatible with the randomization of messages in our SPS-EQ scheme (i.e., multiplication by a scalar). Randomization is perfect, and the witness used for subset-opening can be adapted accordingly. This property has not been achieved by existing constructions (cf. Sect. 1.2) without relying on costly zero-knowledge proofs of randomization.

A Multi-Show Attribute-Based Anonymous Credential System. An *attribute-based anonymous credential system* provides means for anonymous authentication. Such a system is a multi-party protocol involving a user, an organization (or issuer) and a verifying party. The user can obtain a credential on multiple attributes, such as her nationality or age, from an organization and present the credential to a verifier later on, while revealing only certain attributes. Without learning any information about the user *(anonymity)*, the verifier will be convinced that the presented information (the shown attributes) is authentic *(unforgeability)*. In a *multi-show* credential system, a user obtains a credential from an organization, typically in a non-anonymous way, and can later perform an arbitrary number of showings that are unlinkable to each other.

We propose a new way of building multi-show attribute-based anonymous credentials (often called Privacy-ABCs; we simply write ABCs) from SPS-EQ and set commitments. Using our instantiations, we construct the first standard-model multi-show ABC for which anonymity holds against malicious organization keys and which does not assume a trusted setup.

An SPS-EQ scheme allows randomizing a vector of group elements together with a signature on it, a property we use to achieve unlinkability of credential showings. We use set commitments to commit to a user's attributes. To issue a credential, the issuer signs a message vector containing this set commitment; the credential is essentially this signature together with the opening of the set commitment. During a showing, a subset of the issued attributes can then be opened. Unlinkability of showings is achieved via the randomization properties of both signatures and set commitments, which are compatible with each other. Furthermore, to thwart replay attacks of showings, we add a short constant-size proof of knowledge for providing freshness.

We emphasize that our approach to constructing ABCs differs considerably from existing ones, as we do not use zero-knowledge proofs to selectively disclose attributes during showings. This makes *constant-size* showings possible, as achieved by our construction. In particular, the size of credentials as well as the bandwidth required when showing a credential is independent of the number of possible attributes as well as those contained in the credential; it is a small constant number of group elements. This is the first ABC system with this feature. We note that Camenisch et al. [31] recently proposed an approach to ABCs with the same asymptotic complexity.

We introduce a game-based security model for ABCs in the vein of the Bellare, Shi and Zhang's [29] model for group signatures and prove our ABC system secure in it. Our model considers replays and provides a strong form of anonymity against organizations that may maliciously generate keys—both of which are not considered by earlier models. Replay attacks have often been considered an implementation issue, but we believe that such attacks should already be considered in the formal analysis, avoiding right from the start problems that might later appear in implementations.

We note that previously there were no other comprehensive models for attribute-based credential systems. In independent work, Camenisch et al. [31,37] developed simulation-based notions. The model in [37] is on the one hand very comprehensive and covers many potential features of ABCs such as revocation, multi-credential representation, key binding, blind issuance, pseudonyms, etc.; on the other hand, it only supports non-interactive showings. Its comprehensiveness makes it much more complex and harder to work with than our model, which focuses on covering the basic functionality of an ABC system. The model in [31] focuses on ABCs secure in the universal composability (UC) framework. Unfortunately, these strong security guarantees often come with significantly deteriorated efficiency (as seen in the instantiations in [31]). In contrast, our model can be instantiated with highly efficient constructions, as we show. We further note that [37] and the ABC construction in [31] do consider replays and malicious keys too, although the former in a seemingly weaker sense and the latter only assuming a CRS, whereas our construction does not rely on a trusted setup.

Finally, we discuss a variant of our scheme with smaller organization key sizes that is concurrently secure in the CRS model. We provide a comparison of our ABC system to other existing multi- and one-show ABC approaches.

1.2. *Related Work*

Signatures. Blazy et al. [22] introduce *signatures on randomizable ciphertexts* and modify Waters' signature scheme [87] to instantiate them. Given a signature on a ciphertext, anyone can randomize the ciphertext and adapt the signature accordingly, knowing neither signing key nor encrypted message. Their construction is only practical for very small message spaces, which makes it unsuitable for our purposes.

Another related approach is the proofless variant of the Chaum-Pedersen signature [46], used for self-blindable certificates by Verheul [86]. The certificate as well as the initial message can be randomized using the same scalar, preserving the validity of the certificate. This approach works for the construction in [86], but (as also observed in [86]) it is not a secure signature scheme due to its homomorphic property and the possibility of efficient existential forgeries.

Linearly homomorphic signatures [21,34,57] allow for signing subspaces of a vector space by publishing a signature for all of its basis vectors with respect to the same (file) identifier; this identifier "glues" together the single vectors (of a file). Given a sequence of scalar/signature pairs $(\beta_i, \sigma_i)_{i \in [\ell]}$ for vectors $\vec{v}_i$ (with the same identifier), one can publicly compute a signature for the vector $\vec{v} = \sum_{i \in [\ell]} \beta_i \vec{v}_i$.

If one uses a different identifier for every signed vector $\vec{v}$, then such signatures would support a functionality similar to signature adaptation in SPS-EQ, that is, publicly compute signatures for vectors $\vec{v}' = \beta \vec{v}$ (although they are not structure-preserving). Various constructions also provide a privacy feature called strongly/completely context-hiding [10,11], requiring that a signature resulting from homomorphic operations is indistinguishable from a fresh one. Nevertheless, homomorphic signatures are not applicable to our context: for SPS-EQ unforgeability, we must prevent combination of signatures on several (independent) vectors; so every vector must be assigned a unique identifier. This, however, breaks our unlinkability notion, as every signature can be linked to its initial signature via the unique identifier. The same arguments also apply to structure-preserving linearly homomorphic signatures [76]. Homomorphic signatures supporting richer classes of admissible functions (beside linear ones) have also been considered, but are not applicable in our context either (cf. [1,10] for an overview). We note that the general framework of *P-homomorphic signatures* [1,10] is related to our approach in terms of unforgeability and privacy guarantees, but there are no existing instantiations for the functionality that we require.

Chase et al. [39] introduce *malleable signatures* that let one derive, from a signature on a message m, a signature on $m' = T(m)$ for an "allowable" transformation T. This generalizes signature schemes, including quotable [1,11] or redactable signatures [71,83] with an additional context-hiding property. Letting messages be pseudonyms and allowable transformations map one pseudonym to another, the authors use malleable signatures to construct anonymous credential systems and *delegatable* anonymous credential systems [18]. The general construction in [39], however, relies on malleable zero-knowledge proofs [38] and is not practically efficient—even when instantiated with the Groth-Sahai proof system [67]. Although the above framework is conceptually very different from our approach, we note that SPS-EQ can be cast into the definition of malleable signatures: the evaluation algorithm takes only a single message vector with corresponding signature and there is a single type of allowable transformation. In contrast to Chase et al. [39], our construction is practical and while Chase et al. only focus on transformations of single messages (pseudonyms) in their credentials, we consider multi-show attribute-based anonymous credentials, which is the main focus of our construction.

Set Commitments. The best-known approach for commitments to (ordered) sets are *Merkle hash trees* (MHTs) [78], where for a set S the commitment size is $O(1)$ and the opening of a committed set element is of size $O(\log |S|)$. Boneh and Corrigan-Gibbs [17] propose an alternative MHT construction using a novel commitment scheme based on a bivariate polynomial modulo RSA composites. In contrast to MHTs, their construction supports efficient succinct proofs of knowledge (PoK) of committed values.

Kate, Zaverucha and Goldberg [74] define *polynomial-commitment* schemes that allow to commit to polynomials and support (batch) openings of polynomial evaluations. They can be used to commit to ordered sets (by fixing an index set) or to sets by identifying committed values with roots. Their two constructions are analogues of DL and Pedersen commitments and have $O(1)$-size commitments and openings. Camenisch et al. [31] proposed a variant of the Pedersen version from [74]. A related commitment scheme, called *knowledge commitment*, was proposed by Groth [65] and later generalized by Lipmaa [75].

Other commitments to ordered sets are generalized Pedersen [80] or Fujisaki-Okamoto [56] commitments. Both have commitment size $O(1)$, but opening proofs are of size $O(|S|)$. For completeness, let us also mention *vector commitments* [33], which allow to open specific positions as well as subsequent updates at specific positions (but do not necessarily require hiding). *Zero-knowledge sets* [79] are another primitive in this context, which imply commitments [50]. They allow committing to a set and performing membership and non-membership queries on values without revealing any further information on the set.

Unfortunately, all existing approaches do not simultaneously provide constant-size commitments and subset-openings as well as randomization compatible with the randomization of messages in our proposed SPS-EQ.

ABCs. Signatures providing randomization features together with efficient zero-knowledge (ZK) proofs of knowledge of committed values can be used to generically construct ABC systems. The most prominent example is CL credentials [41,42], based on Σ-protocols. Following Groth and Sahai's [67] efficient non-interactive ZK proofs without random oracles, various constructions of non-interactive anonymous credentials [19, 70] and delegatable (hierarchical) anonymous credentials [18,59] have been proposed. These have a non-interactive showing protocol, that is, the show and verify algorithms do not interact when demonstrating credential possession (also the recent model for conventional ABCs in [37] considers non-interactive showings).

We note that although such credential systems with non-interactive protocols extend the scope of applications of anonymous credentials, the most common use case (i.e., authentication and authorization) essentially relies on interaction in order to provide freshness/liveness. We emphasize that our goal is not to construct non-interactive anonymous credentials.

1.3. *Differences to the Original Work*

The original version of this work by Hanser and Slamanig [69] gave an SPS-EQ instantiation that was shown not to be EUF-CMA by Fuchsbauer [60]. We propose a new instantiation (given in Fig. 1), which we prove EUF-CMA-secure and which is more efficient than the one in [69] in terms of key size, signature size *and* number of verification equations. We also show that our scheme satisfies stronger security properties (Definitions 19 and 20) and discuss their relation to the original properties from [69].

While [69] use the notion of polynomial commitments with factor opening, we found set commitments with subset-openings a more natural notion. We also strengthen the ABC security model from [69]: we define anonymity against adversaries that create malicious organization keys (Definition 29) and provide a stronger notion of unforgeability (Definition 28).

1.4. *Subsequent Work*

Since its introduction, SPS-EQ have been used in various contexts. The attribute-based multi-show anonymous credential system initially presented in [69] was extended in [49] by an efficient revocation mechanism, which takes advantage of the randomization of SPS-EQ.

Besides ABCs, SPS-EQ have also been used to efficiently instantiate other cryptographic concepts. They yielded an intuitive construction of practical *round-optimal blind signatures* in the standard model [54,55], which led to an attribute-based one-show anonymous credential system. They were also used to construct conceptually simple *verifiably encrypted signatures* in the standard model [68]. There it is also shown that certain SPS-EQ imply public-key encryption, which separates them from one-way functions. SPS-EQ were used in [53] for an efficient instantiation of access control encryption [48] and as a building block to construct the most efficient fully anonymous dynamic group signature schemes [51].

Fuchsbauer and Gay [52] have recently constructed an SPS-EQ from standard assumptions (such as DLin) in a weaker security model. Their scheme satisfies unforgeability against adversaries that must reveal the discrete logarithms of the message vectors on which they query signatures. They show that their model is sufficient for the use of SPS-EQ in credential schemes and all other applications considered so far, except for blind signatures.

Apart from results concerning SPS-EQ, let us also mention a recent alternative construction of ABCs by Camenisch et al. [31] from what they call unlinkable redactable signatures. In their approach (whose underlying ideas are related to ours) the size of the credentials and showings is asymptotically identical to that of our construction. However, the concrete efficiency of our construction is much better, partly due to the fact that [31] target security in the universal composability (UC) framework (cf. Sect. 5.6).

2. Preliminaries

A function $\epsilon\colon \mathbb{N} \rightarrow \mathbb{R}^+$ is called *negligible* if for all $c > 0$ there is a k_0 such that $\epsilon(k) < 1/k^c$ for all $k > k_0$. By $a \overset{R}{\leftarrow} S$, we denote that a is chosen uniformly at random from a set S. Furthermore, we write $\mathsf{A}(a_1, \ldots, a_n; r)$ if we want to make the randomness r used by a probabilistic algorithm $\mathsf{A}(a_1, \ldots, a_n)$ explicit and denote by $[\mathsf{A}(a_1, \ldots, a_n)]$ the set of points with positive probability of being output by A. For an (additive) group $\mathbb{G}$, we use $\mathbb{G}^*$ to denote $\mathbb{G} \setminus \{0_{\mathbb{G}}\}$.

Definition 1. (*Bilinear map*) Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be cyclic groups of prime order p, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are additive and $\mathbb{G}_T$ is multiplicative. Let P and $\hat{P}$ be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. We call $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ a *bilinear map* or *pairing* if it is efficiently computable and the following holds:

Bilinearity: $e(aP, b\hat{P}) = e(P, \hat{P})^{ab} = e(bP, a\hat{P}) \quad \forall\, a, b \in \mathbb{Z}_p$.

Non-degeneracy: $e(P, \hat{P}) \neq 1_{\mathbb{G}_T}$, i.e., $e(P, \hat{P})$ generates $\mathbb{G}_T$.

If $\mathbb{G}_1 = \mathbb{G}_2$, then e is *symmetric* (Type-1) and *asymmetric* (Type-2 or 3) otherwise. For Type-2 pairings, there is an efficiently computable isomorphism $\Psi\colon \mathbb{G}_2 \to \mathbb{G}_1$ but none from $\mathbb{G}_1 \to \mathbb{G}_2$; for Type-3 pairings, no efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$ are known. Type-3 pairings are currently the optimal choice in terms of efficiency for a given security level [45].

Definition 2. (*Bilinear-group generator*) A *bilinear-group generator* BGGen is a (possibly probabilistic) polynomial-time algorithm that takes as input a security parameter 1^κ and outputs a description of a bilinear group $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ consisting of groups $\mathbb{G}_1 = \langle P \rangle$, $\mathbb{G}_2 = \langle \hat{P} \rangle$ and $\mathbb{G}_T$ of prime order p with $\lceil \log_2 p \rceil = \kappa$ and a pairing $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

Definition 3. (*DL*) Let BGGen be a bilinear-group generator that outputs $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1 = P, P_2 = \hat{P})$. The *discrete logarithm assumption* holds in $\mathbb{G}_i$ for BGGen if for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$, there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ a \xleftarrow{R} \mathbb{Z}_p,\ a' \xleftarrow{R} \mathcal{A}(\mathsf{BG}, aP_i)\ :\ a' = a\right] \le \epsilon(\kappa)\ .$$

The next assumption states that DL remains hard when given $q-1$ additional elements $a^j P_i$, in both groups (hence "co-"). The assumption is implied, e.g., by the Type-3 bilinear-group counterpart of the q-SDH assumption [14,45].

Definition 4. (q*-co-DL*) Let BGGen be a bilinear-group generator that outputs $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$. The *$q$-co-discrete logarithm assumption* holds for BGGen, if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{l} \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ a \xleftarrow{R} \mathbb{Z}_p \\ a' \xleftarrow{R} \mathcal{A}(\mathsf{BG}, (a^j P, a^j \hat{P})_{j\in[q]}) \end{array}\ :\ a' = a\right] \le \epsilon(\kappa)\ .$$

Note that we will use the q-co-DL assumption statically throughout this paper, that is, q is a fixed system parameter and does not depend on the adversary's behavior, as, e.g., in [14].

Definition 5. (*DDH*) Let BGGen be a bilinear-group generator that outputs $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1 = P, P_2 = \hat{P})$. The *decisional Diffie–Hellman assumption* holds in $\mathbb{G}_i$ for BGGen, if for all PPT adversaries $\mathcal{A}$, there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{l} b \xleftarrow{R} \{0,1\},\ \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ r,s,t \xleftarrow{R} \mathbb{Z}_p \\ b^* \xleftarrow{R} \mathcal{A}(\mathsf{BG}, rP_i, sP_i, ((1-b)\cdot t + b\cdot rs)P_i) \end{array} \ :\ b^* = b\right] - \frac{1}{2} \le \epsilon(\kappa)\ .$$

The XDH assumption formalizes the absence of efficiently computable isomorphisms from $\mathbb{G}_1$ to $\mathbb{G}_2$; the SXDH assumption implies that there is none from $\mathbb{G}_2$ to $\mathbb{G}_1$ either.

Definition 6. (*(S)XDH*) Let BGGen be a bilinear-group generator outputting $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$. The *(symmetric) external Diffie–Hellman assumption* holds for BGGen if DDH holds in $\mathbb{G}_1$ (and in $\mathbb{G}_2$).

Our last assumption (Definition 8) is a special case of Boyen's [26] extended version of the uber-assumption [15]. We first recall the basic uber-assumption for Type-3 bilinear groups:

Definition 7. ($(\mathsf{R}, \mathsf{S}, \mathsf{T}, f)$*-DH*) Let BGGen be a bilinear-group generator that outputs $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$; let $\mathsf{R} = (\mathsf{r}_i)_{i\in[r]}$, $\mathsf{S} = (\mathsf{s}_j)_{j\in[s]}$ and $\mathsf{T} = (\mathsf{t}_k)_{k\in[t]}$ be three tuples of n-variate polynomials in $\mathbb{Z}_p[X_1, \ldots, X_n]$ and let $f \in \mathbb{Z}_p[X_1, \ldots, X_n]$. Define $\mathsf{R}(\vec{x}) := (\mathsf{r}_i(\vec{x})P)_{i\in[r]}$, $\mathsf{S}(\vec{x}) := (\mathsf{s}_i(\vec{x})\hat{P})_{i\in[s]}$ and $\mathsf{T}(\vec{x}) := (e(P, \hat{P})^{\mathsf{t}_i(\vec{x})})_{i\in[t]}$. The $(\mathsf{R}, \mathsf{S}, \mathsf{T}, f)$*-Diffie–Hellman assumption* holds for BGGen, if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{l} \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ \vec{x} \xleftarrow{R} \mathbb{Z}_p^n, \\ e(P, \hat{P})^{f(\vec{x})} \\ \quad \xleftarrow{R} \mathcal{A}(\mathsf{BG}, \mathsf{R}(\vec{x}), \mathsf{S}(\vec{x}), \mathsf{T}(\vec{x})) \end{array} \ :\ \begin{array}{l} \forall \mathbf{M} \in \mathbb{Z}_p^{r\times s}\ \forall \vec{b} \in \mathbb{Z}_p^t : 0 \ne f \\ \ne \displaystyle\sum_{(i,j)\in[r]\times[s]} M_{ij}\mathsf{r}_i\mathsf{s}_j + \sum_{k\in[t]} b_k\mathsf{t}_k \end{array}\right] \le \epsilon(\kappa)\ .$$

Essentially, this assumption says that if a polynomial $f \in \mathbb{Z}_p[X_1, ..., X_n]$ is independent of the polynomials in R, S and T, then given their evaluations at some point $\vec{x} \in \mathbb{Z}_p^n$ (as exponents of the group generators), it is hard to evaluate f at vector $\vec{x}$ (as exponent of the group generator). The assumption holds in the generic-group model [15].

Despite its power, the above assumption does not cover the q-co-SDH assumption [14,45], which states that given $\big(a^i P, a^i \hat{P}\big)_{i\in[q]}$, it is hard to output $\big(s, \frac{1}{a+s}P\big)$ for any s of the adversary's choice. SDH allows the adversary to (1) choose its own target function f (defined by s) from some family $\mathcal{F}$ of functions; and moreover (2) $\mathcal{F}$ can contain rational functions and not only polynomials. Boyen [26, Sects. 6.1 and 6.2] thus extends the uber-assumption to cover these two generalizations and argues that validity in the generic-group model is maintained.

We introduce the following assumption, which is implied by Boyen's extended uber-assumption and generalizes the q-co-SDH assumption. The latter can be cast in the uber-framework by stating that the adversary is given the evaluations at some point a for $(\mathsf{R}, \mathsf{S}, \mathsf{T}) = ((X^i)_{i\in[0,q]}, (X^i)_{i\in[0,q]}, 1)$ and must output a rational function of the form $\frac{1}{h(X)} := \frac{1}{X+s}$ and $\frac{1}{h(a)}P$. We extend the family of target functions from $\mathcal{F}_{\mathrm{SDH}} = \big\{\frac{1}{h(X)}\ \big|\ h \in \mathbb{Z}_p[X],\ \deg h = 1\big\}$ to any rational functions whose denominator degree is greater than its enumerator degree; that is

$$\mathcal{F}_q = \left\{ \tfrac{g(X)}{h(X)}\ \middle|\ g, h \in \mathbb{Z}_p[X],\ 0 \le \deg g < \deg h \le q \right\}\ .$$

Note that since any $f = \frac{g}{h} \in \mathcal{F}_q$ is strictly rational (and nonzero since $\deg g \geq 0$), it is independent from all polynomials in $\mathsf{R}, \mathsf{S}, \mathsf{T}$. The asymptotic simulation error in the generic-group model proof of the generalized q-co-SDH assumption attains an error bound cubic in q.

Definition 8. (*q-co-generalized SDH*) Let BGGen be a bilinear-group generator that outputs $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$. Then, the *$q$-co-generalized-strong-Diffie-Hellman assumption* holds for BGGen in $\mathbb{G}_1$, if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{l}\mathsf{BG} \xleftarrow{R} \mathsf{BGGen}(1^\kappa),\ a \xleftarrow{R} \mathbb{Z}_p, \\ \big(g, h, T\big) \xleftarrow{R} \mathcal{A}(\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[q]})\end{array} : \begin{array}{c} T \in \mathbb{G}_1 \ \wedge\ g, h \in \mathbb{Z}_p[X] \ \wedge \\ 0 \leq \deg g < \deg h \leq q \ \wedge \\ e(T, h(a)\hat{P}) = e(g(a)P, \hat{P}) \end{array}\right] \leq \epsilon(\kappa) .$$

Analogously, the above assumption can be defined to require $T \in \mathbb{G}_2$. As with the q-co-DL assumption, we will use the q-co-GSDH assumption statically.

2.1. *Digital Signatures*

Definition 9. (*Signature scheme*) A *digital signature scheme* is a tuple (KeyGen, Sign, Verify) of PPT algorithms:

$\mathsf{KeyGen}(1^\kappa)$: This probabilistic algorithm takes as input a security parameter 1^κ. It outputs a private key sk and a public key pk (we assume that the message space $\mathcal{M}$ can be deduced from pk).

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

$\mathsf{Verify}(m, \sigma, \mathsf{pk})$: This deterministic algorithm takes as input a message $m \in \mathcal{M}$, a signature σ and a public key pk. It outputs 1 if σ is a valid signature for m under pk and 0 otherwise.

A digital signature scheme is secure if it is correct and existentially unforgeable under adaptive chosen-message attacks (EUF-CMA) [62].

Definition 10. (*Correctness*) A digital signature scheme (KeyGen, Sign, Verify) is *correct* if for all $\kappa \in \mathbb{N}$, all key pairs $(\mathsf{sk}, \mathsf{pk}) \in [\mathsf{KeyGen}(1^\kappa)]$ and all $m \in \mathcal{M}$ we have:

$$\Pr\Big[\mathsf{Verify}(m, \mathsf{Sign}(m, \mathsf{sk}), \mathsf{pk}) = 1\Big] = 1 .$$

Definition 11. (*EUF-CMA*) A digital signature scheme (KeyGen, Sign, Verify) is *existentially unforgeable under adaptive chosen-message attacks* if for all PPT algorithms $\mathcal{A}$ with access to a signing oracle $\mathsf{Sign}(\cdot, \mathsf{sk})$ there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[\begin{array}{l}(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}(1^\kappa), \\ (m^*, \sigma^*) \xleftarrow{R} \mathcal{A}^{\mathsf{Sign}(\cdot, \mathsf{sk})}(\mathsf{pk})\end{array} : m^* \notin Q \ \wedge\ \mathsf{Verify}(m^*, \sigma^*, \mathsf{pk}) = 1\right] \leq \epsilon(\kappa) ,$$

where Q is the set of queries that $\mathcal{A}$ has issued to the signing oracle.

2.2. *Zero-Knowledge Proofs of Knowledge*

In this section, we define zero-knowledge proofs of knowledge, which will be used in our construction of ABCs. In particular, we require protocols to prove knowledge of a discrete logarithm. These are best instantiated by starting with Σ-protocols (i.e., three-round public-coin honest-verifier zero-knowledge proofs of knowledge) and then efficiently converting them to (malicious-verifier) zero-knowledge proofs of knowledge, as done in [32]. We provide generic definitions here.

For our definitions, let $L_{\mathcal{R}} = \{x \mid \exists\, w : (x, w) \in \mathcal{R}\} \subseteq \{0, 1\}^*$ be a formal language, where $\mathcal{R} \subseteq \{0, 1\}^* \times \{0, 1\}^*$ is a binary, polynomial-time (witness) relation. For such a relation, the membership of $x \in L_{\mathcal{R}}$ can be decided in polynomial time (in $|x|$) when given a witness w of length polynomial in $|x|$ certifying $(x, w) \in \mathcal{R}$. We consider an interactive protocol $(\mathcal{P}, \mathcal{V})$ between a (potentially unbounded) prover $\mathcal{P}$ and a PPT verifier $\mathcal{V}$ and denote the outcome of the protocol as $(\cdot, b) \leftarrow \big(\mathcal{P}(\cdot, \cdot), \mathcal{V}(\cdot)\big)$ where $b = 0$ indicates that $\mathcal{V}$ rejects and $b = 1$ that it accepts the conversation with $\mathcal{P}$. We require the following properties of an interactive protocol.

Definition 12. (*Completeness*) We call an interactive protocol $(\mathcal{P}, \mathcal{V})$ for a relation $\mathcal{R}$ complete if for all $x \in L_{\mathcal{R}}$ and w such that $(x, w) \in R$ we have that $(\cdot, 1) \leftarrow \big(\mathcal{P}(x, w), \mathcal{V}(x)\big)$ with probability 1.

Definition 13. (*Zero knowledge*) We say that an interactive protocol $(\mathcal{P}, \mathcal{V})$ for a relation $\mathcal{R}$ is zero-knowledge if for all PPT algorithms $\mathcal{V}^*$ there exists a PPT simulator $\mathcal{S}$ such that:

$$\big\{\mathcal{S}^{\mathcal{V}^*}(x)\big\}_{x \in L_{\mathcal{R}}} \approx \big\{\langle\mathcal{P}(x, w), \mathcal{V}^*(x)\rangle\big\}_{(x,w) \in \mathcal{R}} ,$$

where $\langle\mathcal{P}(\cdot, \cdot), \mathcal{V}^*(\cdot)\rangle$ denotes the transcript of the interaction of $\mathcal{P}$ and $\mathcal{V}$, and "$\approx$" denotes (perfect) indistinguishability.

Definition 14. (*Knowledge soundness*) We say that $(\mathcal{P}, \mathcal{V})$ is a proof of knowledge (PoK) relative to an NP-relation $\mathcal{R}$ if for any (possibly unbounded) malicious prover $\mathcal{P}^*$ such that $(\cdot, 1) \leftarrow \big(\mathcal{P}^*(x), \mathcal{V}(x)\big)$ with probability greater than ϵ there exists a PPT knowledge extractor $\mathcal{K}$ (with rewinding black-box access to $\mathcal{P}^*$) such that $\mathcal{K}^{\mathcal{P}^*}(x)$ returns a value w satisfying $(x, w) \in \mathcal{R}$ with probability polynomial in ϵ.

For more formal definitions, see, e.g., [63]. If an interactive protocol is complete, perfect zero knowledge and satisfies knowledge soundness, then we call it a zero-knowledge proofs of knowledge (ZKPoK).

3. Structure-Preserving Signatures on Equivalence Classes

We aim for an efficient, randomizable structure-preserving signature scheme for group-element vectors that allows to jointly randomize messages and signatures in public. We associate messages with representatives of projective equivalence classes defined on the projective space underlying $\mathbb{G}^\ell$ (for $\ell > 1$ and some prime-order group $\mathbb{G}$). Based on such classes, we will construct a signature scheme that allows the randomization of both messages and signatures via a change of representatives and a matching signature update.

Let us detail these equivalence classes. All elements of a vector $(M_i)_{i\in[\ell]} \in (\mathbb{G}^*)^\ell$ share different mutual ratios. These ratios depend on their discrete logarithms and are invariant under the operation $\gamma\colon \mathbb{Z}_p^* \times (\mathbb{G}^*)^\ell \to (\mathbb{G}^*)^\ell$ with $(s, (M_i)_{i\in[\ell]}) \mapsto s \cdot (M_i)_{i\in[\ell]}$. This invariance allows for randomization of messages and coincides with the operation of changing the representative inside projective equivalence classes defined on $\mathbb{G}^\ell$. More precisely, we use the following equivalence relation to partition $(\mathbb{G}^*)^\ell$ into classes:

$$\mathcal{R} = \left\{(\vec{M}, \vec{N}) \in (\mathbb{G}^*)^\ell \times (\mathbb{G}^*)^\ell \;\middle|\; \exists s \in \mathbb{Z}_p^* : \vec{N} = s \cdot \vec{M}\right\} \subseteq (\mathbb{G}^*)^{2\ell} .$$

Note that $\mathcal{R}$ is an equivalence relation if and only if $\mathbb{G}$ has prime order. We exclude the zero element from $\mathbb{G}$, since we require that for any $(M_i)_{i\in[\ell]}$ a randomization $s \cdot (M_i)_{i\in[\ell]}$ looks random in $(\mathbb{G}^*)^\ell$, which is not the case if $M_i = 0$ for some i.

In our scheme, an equivalence class $[\vec{M}]_\mathcal{R}$ is signed by issuing a signature on an arbitrary representative $\vec{M}$ of $[\vec{M}]_\mathcal{R}$. The scheme then allows to choose a different representative $s \cdot \vec{M}$ and to *publicly* adapt a signature for $\vec{M}$ to one for $s \cdot \vec{M}$, i.e., without access to the secret key. One of our goals is to guarantee that two message-signature pairs from the same equivalence class cannot be linked. Messages of the same equivalence class cannot be linked if the DDH assumption holds on the message space. Our approach requires thus a DDH-hard group, which is why we consider structure-preserving signatures (if the messages were vectors of elements from $\mathbb{Z}_p^*$, class membership could be decided efficiently).

3.1. *Definition*

Definition 15. (*SPS-EQ*) A *structure-preserving signature scheme for equivalence relation* $\mathcal{R}$ over $\mathbb{G}_i$ is a tuple $\mathsf{SPS\text{-}EQ}$ of the following polynomial-time algorithms:

$\mathsf{BGGen}_\mathcal{R}(1^\kappa)$ is a bilinear-group generation algorithm which on input a security parameter κ in unary outputs a prime-order bilinear group BG.

$\mathsf{KeyGen}_\mathcal{R}(\mathsf{BG}, 1^\ell)$ is a probabilistic algorithm which on input a bilinear group BG and a vector length $\ell > 1$ in unary outputs a key pair $(\mathsf{sk}, \mathsf{pk})$.

$\mathsf{Sign}_\mathcal{R}(\vec{M}, \mathsf{sk})$ is a probabilistic algorithm which on input a representative $\vec{M} \in (\mathbb{G}_i^*)^\ell$ of an equivalence class $[\vec{M}]_\mathcal{R}$ and a secret key sk outputs a signature σ.

$\mathsf{ChgRep}_\mathcal{R}(\vec{M}, \sigma, \mu, \mathsf{pk})$ is a probabilistic algorithm which on input a representative $\vec{M} \in (\mathbb{G}_i^*)^\ell$ of an equivalence class $[\vec{M}]_\mathcal{R}$, a signature σ for $\vec{M}$, a scalar μ and a public key pk returns an updated signature σ' that is valid for the representative $\vec{M}' = \mu \cdot \vec{M}$.

$\mathsf{Verify}_\mathcal{R}(\vec{M}, \sigma, \mathsf{pk})$ is a deterministic algorithm which given a representative $\vec{M} \in (\mathbb{G}_i^*)^\ell$, a signature σ and a public key pk outputs 1 if σ is valid for $\vec{M}$ under pk and 0 otherwise.

$\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk})$ is a deterministic algorithm which given a secret key sk and a public key pk checks the keys for consistency and returns 1 on success and 0 otherwise.

In case it does not matter which new representative is chosen, $\mathsf{ChgRep}_{\mathcal{R}}$ can be seen as a matching randomization of a signature and its message using randomizer μ without invalidating the signature on the equivalence class. We require the signature resulting from $\mathsf{ChgRep}_{\mathcal{R}}$ to be indistinguishable from a freshly issued signature for the new representative of the same class, that is, $\mathsf{ChgRep}_{\mathcal{R}}$ should also randomize the signature.

The scheme is correct if honestly generated key pairs and signatures verify, and if $\mathsf{ChgRep}_{\mathcal{R}}$ outputs a valid signature.

Definition 16. (*Correctness*) An SPS-EQ scheme $\mathsf{SPS\text{-}EQ}$ over $\mathbb{G}_i$ is *correct* if for all security parameters $\kappa \in \mathbb{N}$, for all $\ell > 1$, all bilinear groups $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) \in [\mathsf{BGGen}_{\mathcal{R}}(1^\kappa)]$, all key pairs $(\mathsf{sk}, \mathsf{pk}) \in [\mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^\ell)]$ and all messages $\vec{M} \in (\mathbb{G}_i^*)^\ell$ and scalars $\mu \in \mathbb{Z}_p^*$ we have:

$$\begin{aligned}
&\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk}) = 1 \quad \text{and}\\
&\Pr\left[\mathsf{Verify}_{\mathcal{R}}(\vec{M}, \mathsf{Sign}_{\mathcal{R}}(\vec{M}, \mathsf{sk}), \mathsf{pk}) = 1\right] = 1 \quad \text{and}\\
&\Pr\left[\mathsf{Verify}_{\mathcal{R}}(\mu \cdot \vec{M},\ \mathsf{ChgRep}_{\mathcal{R}}(\vec{M}, \mathsf{Sign}_{\mathcal{R}}(\vec{M}, \mathsf{sk}), \mu, \mathsf{pk}),\ \mathsf{pk}) = 1\right] = 1\ .
\end{aligned}$$

We define EUF-CMA security w.r.t. equivalence classes. In contrast to the standard notion of EUF-CMA, we consider a forgery a valid signature on a message from any equivalence class for which the forger has not seen signatures. Note that we assume ℓ to be fixed.

Definition 17. (*EUF-CMA*) An SPS-EQ scheme $\mathsf{SPS\text{-}EQ}$ over $\mathbb{G}_i$ is *existentially unforgeable under adaptive chosen-message attacks* if for all $\ell > 1$ and all PPT algorithms $\mathcal{A}$ having access to a signing oracle $\mathsf{Sign}_{\mathcal{R}}(\cdot, \mathsf{sk})$, there is a negligible function $\epsilon(\cdot)$ such that:

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

where Q is the set of queries that $\mathcal{A}$ has issued to the signing oracle.

We now define new properties, which are better suited to work with than the class-hiding game originally introduced in [69]. We start with a class-hiding property on the message space:

Definition 18. (*Class-hiding*) Let $\ell > 1$ and $\mathbb{G}_i^*$ be a base group of a bilinear group. The message space $(\mathbb{G}_i^*)^\ell$ is *class-hiding* if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{l} b \xleftarrow{R} \{0,1\},\ \mathsf{BG} \xleftarrow{R} \mathsf{BGGen}_{\mathcal{R}}(1^\kappa),\ \vec{M} \xleftarrow{R} (\mathbb{G}_i^*)^\ell, \\ \vec{M}^{(0)} \xleftarrow{R} (\mathbb{G}_i^*)^\ell,\ \vec{M}^{(1)} \xleftarrow{R} [\vec{M}]_{\mathcal{R}},\ b^* \xleftarrow{R} \mathcal{A}(\mathsf{BG}, \vec{M}, \vec{M}^{(b)}) \end{array} : \ b^* = b\right] - \frac{1}{2} \le \epsilon(\kappa) .$$

The following shows that the class-hiding property is implied by the DDH assumption.

Proposition 1. *Let $\ell > 1$ and $\mathbb{G}$ be a group of prime order p. Then, $(\mathbb{G}^*)^\ell$ is a class-hiding message space if and only if the DDH assumption holds in $\mathbb{G}$.*

Proof. We first note that DDH (as defined in Definition 5) is equivalent to a variant DDH* where r, s, t are drawn from $\mathbb{Z}_p^*$ instead of $\mathbb{Z}_p$ (as the statistical distance of the respective distributions is negligible). It suffices thus to show that class-hiding is equivalent to DDH*.

"⇒" Let $\mathcal{A}$ be an adversary against DDH*. We define an adversary $\mathcal{B}$ against the class-hiding property of $(\mathbb{G}^*)^\ell$: $\mathcal{B}$ is given an instance $(\mathsf{BG}, \vec{M}, \vec{M}')$, runs $\mathcal{A}$ on (M_1, M_2, M_1', M_2') and outputs whatever $\mathcal{A}$ outputs.

If $\vec{M}' \in [\vec{M}]_{\mathcal{R}}$, then $\vec{M}' = \lambda\vec{M}$ for some $\lambda \in \mathbb{Z}_p^*$ and $(M_1, M_2, M_1', M_2') = (M_1, M_2, \lambda M_1, \lambda M_2)$ is a valid DDH* tuple in $\mathbb{G}$. If $\vec{M}'$ is random, then (M_1, M_2, M_1', M_2') is also random as in the case $b = 0$ in the DDH* game. There are also "false positives", when $\vec{M}' \notin [\vec{M}]_{\mathcal{R}}$ but $(M_1', M_2') = (\lambda M_1, \lambda M_2)$ for some λ. This occurs, however, only with negligible probability; thus, $\mathcal{B}$'s success probability differs only by a negligible amount from that of $\mathcal{A}$, which shows the implication.

"⇐" Let us parametrize the game from Definition 18 by bit b and denote it as Game_b, that is, $\mathcal{A}$ is given $(\mathsf{BG}, \vec{M}, \vec{M}' \xleftarrow{R} (\mathbb{G}^*)^\ell)$ in Game_0 and $(\mathsf{BG}, \vec{M}, \vec{M}' \xleftarrow{R} [\vec{M}]_{\mathcal{R}})$ in Game_1. We next define a hybrid game Game'_j for every $j \in [\ell]$: it chooses $\mu \xleftarrow{R} \mathbb{Z}_p^*$ as well as $R_{j+1}, \ldots, R_\ell \xleftarrow{R} \mathbb{G}^*$ and runs $\mathcal{A}$ on BG, $\vec{M}$ and

$$\vec{M}' := (\mu M_1, \ldots, \mu M_j, R_{j+1}, \ldots, R_\ell) .$$

Note that by definition $\mathsf{Game}'_1 = \mathsf{Game}_0$ and $\mathsf{Game}'_\ell = \mathsf{Game}_1$, respectively.

If there exists an adversary that distinguishes Game_0 from Game_1 with probability $\epsilon(\kappa)$, then for some index $j \in [\ell]$ it distinguishes Game'_{j-1} from Game'_j with probability $\frac{1}{\ell-1}\epsilon(\kappa)$, which is non-negligible if $\epsilon(\kappa)$ is non-negligible. We show how to construct a DDH* distinguisher $\mathcal{B}$ from a distinguisher between Game'_{j-1} and Game'_j.

Given a DDH* instance $(\mathsf{BG}, rP, sP, tP)$, $\mathcal{B}$ picks $(m_i)_{i\in[\ell]} \xleftarrow{R} (\mathbb{Z}_p^*)^\ell$ as well as $R_{j+1}, \ldots, R_\ell \xleftarrow{R} \mathbb{G}^*$, sets

$$\vec{M} \leftarrow \big(m_1 P, \ldots\ldots, m_{j-1}P,\ (rP), m_{j+1}P, \ldots, m_\ell P\big) \tag{1}$$
$$\vec{M}' \leftarrow \big(m_1(sP), \ldots, m_{j-1}(sP), (tP), R_{j+1}, \ldots\ldots, R_\ell\big) \tag{2}$$

and runs $\mathcal{A}$ on $(\mathsf{BG}, \vec{M}, \vec{M}')$. If $(\mathsf{BG}, rP, sP, tP)$ is a "real" instance (i.e., $t = rs$), then the first j elements in (2) are s-multiples of the first j elements in (1), and $\mathcal{B}$ thus simulates Game'_j. If t is random, then so is the jth element in (2) and $\mathcal{B}$ simulates Game'_{j-1}. Hence, any adversary distinguishing Game'_{j-1} from Game'_j can be used to break DDH*. □

The next two definitions have already been used in [55]. The first one formalizes the notion that signatures output by $\mathsf{ChgRep}_{\mathcal{R}}$ are distributed like fresh signatures on the new representative.

Definition 19. (*Signature adaptation*) Let $\ell > 1$. An SPS-EQ scheme $\mathsf{SPS\text{-}EQ}$ on $(\mathbb{G}_i^*)^\ell$ *perfectly adapts signatures* if for all tuples $(\mathsf{sk}, \mathsf{pk}, \vec{M}, \sigma, \mu)$ with

$$\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk}) = 1 \qquad \vec{M} \in (\mathbb{G}_i^*)^\ell \qquad \mathsf{Verify}_{\mathcal{R}}(\vec{M}, \sigma, \mathsf{pk}) = 1 \qquad \mu \in \mathbb{Z}_p^*$$

$\mathsf{ChgRep}_{\mathcal{R}}(\vec{M}, \sigma, \mu, \mathsf{pk})$ and $\mathsf{Sign}_{\mathcal{R}}(\mu\vec{M}, \mathsf{sk})$ are identically distributed.

The following definition demands that this even holds for maliciously generated verification keys. As for such keys there might not even exist a corresponding secret key, we require that adapted signatures are random elements in the space of valid signatures.

Definition 20. (*Signature adaptation under malicious keys*) Let $\ell > 1$. An SPS-EQ scheme $\mathsf{SPS\text{-}EQ}$ on $(\mathbb{G}_i^*)^\ell$ *perfectly adapts signatures under malicious keys* if for all tuples $(\mathsf{pk}, \vec{M}, \sigma, \mu)$ with

$$\vec{M} \in (\mathbb{G}_i^*)^\ell \qquad \mathsf{Verify}_{\mathcal{R}}(\vec{M}, \sigma, \mathsf{pk}) = 1 \qquad \mu \in \mathbb{Z}_p^* \tag{3}$$

we have that the output of $\mathsf{ChgRep}_{\mathcal{R}}(\vec{M}, \sigma, \mu, \mathsf{pk})$ is a uniformly random element in the space of signatures, conditioned on $\mathsf{Verify}_{\mathcal{R}}(\mu\vec{M}, \sigma', \mathsf{pk}) = 1$.

3.2. *Our Construction*

In Fig. 1, we present our SPS-EQ construction defined for a bilinear-group generator BGGen with message space $(\mathbb{G}_1^*)^\ell$. Its signatures consist of two $\mathbb{G}_1$ elements and one $\mathbb{G}_2$ element and public keys are ℓ-tuples from $(\mathbb{G}_2)^*$. Verification is defined via two pairing-product equations. A scheme with message space $(\mathbb{G}_2^*)^\ell$ is easily obtained by swapping the group membership of all elements.

3.3. *Security of Our Construction*

Theorem 1. *The SPS-EQ scheme in Scheme 1 is correct.*

Proof. We have to show that for all $\kappa \in \mathbb{N}$, all $\ell > 1$, all choices of bilinear groups $\mathsf{BG} \overset{R}{\leftarrow} \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$, all choices of key pairs $(\mathsf{sk}, \mathsf{pk}) \overset{R}{\leftarrow} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^\ell)$, all $\vec{M} \in (\mathbb{G}_1^*)^\ell$ and all $\mu \in \mathbb{Z}_p^*$ the following holds:

$$\begin{aligned}
&\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk}) = 1 \quad \wedge \\
&\mathsf{Verify}_{\mathcal{R}}\big(\vec{M}, \mathsf{Sign}_{\mathcal{R}}(\vec{M}, \mathsf{sk};\ y), \mathsf{pk}\big) = 1 \quad \forall\, y \in \mathbb{Z}_p^* \quad \wedge \\
&\mathsf{Verify}_{\mathcal{R}}\big(\mathsf{ChgRep}_{\mathcal{R}}(\vec{M}, \mathsf{Sign}_{\mathcal{R}}(\vec{M}, \mathsf{sk};\ y), \mu, \mathsf{pk};\ \psi), \mathsf{pk}\big) = 1 \quad \forall\, y, \psi \in \mathbb{Z}_p^* .
\end{aligned}$$

$\underline{\mathsf{KeyGen}_{\mathcal{R}}}(\mathsf{BG}, 1^\ell)$: 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.

$\underline{\mathsf{Sign}_{\mathcal{R}}}(\vec{M}, \mathsf{sk})$: On input a representative $\vec{M} = (M_i)_{i\in[\ell]}$ of equivalence class $[\vec{M}]_{\mathcal{R}}$ 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

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

$\underline{\mathsf{Verify}_{\mathcal{R}}}(\vec{M}, \sigma, \mathsf{pk})$: On input a representative $\vec{M} = (M_i)_{i\in[\ell]}$ of equivalence class $[\vec{M}]_{\mathcal{R}}$, 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:

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

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

$\underline{\mathsf{VKey}_{\mathcal{R}}}(\mathsf{sk}, \mathsf{pk})$: 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.

Fig. 1. Scheme 1, an EUF-CMA secure SPS-EQ scheme.

$\mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^\ell)$ returns $\mathsf{sk} \leftarrow (x_i)_{i\in[\ell]} \xleftarrow{R} (\mathbb{Z}_p^*)^\ell$ and $\mathsf{pk} \leftarrow (x_i\hat{P})_{i\in[\ell]}$, which shows the first equation.

$\mathsf{Sign}_{\mathcal{R}}(\vec{M}, \mathsf{sk}; y)$ returns $Z = y\sum_{i\in[\ell]} x_i M_i$, $Y = \frac{1}{y}P$ and $\hat{Y} = \frac{1}{y}\hat{P}$. Plugging this into the first relation in $\mathsf{Verify}_{\mathcal{R}}$, we get

$$\begin{aligned} e(Z, \hat{Y}) &= e\big(y\textstyle\sum_{i\in[\ell]} x_i M_i, \frac{1}{y}\hat{P}\big) = e\big(\textstyle\sum_{i\in[\ell]} x_i M_i, \hat{P}\big)^{y\cdot\frac{1}{y}} = \\ &= \prod_{i\in[\ell]} e(x_i M_i, \hat{P}) = \prod_{i\in[\ell]} e(M_i, \hat{X}_i)\,. \end{aligned}$$

Since $e(Y, \hat{P}) = e(\frac{1}{y}P, \hat{P}) = e(P, \frac{1}{y}\hat{P}) = e(P, \hat{Y})$, the second verification equation is also satisfied.

Finally, $\mathsf{ChgRep}_{\mathcal{R}}\big(\vec{M}, (Z = y\sum_{i\in[\ell]} x_i M_i, Y = \frac{1}{y}P, \hat{Y} = \frac{1}{y}\hat{P}), \mu, \mathsf{pk};\ \psi\big)$ outputs

$$\sigma' = \big(\psi\mu Z,\ \tfrac{1}{\psi}Y,\ \tfrac{1}{\psi}\hat{Y}\big) = \big(\psi y\sum_{i\in[\ell]} x_i \mu M_i,\ \tfrac{1}{\psi}\tfrac{1}{y}P,\ \tfrac{1}{\psi}\tfrac{1}{y}\hat{P}\big)\ ,$$

which is the same as $\mathsf{Sign}_{\mathcal{R}}(\mu\vec{M}, \mathsf{sk}; \psi y)$, and thus verifies by correctness of $\mathsf{Sign}_{\mathcal{R}}$. □

We prove the security of our construction using a direct proof in the generic-group model [84]. Loosely speaking, the generic-group model is a model to study the runtime of generic algorithms in cyclic groups. Such algorithms do not exploit any special structure of the representation of the group elements. Instead, they are only allowed to perform abstract group operations and test whether two group elements are equal; they thus work for any group. This is modeled by providing group operations to an algorithm solely via oracles. In particular, for any discrete logarithm i, a generic algorithm can obtain a random encoding $\sigma(i)$ of iP (where P is a fixed generator) via an oracle and can use further oracles to perform group operations as well as equality checks on encodings of group elements. In the bilinear-group setting, we consider all three groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ as generic and algorithms have additionally access to a pairing oracle.

Theorem 2. *In the generic-group model for Type-3 bilinear groups, Scheme 1 is EUF-CMA secure.*

Proof. In the generic-group model, an adversary only performs generic-group operations (operations in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, pairings and equality tests) by querying the respective group oracle.

We first consider the messages submitted to the signing oracle and the forgery output by the adversary as formal multivariate Laurent polynomials whose variables correspond to the secret values chosen by the challenger, and show that an adversary is unable to symbolically produce an existential forgery (even when it chooses message elements adaptively). Then, in the second part we show that the probability for an adversary to produce an existential forgery by chance is negligible.

The values chosen by the challenger in the unforgeability game, which are unknown to the adversary, are the logarithms $x_1, \ldots, x_\ell$ of the public keys $(\hat{X}_i)_{i\in[\ell]} \in (\mathbb{G}_2^*)^\ell$ and the values $y_1, \ldots, y_q$, picked for the q oracle replies, that is, when the jth signing query for a message $(M_{j,i})_{i\in[\ell]}$ is answered as

$$(Z_j, Y_j, \hat{Y}_j) = \Big(y_j \sum_{i\in[\ell]} x_i M_{j,i},\ \tfrac{1}{y_j}P,\ \tfrac{1}{y_j}\hat{P}\Big)\ .$$

When outputting a forgery $(Z^*, Y^*, \hat{Y}^*)$ for a message $(M_i^*)_{i\in[\ell]}$, the elements the adversary has seen, besides P and $\hat{P}$, are $(Z_j, Y_j)_{j\in[q]}$ in $\mathbb{G}_1$, and $(\hat{Y}_j)_{j\in[q]}$ as well as $(\hat{X}_i)_{i\in[\ell]}$ in $\mathbb{G}_2$. The forgery must thus have been computed by choosing

$$\begin{aligned}&\pi_z, \pi_y, \pi_{\hat{y}}, \pi_{m^*,i}, \rho_{z,j}, \rho_{y,j}, \rho_{m^*,i,j}, \psi_{y,j}, \psi_{\hat{y},j}, \psi_{m^*,i,j}, \chi_{\hat{y},i}\\ &\quad \in \mathbb{Z}_p \ \text{ for } \ j \in [q] \text{ and } i \in [\ell]\end{aligned}$$

and setting

$$\begin{aligned}
Z^* &= \pi_z P + \sum_{j\in[q]} \rho_{z,j} Z_j + \sum_{j\in[q]} \psi_{z,j} Y_j \\
Y^* &= \pi_y P + \sum_{j\in[q]} \rho_{y,j} Z_j + \sum_{j\in[q]} \psi_{y,j} Y_j \\
\hat{Y}^* &= \pi_{\hat{y}} \hat{P} + \sum_{i\in[\ell]} \chi_{\hat{y},i} \hat{X}_i + \sum_{j\in[q]} \psi_{\hat{y},j} \hat{Y}_j \\
M_i^* &= \pi_{m^*,i} P + \sum_{j\in[q]} \rho_{m^*,i,j} Z_j + \sum_{j\in[q]} \psi_{m^*,i,j} Y_j
\end{aligned}$$

Similarly, for all $j \in [q]$ the message $(M_{j,i})_{i\in[\ell]}$ submitted in the jth query is computed as a linear combination of all the $\mathbb{G}_1$ elements the adversary has seen so far, that is,

$$P, Z_1, Y_1, \ldots, Z_{j-1}, Y_{j-1} .$$

By considering all these group elements and taking their discrete logarithms to the bases P and $\hat{P}$, respectively, we obtain the following linear combinations:

$$\begin{aligned}
z^* &= \pi_z + \sum_{j\in[q]} \rho_{z,j} z_j + \sum_{j\in[q]} \psi_{z,j} \frac{1}{y_j} \\
y^* &= \pi_y + \sum_{j\in[q]} \rho_{y,j} z_j + \sum_{j\in[q]} \psi_{y,j} \frac{1}{y_j} \\
\hat{y}^* &= \pi_{\hat{y}} + \sum_{i\in[\ell]} \chi_{\hat{y},i} x_i + \sum_{j\in[q]} \psi_{\hat{y},j} \frac{1}{y_j} \\
m_i^* &= \pi_{m^*,i} + \sum_{j\in[q]} \rho_{m^*,i,j} z_j + \sum_{j\in[q]} \psi_{m^*,i,j} \frac{1}{y_j} \\
m_{j,i} &= \pi_{m,j,i} + \sum_{k\in[j-1]} \rho_{m,j,i,k} z_k + \sum_{k\in[j-1]} \psi_{m,j,i,k} \frac{1}{y_k}
\end{aligned}$$

Observe that all message elements as well as the elements Y^*, $\hat{Y}^*$ of the forgery must be different from $0_{\mathbb{G}_1}$ and $0_{\mathbb{G}_2}$, respectively, by definition. Plugging the forgery into the verification relations yields:

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

and taking discrete logarithms to the basis $e(P, \hat{P})$ in $\mathbb{G}_T$, we obtain the following equations:

$$\sum_{i\in[\ell]} m_i^* x_i = z^* \hat{y}^* \tag{4}$$

$$y^* = \hat{y}^* \tag{5}$$

The values m_i^*, z^*, $\hat{y}^*$, y^* are multivariate Laurent polynomials of total degree $O(q)$ in $x_1, \ldots, x_\ell, y_1, \ldots, y_q$. Our further analysis will be simplified by the following fact.

Claim 1. *For all $n \geq 1$, the monomials that constitute z_n have the form*

$$\frac{1}{y_s^b} \prod_{k \in [t]} y_{j_k} \prod_{k \in [t]} x_{i_k} \tag{6}$$

with $1 \leq t \leq n$; for all $k_1 \neq k_2$: $j_{k_1} \neq j_{k_2}$; for all k: $j_k \leq n \wedge s < j_k$; $j_t = n$; and $b \in \{0, 1\}$.

In particular, the monomials in z_n can contain up to n y's and x's in the numerator and there are as many x's as y's. All of the y's are different, one of them is y_n and the indices of the other y's are smaller than n. There can be (at most) one y in the denominator, and its index is smaller than that of all other y's.

Proof (of Claim 1). We prove the claim by induction on n.

$\underline{n = 1:}$ As before the first signing query, the only element from $\mathbb{G}_1$ available to the adversary is P, we have $m_{1,i} = \pi_{m,1,i}$ and therefore

$$z_1 = \sum_{i \in [\ell]} \pi_{m,1,i} y_1 x_i \ ,$$

which proves the base case.

$\underline{n \to n + 1:}$ Assume for all $k \in [n]$ the monomials of all z_k are of the form in (6). Since

$$m_{n+1,i} = \pi_{m,n+1,i} + \textstyle\sum_{k \in [n]} \rho_{m,n+1,i,k} z_k + \sum_{k \in [n]} \psi_{m,n+1,i,k} \frac{1}{y_k} \ ,$$

by the definition of $\mathsf{Sign}_{\mathcal{R}}$ we have

$$\begin{aligned} z_{n+1} = \sum_{i \in [\ell]} \pi_{m,n+1,i} \ y_{n+1} x_i + \sum_{i \in [\ell]} \sum_{k \in [n]} \rho_{m,n+1,i,k} \ y_{n+1} z_k x_i \\ + \sum_{i \in [\ell]} \sum_{k \in [n]} \psi_{m,n+1,i,k} \ y_{n+1} \frac{1}{y_k} x_i \ . \end{aligned} \tag{7}$$

The monomials in the first and the last sum are as claimed in the statement. By the induction hypothesis any monomial contained in any z_k is of the form

$$\frac{1}{y_s^b} \prod_{p \in [t]} y_{j_p} \prod_{p \in [t]} x_{i_p} \ ,$$

with $t \leq n$, $j_t = k$ and $s < j_p$ for all j_p as well as $j_p < k$, for all j_p with $p < t$ (which are all different). Each such monomial leads thus to a monomial in the 2nd sum in (7) of the form $\frac{1}{y_s^b} \big(y_{n+1} \prod_{p \in [t]} y_{j_p}\big)\big(x_i \prod_{p \in [t]} x_{i_p}\big) =$

$\frac{1}{y_s^b} \prod_{p\in[t']} y_{j_p} \prod_{p\in[t']} x_{i_p}$, with $t' := t+1 \leq n+1$, $j_{t'} := n+1$, $i_{t+1} := i$. Moreover $t' \leq n+1$, all j_p are still different and $\leq n$ and $s < j_p$ for all j_p, which proves the induction step.

Together this proves the claim. □

We will use that by Claim 1 in any monomial in z_k there are always exactly as many y's as x's in the numerator and there are at least one y and one x; moreover, there is at most one y in the denominator (and which does not cancel down). Moreover, we have:

Corollary 1. *Any monomial can only occur in one unique z_n.*

Proof. This is implied by Claim 1 as follows: For any monomial, let i^* be the maximal index such that the monomial contains y_{i^*}. Then, the monomial does not occur in z_n with $n > i^*$, since z_n contains y_n contradicting maximality. It does not occur in z_n with $n < i^*$ either, since all y_j contained in z_n have $j \leq n$, meaning y_{i^*} does not occur in z_n; a contradiction. □

We start by investigating Eq. (5):

$$\begin{aligned} y^* &= \hat{y}^* \\ \pi_y + \sum_{j\in[q]} \rho_{y,j} z_j + \sum_{j\in[q]} \psi_{y,j} \frac{1}{y_j} &= \pi_{\hat{y}} + \sum_{i\in[\ell]} \chi_{\hat{y},i} x_i + \sum_{j\in[q]} \psi_{\hat{y},j} \frac{1}{y_j} \end{aligned}$$

By equating coefficients, and taking into account that by Claim 1 no z_j contains monomials of the form 1, x_i, or $\frac{1}{y_j}$, we obtain $\rho_{y,j} = 0$ for all $j \in [q]$ and

(i) $\pi_{\hat{y}} = \pi_y$
(ii) $\chi_{\hat{y},i} = 0 \quad \forall i \in [\ell]$
(iii) $\psi_{\hat{y},j} = \psi_{y,j} \quad \forall j \in [q]$

Let us now investigate Eq. (4) (where in $\hat{y}^*$ we replace $\pi_{\hat{y}}$, $\chi_{\hat{y},i}$ and $\psi_{\hat{y},j}$ as per (i), (ii) and (iii), respectively):

$$\begin{aligned} \sum_{i\in[\ell]} m_i^* x_i &= z^* \hat{y}^* \\ \sum_{i\in[\ell]} \Big(\pi_{m^*,i} + \sum_{j\in[q]} \rho_{m^*,i,j} z_j + \sum_{j\in[q]} \psi_{m^*,i,j} \frac{1}{y_j}\Big) x_i &= \\ = \Big(\pi_z + \sum_{j\in[q]} \rho_{z,j} z_j &+ \sum_{j\in[q]} \psi_{z,j} \frac{1}{y_j}\Big)\Big(\pi_y + \sum_{k\in[q]} \psi_{y,k} \frac{1}{y_k}\Big) \\ = \pi_z \pi_y + \sum_{j\in[q]} \rho_{z,j} \pi_y z_j &+ \sum_{j\in[q]} \left(\psi_{z,j} \pi_y + \pi_z \psi_{y,j}\right) \frac{1}{y_j} + \\ \sum_{j\in[q]} \sum_{k\in[q]} \rho_{z,j} \psi_{y,k} \frac{1}{y_k} z_j &+ \sum_{j\in[q]} \sum_{k\in[q]} \psi_{z,j} \psi_{y,k} \frac{1}{y_j y_k} . \end{aligned}$$

Equating coefficients for 1, we get:

(iv) $\pi_z \pi_y = 0$

Since by Claim 1, no terms in $z_j x_i$, z_j and $\frac{1}{y_k} z_j$ are of the form $\frac{1}{y_j}$ or $\frac{1}{y_j y_k}$, equating coefficients for $\frac{1}{y_j}$ and $\frac{1}{y_j y_k}$ for all j, k yields:

(v) $\psi_{z,j} \pi_y + \pi_z \psi_{y,j} = 0 \quad \forall j \in [q]$
(vi) $\psi_{z,j} \psi_{y,k} = 0 \quad \forall j, k \in [q]$

By (iv)–(vi), we have simplified Eq. (4) to the following:

$$\sum_{i \in [\ell]} \Big(\pi_{m^*,i} + \sum_{j \in [q]} \rho_{m^*,i,j} z_j + \sum_{j \in [q]} \psi_{m^*,i,j} \frac{1}{y_j} \Big) x_i \\ = \sum_{j \in [q]} \rho_{z,j} \pi_y \, z_j + \sum_{j \in [q]} \sum_{k \in [q]} \rho_{z,j} \psi_{y,k} \frac{1}{y_k} z_j \ . \tag{8}$$

Let us analyze the monomials contained in the z_j's. By (6) in Claim 1, there is an equal number of y's and x's in numerators of such monomials. Therefore, on the LHS the number of x's in all monomials is always greater than that of y's, meaning monomials of type (6) only occur on the RHS of (8).

We now show that $\rho_{z,n} \pi_y \, z_n = 0$ for all $n \in [q]$. Assume that for some $n \in [q]$ this is not the case. Since none of the monomials in z_n can appear on the LHS and, by Corollary 1, they do not appear in any other z_i, $i \neq n$, z_n must be subtracted by a term contained in $\frac{1}{y_k} z_j$ for some $j, k \in [q]$. The term in this z_j must not have y_k in the numerator, as otherwise it would cancel down and the number of y's and x's would be different, meaning it would not correspond to any monomial in z_n (which are of the form (6)). This also means that any monomial contained in z_n (in the first sum on the RHS) must have y_k in the denominator if it is to be equal to a term in $\frac{1}{y_k} z_j$.

Next, we observe that monomials in z_n can only be equal to terms in $\frac{1}{y_k} z_j$ if $j = n$. This is because the maximal i^* with y_{i^*} appearing in z_n would be different for any other z_j, $j \neq n$ (cf. the proof of Corollary 1). But this means that any monomial in z_n, which by the above must have y_k in the denominator, also occurs in the z_n in the double sum, yielding a term with y_k^2 in the denominator. Since this cannot occur anywhere else in the equation by Corollary 1, we arrived at a contradiction. We have thus:

(vii) $\rho_{z,j} \pi_y \, z_n = 0 \quad \forall j \in [q]$

Equation (4) has now the following, simplified representation:

$$\sum_{i \in [\ell]} \Big(\pi_{m^*,i} + \sum_{j \in [q]} \rho_{m^*,i,j} z_j + \sum_{j \in [q]} \psi_{m^*,i,j} \frac{1}{y_j} \Big) x_i = \sum_{j \in [q]} \sum_{k \in [q]} \rho_{z,j} \psi_{y,k} \frac{1}{y_k} z_j \tag{9}$$

From Claim 1, we have that every monomial of z_j has an equal number of y's and x's in the numerator; for all monomials of the LHS, we thus have: (number of y's) = (number of x's) − 1. For such a term to occur on the RHS, this has to be a monomial N in z_j that has y_k in the numerator, so it cancels down and yields a term with more x's than y's. We show that this must be z_k, that is, we show that $\rho_{z,j} \psi_{y,k} = 0$ for all $j \neq k$.

First this holds for $k > j$, since the "largest" y contained in z_j is y_j and thus y_k does not cancel. Second for $k < j$, let us assume that there is at least one pair of coefficients $\rho_{z,j}\psi_{y,k} \neq 0$ with $k < j$. Observe that $\frac{1}{y_k}z_j$ on the RHS still contains y_j as "largest" y-value (by Claim 1). The monomials composing $\frac{1}{y_k}z_j$ do thus only occur in z_j on the LHS, thus $\rho_{m^*,i,j} \neq 0$ for some $i \in [\ell]$. Thus, the monomial N from z_j on the RHS which contains y_k also occurs on the LHS. However, as by Claim 1 every y occurs only once in every monomial, after canceling out y_k from z_j no y_k remains in N on the RHS. As, however, y_k is present in the corresponding monomial in z_j on the LHS, there is no corresponding term on the RHS. A contradiction. We thus obtain:

(viii) $\rho_{z,j}\psi_{y,k} = 0 \quad \forall j, k \in [q], j \neq k$

Since the RHS of (9) cannot be 0 (otherwise, all m_i^* on the LHS would be 0, which is not a valid forgery), we have:

(ix) $\exists\, k \in [q] : \rho_{z,k}\psi_{y,k} \neq 0$

We now argue that there exists exactly one such k: if we had $\rho_{z,k}\psi_{y,k} \neq 0$ as well as $\rho_{z,k'}\psi_{y,k'} \neq 0$ for $k \neq k'$, then $\rho_{z,k} \neq 0$ and $\psi_{y,k'} \neq 0$ and thus $\rho_{z,k}\psi_{y,k'} \neq 0$, which contradicts (viii). We have thus:

(x) $\exists!\, n \in [q] : \rho_{z,n}\psi_{y,n} \neq 0$

By (viii) and (x), Eq. (9) simplifies to

$$\begin{aligned}
&\sum_{i\in[\ell]}\Big(\pi_{m^*,i} + \sum_{j\in[q]} \rho_{m^*,i,j} z_j + \sum_{j\in[q]} \psi_{m^*,i,j}\frac{1}{y_j}\Big)x_i \\
&\quad = \rho_{z,n}\psi_{y,n}\,\frac{1}{y_n} z_n \\
&\quad = \rho_{z,n}\psi_{y,n} \sum_{i\in[\ell]} m_{n,i}x_i \\
&\quad = \rho_{z,n}\psi_{y,n} \sum_{i\in[\ell]}\Big(\pi_{m,n,i} + \sum_{j\in[n-1]} \rho_{m,n,i,j} z_j + \sum_{j\in[n-1]} \psi_{m,n,i,j}\frac{1}{y_j}\Big)x_i\ ,
\end{aligned}$$

where in the 2nd line we substituted z_n by its definition, namely $y_n \sum_{k\in[\ell]} m_{n,k}x_k$, and in the 3rd line we replaced $m_{n,i}$ by its definition. Since by Claim 1, x_i, $z_j x_i$ and $\frac{1}{y_j}x_i$, for all $i \in [\ell]$, $j \in [q]$, do not have common monomials, equating coefficients yields (with $\alpha := \rho_{z,n}\psi_{y,n}$):

$$\pi_{m^*,i} = \alpha\, \pi_{m,n,i} \qquad \rho_{m^*,i,j} = \alpha\, \rho_{m,n,i,j} \qquad \psi_{m^*,i,j} = \alpha\, \psi_{m,n,i,j}$$

This finally means that the message for the forgery is just a multiple of the previously queried message M_n, which completes the first part of the proof.

It remains to show that the probability that an adversary produces an existential forgery by "accident", i.e., that two formally different polynomials collide by evaluating to the same value (or, equivalently, that the difference polynomial evaluates to zero), is negligible. Suppose that the adversary makes q queries to the signing oracle and $O(q)$ queries to the group oracles. Then, all involved formal polynomials resulting from querying the group oracles are of degree $O(q)$ and overall there are $O(\binom{q}{2}) = O(q^2)$ polynomials that could collide (i.e., whose difference polynomial evaluates to zero). Then, by the Schwartz-Zippel lemma and the collision argument, the probability of such an error in the simulation of the generic group is $O(\frac{q^3}{p})$ and is, therefore, negligible in the security parameter. □

Lemma 1. *Scheme 1 has perfect adaptation of signatures and perfect adaptation of signatures under malicious keys.*

Proof. Let $\vec{M} \in (\mathbb{G}_1^*)^\ell$, $\mathsf{pk} \in (\mathbb{G}_2^*)^\ell$ and $(x_i)_{i\in[\ell]}$ be such that $\mathsf{pk} = (x_i\hat{P})_{i\in[\ell]}$. A signature $(Z, Y, \hat{Y}) \in \mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2^*$ satisfying $\mathsf{Verify}_{\mathcal{R}}(\vec{M}, (Z, Y, \hat{Y}), \mathsf{pk}) = 1$ is of the form $(y\sum x_i M_i, \frac{1}{y}P, \frac{1}{y}\hat{P})$ for some $y \in \mathbb{Z}_p^*$. $\mathsf{ChgRep}_{\mathcal{R}}(\vec{M}, (Z, Y, \hat{Y}), \mu, \mathsf{pk})$ for $\mu \in \mathbb{Z}_p^*$ outputs $(y\psi\sum x_i\mu M_i, \frac{1}{y\psi}P, \frac{1}{y\psi}\hat{P})$, which is a uniformly random element σ in $\mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2^*$ conditioned on $\mathsf{Verify}_{\mathcal{R}}(\mu\vec{M}, \sigma, \mathsf{pk}) = 1$.

Scheme 1 moreover satisfies Definition 19, since $\mathsf{sk} = (x_i)_{i\in[\ell]}$ is the only element satisfying $\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk}) = 1$ and $\mathsf{Sign}_{\mathcal{R}}(\mu\vec{M}, \mathsf{sk})$ outputs a uniformly random element σ in $\mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2^*$ conditioned on $\mathsf{Verify}_{\mathcal{R}}(\mu\vec{M}, \sigma, \mathsf{pk}) = 1$ (like $\mathsf{ChgRep}_{\mathcal{R}}$). □

4. Set Commitments

We now introduce a new commitment type that allows for committing to sets and besides ordinary opening also supports opening of subsets. After formalizing the primitive, we give an efficient construction with succinct commitments and openings.

Kate, Zaverucha and Goldberg [74] introduce the notion of constant-size polynomial commitments. They present two schemes, one computationally and one perfectly hiding. Following a similar approach, we construct set commitments which allow us to commit to a set $S \subset \mathbb{Z}_p$ by committing to a monic polynomial whose roots are the elements of S. A feature we are aiming for is opening of subsets of the committed set, which corresponds to opening non-trivial factors of the committed polynomial. Our scheme is perfectly hiding and computationally binding.

4.1. *Definitions*

We first present the model and security properties of our set-commitment scheme. They are adapted from the polynomial-commitment scheme in [69], tailored to sets encoded as monic polynomials.

Definition 21. (*Set commitments*) A *set-commitment scheme* SC consists of the following PPT algorithms.

$\mathsf{Setup}(1^\kappa, 1^t)$: This probabilistic algorithm takes as input a security parameter κ and an upper bound t for the cardinality of committed sets, both in unary form. It outputs public parameters pp (which include a description of an efficiently samplable message space $\mathcal{S}_{\mathsf{pp}}$ containing sets of maximum cardinality t).

$\mathsf{Commit}(\mathsf{pp}, S)$: This probabilistic algorithm takes as input the public parameters pp defining message space $\mathcal{S}_{\mathsf{pp}}$ and a non-empty set $S \in \mathcal{S}_{\mathsf{pp}}$. It outputs a commitment C to set S and opening information O.

$\mathsf{Open}(\mathsf{pp}, C, S, O)$: This deterministic algorithm takes as input the public parameters pp, a commitment C, a set S and opening information O. If O is a valid opening of C to $S \in \mathcal{S}_{\mathsf{pp}}$, it outputs 1, and 0 otherwise.

$\mathsf{OpenSubset}(\mathsf{pp}, C, S, O, T)$: This (deterministic) algorithm takes as input the public parameters pp, a commitment C, a set $S \in \mathcal{S}_{\mathsf{pp}}$, opening information O and a non-empty set T. It returns $\perp$ if $T \not\subseteq S$; else it returns a witness W for T being a subset of the set S committed to in C.

$\mathsf{VerifySubset}(\mathsf{pp}, C, T, W)$: This deterministic algorithm takes as input the public parameters pp, a commitment C, a non-empty set T and a witness W. If W is a witness for T being a subset of the set committed to in C, it outputs 1, and 0 otherwise.

We call a set-commitment scheme *secure* if it is *correct*, *binding*, *subset-sound* and *hiding*. The properties are as follows, where the definitions of correctness, binding and hiding are as for standard commitment schemes.

Definition 22. (*Correctness*) A set-commitment scheme SC is *correct* if for all $t > 0$, all $\kappa > 0$, all $\mathsf{pp} \in [\mathsf{Setup}(1^\kappa, 1^t)]$, all $S \in \mathcal{S}_{\mathsf{pp}}$ and all non-empty $T \subseteq S$ the following holds:

1. $\Pr\left[(C, O) \overset{R}{\leftarrow} \mathsf{Commit}(\mathsf{pp}, S) \;:\; \mathsf{Open}(\mathsf{pp}, C, S, O) = 1\right] = 1$.
2. $\Pr\left[\begin{array}{l}(C, O) \overset{R}{\leftarrow} \mathsf{Commit}(\mathsf{pp}, S),\\ W \leftarrow \mathsf{OpenSubset}(\mathsf{pp}, C, S, O, T)\end{array} : \mathsf{VerifySubset}(\mathsf{pp}, C, T, W) = 1\right] = 1$.

Definition 23. (*Binding*) A set-commitment scheme SC is *binding* if for all $t > 0$ and all PPT adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[\begin{array}{l}\mathsf{pp} \overset{R}{\leftarrow} \mathsf{Setup}(1^\kappa, 1^t),\\ (C, S, O, S', O') \overset{R}{\leftarrow} \mathcal{A}(\mathsf{pp})\end{array} : \begin{array}{r}\mathsf{Open}(\mathsf{pp}, C, S, O) = 1 \wedge\\ \mathsf{Open}(\mathsf{pp}, C, S', O') = 1 \wedge\\ S \neq S'\end{array}\right] \leq \epsilon(\kappa) .$$

Subset soundness requires it to be infeasible to perform subset openings to sets that are not contained in the committed set.

Definition 24. (*Subset soundness*) A set-commitment scheme SC is *subset-sound* if for all $t > 0$ and all PPT adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[\begin{array}{l}\mathsf{pp} \overset{R}{\leftarrow} \mathsf{Setup}(1^\kappa, 1^t),\\ (C, S, O, T, W) \overset{R}{\leftarrow} \mathcal{A}(\mathsf{pp})\end{array} : \begin{array}{r}\mathsf{Open}(\mathsf{pp}, C, S, O) = 1 \wedge\\ \mathsf{VerifySubset}(\mathsf{pp}, C, T, W) = 1 \wedge\\ T \not\subseteq S\end{array}\right] \leq \epsilon(\kappa) .$$

Our hiding notion strengthens the standard one by giving the adversary access to an $\mathsf{OpenSubset}$ oracle that opens the challenge commitment to any subset in the intersection of the two candidate sets.

Definition 25. (*Hiding*) A set-commitment scheme SC is *hiding* if for all $t > 0$ and all PPT adversaries $\mathcal{A}$ with access to an oracle $\mathsf{OpenSubset}$ there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[\begin{array}{l} b \xleftarrow{R} \{0,1\},\ \mathsf{pp} \xleftarrow{R} \mathsf{Setup}(1^\kappa, 1^t), \\ (S_0, S_1, \mathsf{st}) \xleftarrow{R} \mathcal{A}(\mathsf{pp}), \\ (C, O) \xleftarrow{R} \mathsf{Commit}(\mathsf{pp}, S_b), \\ b^* \xleftarrow{R} \mathcal{A}^{\mathsf{OpenSubset}(\mathsf{pp}, C, S_b, O, \cdot \cap S_0 \cap S_1)}(\mathsf{st}, C) \end{array} : \ b^* = b\right] - \frac{1}{2} \le \epsilon(\kappa)\ .$$

The scheme SC is *perfectly hiding* if the above holds for $\epsilon \equiv 0$.

4.2. *The Construction*

We now give a construction SC of a set-commitment scheme based on a bilinear-group generator BGGen. For the sake of compact representation, for $\emptyset \neq S \subset \mathbb{Z}_p$ we define the polynomials $f_S(X) := \prod_{s \in S}(X - s) = \sum_{i=0}^{|S|} f_i \cdot X^i$ and $f_\emptyset(X) := 1$. For a group generator P, since $f_S(a)P = \sum_{i=0}^{|S|}(f_i \cdot a^i)P$, one can efficiently compute $f_S(a)P$ when given $\left(a^i P\right)_{i=0}^{|S|}$ but not a itself.

$\mathsf{Setup}(1^\kappa, 1^t)$: On input a security parameter 1^κ and a maximum set cardinality 1^t run $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) \xleftarrow{R} \mathsf{BGGen}(1^\kappa)$, pick $a \xleftarrow{R} \mathbb{Z}_p$ and output $\mathsf{pp} \leftarrow (\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]})$, which defines message space

$$\mathcal{S}_{\mathsf{pp}} = \{S \subset \mathbb{Z}_p \mid 0 < |S| \le t\}\ .$$

$\mathsf{Commit}(\mathsf{pp}, S)$: On input $\mathsf{pp} = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]})$ and a set $S \in \mathcal{S}_{\mathsf{pp}}$:

- If for some $a' \in S$: $a'P = aP$, output $C \xleftarrow{R} \mathbb{G}_1^*$ and opening $O \leftarrow (1, a')$;
- Else pick $\rho \xleftarrow{R} \mathbb{Z}_p^*$, compute $C \leftarrow \rho \cdot f_S(a)P \in \mathbb{G}_1^*$ and output (C, O) with $O \leftarrow (0, \rho)$.

$\mathsf{Open}(\mathsf{pp}, C, S, O)$: On input $\mathsf{pp} = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]})$, a commitment C, set S, and opening $O = (b, \rho)$: if $C \notin \mathbb{G}_1^*$ or $\rho \notin \mathbb{Z}_p^*$ or $S \notin \mathcal{S}_{\mathsf{pp}}$, then return $\perp$.

- If $O = (1, a')$ and $a'P = aP$, then return 1; else return 0.
- If $O = (0, \rho)$ and $C = \rho \cdot f_S(a)P$, return 1; else return 0.

$\mathsf{OpenSubset}(\mathsf{pp}, C, S, O, T)$: On input $\mathsf{pp} = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]})$, a commitment C, a set S, opening O and a set T, if $\mathsf{Open}(\mathsf{pp}, C, S, O) = 0$ or $T \not\subseteq S$ or $T = \emptyset$, then return $\perp$.

- If $O = (1, a')$: if $a' \in T$, return $W \leftarrow \perp$; else return $W \leftarrow f_T(a')^{-1} \cdot C$.
- If $O = (0, \rho)$, output $W \leftarrow \rho \cdot f_{S \setminus T}(a)P$.

$\mathsf{VerifySubset}(\mathsf{pp}, C, T, W)$: On input $\mathsf{pp} = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]})$, a commitment C, a set T and a witness W: if $C \notin \mathbb{G}_1^*$ or $T \notin \mathcal{S}_{\mathsf{pp}}$, return 0.

- If for some $a' \in T$: $a'P = aP$, then: if $W = \perp$, return 1; else return 0.
- Else: if $W \in \mathbb{G}_1^*$ and $e(W, f_T(a)\hat{P}) = e(C, \hat{P})$, return 1; else return 0.

We have augmented the scheme from [69] by a special opening (of the form $(1, a)$) for the case that a set S contains the trapdoor a. (Under the t-co-DL assumption, such sets are infeasible to find.) This makes the scheme perfectly correct and perfectly hiding while still maintaining computational binding and subset soundness.

We have defined the scheme in a way that reduces the computational complexity of the prover in the ABC system in Sect. 5.4. To improve the performance of $\mathsf{VerifySubset}$, one could define a scheme with $W \in \mathbb{G}_2$ (for which $\mathsf{VerifySubset}$ would have to compute $f_T(a)P$).

Security. We prove SC secure under the q-co-DL and the q-co-GSDH assumption. We use both assumptions in a static way, as $q \leftarrow t$ is a system parameter and fixed a priori.

Theorem 3. SC *is correct.*

Proof. Let $t, \kappa > 0$ and $(\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]}) \xleftarrow{R} \mathsf{Setup}(1^\kappa, 1^t)$ with $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$, let $S \subset \mathbb{Z}_p$ with $0 < |S| \leq t$ and let $\emptyset \neq T \subseteq S$. We consider two cases.

(1) $a \in S$. $\mathsf{Commit}(\mathsf{pp}, \mathsf{S})$ returns (C, O) with $C \in \mathbb{G}_1^*$ and $O = (1, a)$. Open on input $(C, S, (1, a))$ returns 1, which shows the first property. $\mathsf{OpenSubset}(\mathsf{pp}, C, S, O, T)$ returns $W \leftarrow \perp$ if $a \in T$ and $W \leftarrow f_T(a)^{-1} \cdot C$ if $a \notin T$. If $a \in T$, then $\mathsf{VerifySubset}(\mathsf{pp}, C, T, W)$ returns 1, as $W = \perp$. If $a \notin T$, it returns 1 if $C, W \in \mathbb{G}_1^*$ and $e(W, f_T(a)\hat{P}) = e(C, \hat{P})$. This is satisfied, since $W \in \mathbb{G}_1^*$ and $e(W, f_T(a)\hat{P}) = e(f_T(a)^{-1} \cdot C, f_T(a)\hat{P}) = e(C, \hat{P})$.

(2) $a \notin S$. $\mathsf{Commit}(\mathsf{pp}, \mathsf{S})$ returns $(C, (0, \rho))$ with $C = \rho \cdot f_S(a)P$ and $\rho \in \mathbb{Z}_p^*$. Open returns 1, since $\rho \in \mathbb{Z}_p^*$, $S \in \mathcal{S}_{\mathsf{pp}}$, $f_S(a) \neq 0$, thus $C \in \mathbb{G}_1^*$ and C has the required form. $\mathsf{OpenSubset}(\mathsf{pp}, C, (0, \rho), T)$ returns $W \leftarrow \rho \cdot f_{S\setminus T}(a)P$. On input (pp, C, T, W), $\mathsf{VerifySubset}$ returns 1 if $C, W \in \mathbb{G}_1^*$ and $e(W, f_T(a)\hat{P}) = e(C, \hat{P})$. Since $\rho \in \mathbb{Z}_p^*$ and $a \notin S$ we have $W = \rho \cdot f_{S\setminus T}(a)P \in \mathbb{G}_1^*$; moreover, $e(W, f_T(a)\hat{P}) = e(\rho \cdot f_S(a) \cdot f_T(a)^{-1} \cdot P, f_T(a)\hat{P}) = e(\rho \cdot f_S(a)P, \hat{P}) = e(C, \hat{P})$; so $\mathsf{VerifySubset}$ returns 1. □

Theorem 4. *If the t-co-DL assumption (Definition 4) holds, then* SC *is binding.*

Proof. We show that if $\mathcal{A}$ is able to output a commitment C and two valid openings to distinct sets S, S', then we can construct an adversary $\mathcal{B}$ that breaks t-co-DL: $\mathcal{B}$ obtains an instance $I = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]})$, sets $\mathsf{pp} \leftarrow I$ and runs $\mathcal{A}(\mathsf{pp})$. If $\mathcal{A}$ outputs a collision (C, S, O, S', O'), then by $\mathsf{Open}(\mathsf{pp}, C, S, O) = 1$ and $\mathsf{Open}(\mathsf{pp}, C, S', O') = 1$ with $S \neq S'$, it holds that $C \in \mathbb{G}_1^*$ and $\emptyset \neq S, S' \subset \mathbb{Z}_p$. If $O = (1, a')$, then by $\mathsf{Open}(\mathsf{pp}, C, S, O) = 1$, we have $a'P = aP$ and $\mathcal{B}$ outputs a' as solution to the t-co-DL problem. The case $O' = (1, a')$ is dealt analogously. Else, we have $O = (0, \rho)$, $O' = (0, \rho')$ with $\rho, \rho' \in \mathbb{Z}_p^*$ and:

$$\rho \cdot f_S(a)P = C = \rho' \cdot f_{S'}(a)P \ , \tag{10}$$

from which we have $\rho \cdot f_S(a) - \rho' \cdot f_{S'}(a) = 0$. Since S and S' are both non-empty and distinct, we have $\deg f_S > 0$ and $\deg f_{S'} > 0$ and $f_S \neq f_{S'}$. Furthermore, f_S and $f_{S'}$ are monic and $\rho, \rho' \neq 0$, thus $t(X) \leftarrow \rho \cdot f_S(X) - \rho' \cdot f_{S'}(X) \neq 0$ while $t(a) = 0$ by (10). Therefore, a is a root of the nonzero polynomial $t(X) \in \mathbb{Z}_p[X]$ and $t(X)$ is known to $\mathcal{B}$. Factoring $t(X)$ yields a, which $\mathcal{B}$ outputs as solution to the t-co-DL problem. □

Theorem 5. *If the t-co-GSDH assumption (Definition 8) holds, then* SC *is subset-sound.*

Proof. We show that if $\mathcal{A}$ is able to output (C, S, O, T, W), such that O is a valid opening of C to set S, $\mathsf{VerifySubset}(\mathsf{pp}, C, T, W) = 1$ and $T \not\subseteq S$, then we can construct an adversary $\mathcal{B}$ against t-co-GSDH as follows. On input an instance $I = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]})$, $\mathcal{B}$ sets $\mathsf{pp} \leftarrow I$ and runs $\mathcal{A}(\mathsf{pp})$; assume $\mathcal{A}$ breaks subset soundness by outputting (C, S, O, T, W).

We first deal with the case $a \in T$, which $\mathcal{B}$ can efficiently check. In this case, $\mathcal{B}$ chooses $c \in \mathbb{Z}_p \setminus \{-a\}$, and outputs a solution $(1, X + c, \frac{1}{a+c}P)$ to t-co-GSDH.

For the rest of the proof, assume $a \notin T$. If $\mathcal{A}$ is successful, we have $\mathsf{Open}(\mathsf{pp}, C, S, O) = 1$. If $O = (1, a')$, then $a'P = aP$ and $\mathcal{B}$ chooses $c \in \mathbb{Z}_p \setminus \{-a'\}$, and outputs a solution $(1, X + c, \frac{1}{a'+c}P)$ to t-co-GSDH. Else, we have $O = (0, \rho)$ with $\emptyset \neq S \subset \mathbb{Z}_p$, $|S| \leq t$, $\rho \in \mathbb{Z}_p^*$ and

$$C = \rho \cdot f_S(a)P \in \mathbb{G}_1^* \ . \tag{11}$$

From $\mathsf{VerifySubset}(\mathsf{pp}, C, T, W) = 1$, we have $\emptyset \neq T \subset \mathbb{Z}_p$, $|T| \leq t$, $W \in \mathbb{G}_1^*$ and $e(W, f_T(a)\hat{P}) = e(C, \hat{P})$, which by (11) equals $e(\rho \cdot f_S(a)P, \hat{P})$. Since $\rho \neq 0$, we have

$$e(\rho^{-1}W, f_T(a)\hat{P}) = e(f_S(a)P, \hat{P}) \ . \tag{12}$$

We further distinguish two cases:

(1) $0 < |S| < |T|$. Then, $0 < \deg f_S < \deg f_T \leq t$, which together with (12) means that $(f_S, f_T, \rho^{-1}W)$ is a solution to the t-co-GSDH problem.

(2) $0 < |T| \leq |S|$. Then, $0 < \deg f_T \leq \deg f_S$. By polynomial division, we obtain h, r with $f_S(X) = h(X)f_T(X) + r(X)$ and $\deg r < \deg f_T$. Since $T \not\subseteq S$, we have $0 \leq \deg r$ and moreover $\deg h \leq \deg f_S \leq t$. Plugging this into (12), we get:

$$e\big(\rho^{-1}W, f_T(a)\hat{P}\big) = e\big(h(a)f_T(a)P + r(a)P, \hat{P}\big) = e\big(h(a)P, f_T(a)\hat{P}\big) + e\big(r(a)P, \hat{P}\big)$$

and thus

$$e\big(\rho^{-1}W - h(a)P, f_T(a)\hat{P}\big) = e\big(r(a)P, \hat{P}\big) \ .$$

Together with $0 \leq \deg r < \deg f_T \leq t$, this means that $(r, f_T, \rho^{-1}W - h(a)P)$ is a solution to the t-co-GSDH problem, which $\mathcal{B}$ can efficiently compute from pp, since $\deg h \leq t$. □

Theorem 6. SC *is perfectly hiding.*

Proof. We consider the view of an unbounded adversary $\mathcal{A}$ in the hiding experiment and assume w.l.o.g. that every query T to the $\mathsf{OpenSubset}$ oracle satisfies $T \subset \mathbb{Z}_p$ and $\emptyset \neq T \subseteq (S_0 \cap S_1)$. We distinguish several cases.
(1) $\mathcal{A}$ chooses S_0, S_1 with $a \in S_0 \cap S_1$. Then, for both $b \in \{0,1\}$, C_b is uniformly random in $\mathbb{G}_1^*$ and the jth query T_j to $\mathsf{OpenSubset}$ is answered with $\perp$ if $a \in T_j$, and with $W_{j,b} = f_T(a)^{-1} \cdot C_b$ if $a \notin T_j$. The bit b is thus information-theoretically hidden from $\mathcal{A}$.
(2) a is contained in one of the sets S_0, S_1; say $a \in S_0$. Note that for all queries T_j, we have $a \notin T_j$. If $b = 0$, then $\mathcal{A}$ receives a uniformly random C_0 and when it queries T_j to the $\mathsf{OpenSubset}$ oracle, it receives $W_{j,0} = f_{T_j}(a)^{-1} \cdot C_0$. If $b = 1$, then $\mathcal{A}$ receives $C_1 = \rho \cdot f_{S_1}(a) P$ for a random $\rho \in \mathbb{Z}_p^*$, and query T_j to the $\mathsf{OpenSubset}$ oracle returns witness $W_{j,1} = \rho \cdot f_{S_1 \setminus T_j}(a) \cdot P = \rho \cdot f_{S_1}(a) \cdot f_{T_j}(a)^{-1} \cdot P = f_{T_j}(a)^{-1} \cdot C_1$. Hence, for both $b \in \{0,1\}$ we have C_b is uniformly random in $\mathbb{G}_1^*$ and $W_{j,b} = f_{T_j}(a)^{-1} \cdot C_b$ for all j; the bit b is thus information-theoretically hidden from $\mathcal{A}$.
(3) $\mathcal{A}$ chooses S_0, S_1 with $a \notin S_0 \cup S_1$. Then, for both $b \in \{0,1\}$: $C_b = \rho \cdot f_{S_b}(a) P$ for random $\rho \in \mathbb{Z}_p^*$ and a query for T_j is answered by $W_{j,b} = \rho \cdot f_{S_b \setminus T_j}(a) P = f_{T_j}(a)^{-1} \cdot C_b$. Again for both $b \in \{0,1\}$, $\mathcal{A}$ receives a uniformly random element C_b and query replies that do not depend on b; the bit b is thus information-theoretically hidden from $\mathcal{A}$. □

5. Building an ABC System

In this section, we present an application of SPS-EQ and set commitments introduced in the two previous sections; we use them as basic building blocks for an attribute-based credential system. ABC systems are usually constructed in one of two ways. They can be built from blind signatures: a user obtains a blind signature from an issuer on (commitments to) attributes and later shows the signature, provides the shown attributes and proves knowledge of all unrevealed attributes [24,28,55]. The drawback of this approach is that such credentials can only be shown once in an unlinkable fashion (*one-show*).

Anonymous credentials supporting an arbitrary number of unlinkable showings (*multi-show*) can be obtained in a similar vein using a different type of signatures: A user obtains a signature on (commitments to) attributes, then *randomizes* the signature (so that the resulting signature is unlinkable to the issued one) and proves in zero-knowledge the correspondence of this signature to the shown attributes as well as the undisclosed attributes [41,42].[1] Our approach also achieves multi-show ABCs, but differs from the latter. We randomize both the signature and the message (which is a set commitment to attributes) and then use subset-opening of set commitments for selective constant-size showings of attributes. We thereby completely avoid costly ZKPoKs

[1] More generally, the user could prove knowledge of a signature without revealing it. Although this can be a significant performance bottleneck, this allows for using ABCs with conventional signatures such as ECDSA, as in [36].

over the attributes, which in all other existing approaches require communication and typically also computation in the number of shown/encoded attributes.

We start by discussing the functionality and security of ABCs in Sects. 5.1 and 5.2. After providing some intuition for our construction (Sect. 5.3), we present the scheme (Sect. 5.4) and analyze its security (Sect. 5.5). Finally, we give a performance and functionality comparison with other schemes in Sect. 5.6.

5.1. *Model of ABCs*

In an ABC system, there are different organizations issuing credentials to users. These users can then anonymously demonstrate possession of their credentials to verifiers. The system is called *multi-show* when transactions (issuing and showings) performed by the same user cannot be linked. A credential cred for user i is issued by an organization for a set of attributes $\mathtt{A}$ and the user can show a subset $\mathtt{D}$ of $\mathtt{A}$ while hiding the other attributes. Note that in our definition there is no setup and we do not assume any trusted parameters at all.

Definition 26. (*ABC system*) An *attribute-based anonymous credentials system* consists of the following PPT algorithms:

$\mathsf{OrgKeyGen}(1^\kappa, 1^t)$: A probabilistic algorithm that gets (unary representations of) a security parameter κ and an upper bound t for the size of attribute sets. It outputs a key pair $(\mathsf{osk}, \mathsf{opk})$ for an organization.

$\mathsf{UserKeyGen}(\mathsf{opk})$: A probabilistic algorithm that gets an organization public key and outputs a key pair $(\mathsf{usk}, \mathsf{upk})$ for a user.

$\big(\mathsf{Obtain}(\mathsf{usk}, \mathsf{opk}, \mathtt{A}), \mathsf{Issue}(\mathsf{upk}, \mathsf{osk}, \mathtt{A})\big)$: These algorithms are run by a user and an organization, respectively, who interact during execution. Obtain is a probabilistic algorithm that takes as input the user's secret key usk, an organization's public key opk and a non-empty attribute set $\mathtt{A}$ of size $|\mathtt{A}| \leq t$. Issue is a probabilistic algorithm that takes as input a user public key upk, the organization's secret key osk and a non-empty attribute set $\mathtt{A}$ of size $|\mathtt{A}| \leq t$. At the end of this protocol, Obtain outputs a credential cred for the user for attributes $\mathtt{A}$ or $\perp$ if the execution failed.

$\big(\mathsf{Show}(\mathsf{opk}, \mathtt{A}, \mathtt{D}, \mathsf{cred}), \mathsf{Verify}(\mathsf{opk}, \mathtt{D})\big)$: These algorithms are run by a user and a verifier, respectively, who interact during execution. Show is a probabilistic algorithm that takes as input the organization public key opk, an attribute set $\mathtt{A}$ of size $|\mathtt{A}| \leq t$, a non-empty set $\mathtt{D} \subseteq \mathtt{A}$ (representing the attributes to be shown) and a credential cred. Verify is a deterministic algorithm that takes as input the organization's public key opk and a set $\mathtt{D}$. At the end of the protocol, Verify outputs 1 or 0 indicating whether it accepts the credential showing or not.

5.2. *Security of ABCs*

We present a security model for multi-show ABCs, which is game based and in the spirit of group signatures [29] and considers malicious organization keys. We note that at the time of designing our model, there were no other comprehensive models for

ABC systems.[2] We start with a high-level overview of the required security properties and note that we consider only a single organization in our model of unforgeability and anonymity (since all organizations have independent signing keys, an extension to multiple organizations is straightforward):

Correctness: A showing of a credential with respect to a non-empty set $\mathtt{D}$ of attributes always verifies if the credential was issued honestly for some attribute set $\mathtt{A}$ with $\mathtt{D} \subseteq \mathtt{A}$.

Unforgeability: A user cannot perform a valid showing of attributes for which she does not possess a credential. Moreover, no coalition of malicious users can combine their credentials and prove possession of a set of attributes which no single member has. This holds even after seeing showings of arbitrary credentials by honest users (the notion thus covers replay attacks).

Anonymity: During a showing, no verifier and no (malicious) organization (even if they collude) is able to identify the user or learn anything about the user, except that she owns a valid credential for the shown attributes. Furthermore, different showings of the same credential are unlinkable.

We now provide formal definitions of these properties, for which we introduce the following global variables and oracles.

Global Variables. At the beginning of each experiment, either the experiment computes an organization key pair $(\mathsf{osk}, \mathsf{opk})$ or the adversary outputs opk. In the anonymity game, there is a bit b, which the adversary must guess.

In order to keep track of all honest and corrupt users, we introduce the sets $\mathtt{HU}$, and $\mathtt{CU}$, respectively. We use the lists $\mathtt{UPK}$, $\mathtt{USK}$, $\mathtt{CRED}$, $\mathtt{ATTR}$ and $\mathtt{OWNR}$ to track user public and secret keys, issued credentials and corresponding attributes and to which user they were issued. Furthermore, we use the sets J_{LoR} and I_{LoR} to store the issuance indices and corresponding users that have been set during the first call to the left-or-right oracle in the anonymity game.

Oracles. The oracles are as follows:

$\mathcal{O}_{\mathtt{HU}}(i)$ takes as input a user identity i. If $i \in \mathtt{HU} \cup \mathtt{CU}$, it returns $\perp$. Otherwise, it creates a new honest user i by running $(\mathtt{USK}[i], \mathtt{UPK}[i]) \xleftarrow{R} \mathsf{UserKeyGen}(\mathsf{opk})$, adding i to $\mathtt{HU}$ and returning $\mathtt{UPK}[i]$.

$\mathcal{O}_{\mathtt{CU}}(i, \mathsf{upk})$ takes as input a user identity i and (optionally) a user public key upk; if user i does not exist yet, a new corrupt user with public key upk is registered, while if i is honest, its secret key and all credentials are leaked.

In particular, if $i \in \mathtt{CU}$ or if $i \in I_{LoR}$ (that is, i is a challenge user in the anonymity game), then the oracle returns $\perp$. If $i \in \mathtt{HU}$, then the oracle removes i from $\mathtt{HU}$ and adds it to $\mathtt{CU}$; it returns $\mathtt{USK}[i]$ and $\mathtt{CRED}[j]$ for all j with $\mathtt{OWNR}[j] = i$. Otherwise (i.e., $i \notin \mathtt{HU} \cup \mathtt{CU}$), it adds i to $\mathtt{CU}$ and sets $\mathtt{UPK}[i] \leftarrow \mathsf{upk}$.

$\mathcal{O}_{\mathsf{ObtIss}}(i, \mathtt{A})$ takes as input a user identity i and a set of attributes $\mathtt{A}$. If $i \notin \mathtt{HU}$, it returns $\perp$. Otherwise, it issues a credential to i by running

[2] As already mentioned earlier, there are independently (and subsequently) developed very strong simulation-based models in [31,37].

$$(\mathsf{cred}, \top) \xleftarrow{R} \big(\mathsf{Obtain}(\mathtt{USK}[i], \mathsf{opk}, \mathtt{A}), \mathsf{Issue}(\mathtt{UPK}[i], \mathsf{osk}, \mathtt{A})\big) .$$

If $\mathsf{cred} = \bot$, it returns $\bot$. Else, it appends $(i, \mathsf{cred}, \mathtt{A})$ to $(\mathtt{OWNR}, \mathtt{CRED}, \mathtt{ATTR})$[3] and returns $\top$.

$\mathcal{O}_{\mathsf{Obtain}}(i, \mathtt{A})$ lets the adversary, who in the anonymity game impersonates a malicious organization, issue a credential to an honest user. It takes as input a user identity i and a set of attributes $\mathtt{A}$. If $i \notin \mathtt{HU}$, it returns $\bot$. Otherwise, it runs

$$(\mathsf{cred}, \,\cdot\,) \xleftarrow{R} \big(\mathsf{Obtain}(\mathtt{USK}[i], \mathsf{opk}, \mathtt{A}), \,\cdot\,\big) ,$$

where the Issue part is executed by the adversary. If $\mathsf{cred} = \bot$, it returns $\bot$. Else, it appends $(i, \mathsf{cred}, \mathtt{A})$ to $(\mathtt{OWNR}, \mathtt{CRED}, \mathtt{ATTR})$ and returns $\top$.

$\mathcal{O}_{\mathsf{Issue}}(i, \mathtt{A})$ lets the adversary, who in the unforgeability game can impersonate a malicious user, obtain a credential from an honest organization. It takes as input a user identity i and a set of attributes $\mathtt{A}$. If $i \notin \mathtt{CU}$, it returns $\bot$. Otherwise, it runs

$$(\,\cdot\,, I) \xleftarrow{R} \big(\,\cdot\,, \mathsf{Issue}(\mathtt{UPK}[i], \mathsf{osk}, \mathtt{A})\big) ,$$

where the Obtain part is executed by the adversary. If $I = \bot$, it returns $\bot$. Else, it appends $(i, \bot, \mathtt{A})$ to $(\mathtt{OWNR}, \mathtt{CRED}, \mathtt{ATTR})$ and returns $\top$.

$\mathcal{O}_{\mathsf{Show}}(j, \mathtt{D})$ lets the adversary play a dishonest verifier in a credential showing by an honest user. It takes as input an index of an issuance j and a set of attributes $\mathtt{D}$. Let $i \leftarrow \mathtt{OWNR}[j]$. If $i \notin \mathtt{HU}$, it returns $\bot$. Otherwise, it runs

$$(S, \,\cdot\,) \xleftarrow{R} \big(\mathsf{Show}(\mathsf{opk}, \mathtt{ATTR}[j], \mathtt{D}, \mathtt{CRED}[j]), \,\cdot\,\big) ,$$

where the Verify part is executed by adversary.

$\mathcal{O}_{LoR}(j_0, j_1, \mathtt{D})$ is the challenge oracle in the anonymity game where the adversary must distinguish (multiple) showings of two credentials $\mathtt{CRED}[j_0]$ and $\mathtt{CRED}[j_1]$. The oracle takes two issuance indices j_0 and j_1 and a set of attributes $\mathtt{D}$. If $J_{LoR} \neq \emptyset$ and $J_{LoR} \neq \{j_0, j_1\}$, it returns $\bot$. Let $i_0 \leftarrow \mathtt{OWNR}[j_0]$ and $i_1 \leftarrow \mathtt{OWNR}[j_1]$. If $J_{LoR} = \emptyset$, then it sets $J_{LoR} \leftarrow \{j_0, j_1\}$ and $I_{LoR} \leftarrow \{i_0, i_1\}$. If $i_0, i_1 \notin \mathtt{HU}$ or $\mathtt{D} \not\subseteq \mathtt{ATTR}[j_0] \cap \mathtt{ATTR}[j_1]$, it returns $\bot$. Else, it runs

$$(S, \,\cdot\,) \xleftarrow{R} \big(\mathsf{Show}(\mathsf{opk}, \mathtt{ATTR}[j_b], \mathtt{D}, \mathtt{CRED}[j_b]), \,\cdot\,\big) ,$$

(with b set by the experiment) where the Verify part is executed by the adversary.
Using the global variables and oracles just defined, we now define security of an ABC system:

Definition 27. (*Correctness*) An ABC system is *correct*, if for all $\kappa > 0$, all $t > 0$ and all $\mathtt{A}$ with $0 < |\mathtt{A}| \leq t$ and all $\emptyset \neq \mathtt{D} \subseteq \mathtt{A}$ it holds that:

[3] We use this as a shorthand for “appends i to $\mathtt{OWNR}$, cred to $\mathtt{CRED}$ and $\mathtt{A}$ to $\mathtt{ATTR}$.

$$\Pr\left[\begin{array}{l}(\mathsf{osk},\mathsf{opk}) \overset{R}{\leftarrow} \mathsf{OrgKeyGen}(1^\kappa, 1^t),\\ (\mathsf{usk},\mathsf{upk}) \overset{R}{\leftarrow} \mathsf{UserKeyGen}(\mathsf{opk}),\\ (\mathsf{cred}, \top) \overset{R}{\leftarrow} (\mathsf{Obtain}(\mathsf{usk},\mathsf{opk},\mathtt{A}),\\ \mathsf{Issue}(\mathsf{upk},\mathsf{osk},\mathtt{A}))\end{array} : \begin{array}{r}(\top, 1) \overset{R}{\leftarrow} (\mathsf{Show}(\mathsf{opk},\mathtt{A},\mathtt{D},\mathsf{cred}),\\ \mathsf{Verify}(\mathsf{opk},\mathtt{D}))\end{array}\right] = 1\,.$$

Definition 28. (*Unforgeability*) An ABC system is *unforgeable*, if for all $t > 0$ and all PPT adversaries $\mathcal{A}$ having oracle access to $\mathcal{O} := \{\mathcal{O}_{\mathtt{HU}}, \mathcal{O}_{\mathtt{CU}}, \mathcal{O}_{\mathsf{ObtIss}}, \mathcal{O}_{\mathsf{Issue}}, \mathcal{O}_{\mathsf{Show}}\}$, there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{l}(\mathsf{osk},\mathsf{opk}) \overset{R}{\leftarrow} \mathsf{OrgKeyGen}(1^\kappa, 1^t),\\ (\mathtt{D}, \mathsf{st}) \overset{R}{\leftarrow} \mathcal{A}^{\mathcal{O}}(\mathsf{opk}),\\ (\cdot, b^*) \overset{R}{\leftarrow} (\mathcal{A}(\mathsf{st}), \mathsf{Verify}(\mathsf{opk},\mathtt{D}))\end{array} : \begin{array}{l} b^* = 1 \ \wedge \\ \forall j : \mathtt{OWNR}[j] \in \mathtt{CU} \\ \qquad \Rightarrow \mathtt{D} \not\subseteq \mathtt{ATTR}[j]\end{array}\right] \le \epsilon(\kappa)\,.$$

Definition 29. (*Anonymity*) An ABC system is *anonymous*, if for all $t > 0$ and all PPT adversaries $\mathcal{A}$ having oracle access to $\mathcal{O} := \{\mathcal{O}_{\mathtt{HU}}, \mathcal{O}_{\mathtt{CU}}, \mathcal{O}_{\mathsf{Obtain}}, \mathcal{O}_{\mathsf{Show}}, \mathcal{O}_{LoR}\}$, there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{l} b \overset{R}{\leftarrow} \{0,1\},\ (\mathsf{opk},\mathsf{st}) \overset{R}{\leftarrow} \mathcal{A}(1^\kappa, 1^t),\\ b^* \overset{R}{\leftarrow} \mathcal{A}^{\mathcal{O}}(\mathsf{st})\end{array} : b^* = b\right] - \frac{1}{2} \le \epsilon(\kappa)\,.$$

5.3. *Intuition of Our Construction*

Our construction of ABCs is based on SPS-EQ, on set commitments with subset-openings and on a *single* constant-size proof of knowledge for proving freshness. In contrast to this, the proofs of knowledge in existing ABC systems [28,40–44] require computation and communication that is linear in the number of shown (or even issued) attributes. However, aside from selective disclosure of attributes, they usually allow to prove statements about non-revealed attribute values, such as AND, OR and NOT, interval proofs, as well as conjunctions and disjunctions of the aforementioned. We achieve less expressiveness; our construction supports selective disclosure as well as AND statements about attributes (as the constructions in [31,43,44], of which only the latter also achieves constant-size showings). A user can thus either open some attributes and their corresponding values or solely prove that some attributes are encoded in the respective credential without revealing their concrete values. Note that one can always associate sets of values to attributes, so that users are not required to reveal the full attribute value, but only predefined “statements” about the attribute value, e.g., “01.01.1980”, “> 16” or “> 18” for an attribute label `birthdate`. This allows emulation of proving properties about attribute values.

Example. To give an idea of the expressiveness of our construction, we include an example of an attribute set $\mathtt{A}$. We are given a user with the following set of attribute and value strings:

$$\mathtt{A} = \{\text{“}\mathtt{gender, male}\text{"}, \text{“}\mathtt{birthdate, 01.01.1980}\text{"}, \\ \text{“}\mathtt{drivinglicense, \#}\text{"}, \text{“}\mathtt{drivinglicense, car}\text{"}\}\,.$$

Note that # indicates an attribute value that allows to prove possession of the attribute without revealing any concrete value. A showing could, for instance, involve the following attributes $\mathtt{D}$ and its hidden complement $\mathtt{A} \setminus \mathtt{D}$:

$$\mathtt{D} = \{\text{“}\mathtt{gender, male}\text{"}, \text{“}\mathtt{drivinglicense, \#}\text{"}\}$$
$$\mathtt{A} \setminus \mathtt{D} = \{\text{“}\mathtt{birthdate, 01.01.1980}\text{"}, \text{“}\mathtt{drivinglicense, car}\text{"}\}\ .$$

Outline. We assume attributes to be values from $\mathbb{Z}_p$ and note that we can define attributes of arbitrary format by using a collision-resistant hash function $H\colon \{0,1\}^* \to \mathbb{Z}_p$. In our construction, a credential cred of user i consists of an element C from a bilinear group, a scalar $r \in \mathbb{Z}_p^*$, an opening O of C and an SPS-EQ signature σ on $(C, r \cdot C, P)$. The element C is a set commitment to a set of attributes $\mathtt{A} \subset \mathbb{Z}_p$, whose randomness is the user secret usk (thus, its opening O contains usk or the commitment trapdoor a, if $a \in \mathtt{A}$). When obtaining a credential, the user performs a ZKPoK $\Pi^{\mathcal{R}_\mathsf{U}}(\mathsf{upk})$ to prove knowledge of usk, which allows us to extract usk for corrupt users in the proof of unforgeability.

The values C and r define an equivalence class $[(C, r \cdot C, P)]_{\mathcal{R}}$ that is unique for each credential with overwhelming probability. (The scalar r and the third credential component are required to prove unforgeability.) During a showing, a random representative of this class, $(C_1, C_2, C_3) \xleftarrow{R} [(C, r \cdot C, P)]_{\mathcal{R}}$, together with an updated signature σ' is presented. The randomized commitment C_1 is then subset-opened to the shown attributes $\mathtt{D} \subseteq \mathtt{A}$ (representing selective disclosure). Hence, showings additionally include a witness W and a verifier checks whether the encodings of the disclosed attributes and W give a valid subset-opening of C_1.

Freshness. We have to prevent transcripts of valid showings from being replayed by someone not in possession of the credential. To this end, we require the user to conduct an (interactive) proof of knowledge $\mathsf{PoK}\{\beta \mid C_3 = \beta P\}$ of the discrete logarithm of the third component $C_3 = \mu P$ of a shown credential $\mathsf{cred}' = ((C_1, C_2, C_3), \sigma')$, i.e., the randomizer μ used in the showing protocol, which provides a fresh challenge for every showing. For the unforgeability reduction, we have the user additionally prove knowledge of $r = \log_{C_1} C_2$ by conducting a proof of knowledge $\mathsf{PoK}\{\alpha \mid C_2 = \alpha C_1\}$. We use the compact notation $\Pi^{\mathcal{R}_\mathsf{F}}(C_1, C_2, C_3)$ for the AND-composition of both proofs, i.e., $\Pi^{\mathcal{R}_\mathsf{F}}(C_1, C_2, C_3) := \mathsf{PoK}\{(\alpha, \beta) \mid C_2 = \alpha C_1 \ \wedge\ C_3 = \beta P\}$.

Malicious Organization Keys. In contrast to anonymity notions usually considered for ABCs, our model guarantees anonymity even against adversaries that generate the organization keys maliciously. Our construction is in the standard model and organization public keys consist of an SPS-EQ public key pk and set-commitment parameters pp_sc. We augment the issuing protocol sketched above and let the (malicious) organization prove knowledge of a secret key that matches its public key to the user (which allows us to extract the signing key in the anonymity proof).

For an SPS-EQ scheme $\mathsf{SPS\text{-}EQ}$, we define an NP-relation $\mathcal{R}_\mathsf{O}$, whose statements and witnesses are organization public and private keys, i.e.: $(\mathsf{pk}, \mathsf{sk}) \in \mathcal{R}'_\mathsf{O} \Leftrightarrow \mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk}) = 1$. In our proof of anonymity, we also need to extract the set-commitment trapdoor $a \in \mathbb{Z}_p$, so we augment the above relation to:

$$((aP, \mathsf{pk}), (w_1, w_2)) \in \mathcal{R}_\mathsf{O} \iff (aP = w_1 P \ \wedge\ \mathsf{VKey}_{\mathcal{R}}(w_2, \mathsf{pk}) = 1)\ ,$$

where aP is from the set-commitment parameters $\mathsf{pp}_{\mathsf{sc}}$ contained in opk. For compactness, we use the notation $\Pi^{\mathcal{R}_{\mathsf{O}}}(\mathsf{opk})$ and require the proof to be a perfect zero-knowledge proof of knowledge.

ZKPoKs and Concurrent Security. We will consider all ZKPoKs in a black-box way and assume that they are 4-move ZKPoK proofs from [32], which are based on Σ-protocols and feature rewindable black-box access to the verifier (for perfect zero-knowledge) and the prover (for knowledge soundness), respectively.

Note, however, that the ZKPoKs from [32] are not concurrently secure and so neither is any instantiation of Scheme 2 using them. Thus, each organization, each user and each verifier must not run more than one protocol execution at once. We will briefly discuss the idea of a concurrently secure scheme variant in the CRS model in Remark 1.

5.4. *The Construction of the ABC System*

Our ABC construction is based on any perfectly adapting structure-preserving signature scheme on equivalence classes and the set-commitment scheme from Sect. 4.2 and is described in Scheme 2 (Fig. 2). In particular, since the organization public key is fully determined by the adversary (for malicious-key anonymity), we assume the bilinear-group generation algorithm inside the set-commitment-setup algorithm to be deterministic[4] and produce the same bilinear group for each security parameter.[5] We will base our proofs on assumptions that are modified accordingly, i.e., that are with respect to a *deterministic* BGGen producing one bilinear group per security parameter.

Randomizable Set Commitment. The instantiation of set commitments presented in Sect. 4.2 is randomizable in the sense that commitments as well as subset-opening witnesses can be consistently randomized. For a compact presentation of our ABC construction and to smoothly integrate the set-commitment scheme with the $\mathsf{SPS\text{-}EQ}$ scheme, in Scheme 2 we make the randomness ρ of the Commit algorithm explicit, i.e., write $\mathsf{Commit}(\mathsf{pp}, S; \rho)$. We also stress that in $(\mathsf{Show}, \mathsf{Verify})$ after the $\mathsf{OpenSubset}$ algorithm has been run, we randomize the witness W using μ to obtain W'. Observe that the resulting witness is then consistently randomized with the set commitment C_1.

Optimizations. Note that the first move in the showing protocol can be combined with the first move of $\Pi^{\mathcal{R}_{\mathsf{F}}}$, meaning the showing protocol consists of a total of 4 moves, when using 4-move ZKPoKs. Furthermore, note that issuing can be made more efficient with regard to both communication complexity and computational effort, as osk contains set-commitment trapdoor a: instead of using a pairing to check C for consistency, the issuer can compute it herself as $C \leftarrow f_{\mathbb{A}}(a) \cdot \mathsf{upk}$. (We wrote our scheme so that a is never used and $\mathsf{pp}_{\mathsf{sc}}$ can then be moved to public parameters in the concurrently secure variant discussed below.)

[4] This assumption was also made by Bellare et al. [23] and is justified by actual implementations. For example, BN-curves [25], the most common choice for Type-3 pairings, are generated deterministically.

[5] Hence, the only random choice made by the set-commitment setup algorithm is picking the commitment trapdoor a. Inside $\mathsf{OrgKeyGen}$, we will make this randomness explicit.

$\underline{\mathsf{OrgKeyGen}}(1^\kappa, 1^t)$: Given $\kappa, t > 0$, compute $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) \leftarrow \mathsf{BGGen}(1^\kappa)$; pick $a \overset{R}{\leftarrow} \mathbb{Z}_p$, run $\mathsf{pp_{sc}} = (\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]}) \leftarrow \mathsf{Setup}(1^\kappa, 1^t; a)$, which defines $\mathcal{S}_{\mathsf{pp}} \leftarrow \{\mathtt{A} \subset \mathbb{Z}_p \mid 0 < |\mathtt{A}| \leq t\}$.
Run $(\mathsf{sk}, \mathsf{pk}) \overset{R}{\leftarrow} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^\ell)$ for $\ell = 3$ and return $(\mathsf{osk}, \mathsf{opk}) \leftarrow ((a, \mathsf{sk}), (\mathsf{pp_{sc}}, \mathsf{pk}))$.

$\underline{\mathsf{UserKeyGen}}(\mathsf{opk})$: From opk derive security parameter $\kappa > 0$, deterministically compute $\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$, pick $\mathsf{usk} \overset{R}{\leftarrow} \mathbb{Z}_p^*$, set $\mathsf{upk} \leftarrow \mathsf{usk} \cdot P$ and return $(\mathsf{usk}, \mathsf{upk})$.

$\underline{(\mathsf{Obtain}, \mathsf{Issue})}$: Using $\Pi^{\mathcal{R}_\mathsf{O}}\big(\mathsf{opk} = ((\mathsf{BG}, (a^i P, a^i \hat{P})_i), \mathsf{pk})\big) := \mathsf{PoK}\big\{(\alpha, \vec{\beta}) \mid \alpha P = aP \wedge \mathsf{VKey}_{\mathcal{R}}(\vec{\beta}, \mathsf{pk}) = 1\big\}$ and $\Pi^{\mathcal{R}_\mathsf{U}}(\mathsf{upk}) := \mathsf{PoK}\{\alpha \mid \alpha P = \mathsf{upk}\}$, Obtain and Issue interact as follows:

$\mathsf{Obtain}(\mathsf{usk}, \mathsf{opk}, \mathtt{A})$		$\mathsf{Issue}(\mathsf{upk}, \mathsf{osk}, \mathtt{A})$
$\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$		If $\mathtt{A} \notin \mathcal{S}_{\mathsf{pp}}$, return $\perp$
If $\mathtt{A} \notin \mathcal{S}_{\mathsf{pp}}$, return $\perp$		
	$\xleftrightarrow{\Pi^{\mathcal{R}_\mathsf{U}}(\mathsf{upk})}$	If $\Pi^{\mathcal{R}_\mathsf{U}}(\mathsf{upk})$ fails, return $\perp$
If $\Pi^{\mathcal{R}_\mathsf{O}}(\mathsf{opk})$ fails, return $\perp$	$\xleftrightarrow{\Pi^{\mathcal{R}_\mathsf{O}}(\mathsf{opk})}$	
$(C, O) \leftarrow \mathsf{Commit}(\mathsf{pp_{sc}}, \mathtt{A}; \mathsf{usk})$		
$r \overset{R}{\leftarrow} \mathbb{Z}_p^*,\ R \leftarrow r \cdot C$	$\xrightarrow{C, R}$	If $e(C, \hat{P}) \neq e(\mathsf{upk}, f_{\mathtt{A}}(a)\hat{P})$ and $\forall a' \in \mathtt{A} : a'P \neq aP$ then return $\perp$
If $\mathsf{Verify}_{\mathcal{R}}((C, R, P), \sigma, \mathsf{pk}) = 0$ return $\perp$	$\xleftarrow{\sigma}$	Else $\sigma \overset{R}{\leftarrow} \mathsf{Sign}_{\mathcal{R}}((C, R, P), \mathsf{sk})$
Else return $\mathsf{cred} \leftarrow (C, \sigma, r, O)$		

$\underline{(\mathsf{Show}, \mathsf{Verify})}$: Using $\Pi^{\mathcal{R}_\mathsf{F}}(C_1, C_2, C_3) := \mathsf{PoK}\{(\alpha, \beta) \mid C_2 = \alpha C_1 \wedge C_3 = \beta P\}$, Show and Verify interact as follows:

$\mathsf{Show}(\mathsf{opk}, \mathtt{A}, \mathtt{D}, \mathsf{cred})$		$\mathsf{Verify}(\mathsf{opk}, \mathtt{D})$
Let $\mathsf{cred} = (C, \sigma, r, O)$; $\mu \overset{R}{\leftarrow} \mathbb{Z}_p^*$		Let $\mathsf{opk} = (\mathsf{pp_{sc}}, \mathsf{pk})$
$\sigma' \overset{R}{\leftarrow} \mathsf{ChgRep}_{\mathcal{R}}((C, r \cdot C, P), \sigma, \mu, \mathsf{pk})$		
$\mathsf{cred}' \leftarrow \big((C_1, C_2, C_3) = \mu \cdot (C, r \cdot C, P), \sigma'\big)$		
$W \leftarrow \mathsf{OpenSubset}(\mathsf{pp_{sc}}, C, \mathtt{A}, O, \mathtt{D})$		
$W' \leftarrow \mu \cdot W$	$\xrightarrow{\mathsf{cred}', W'}$	
	$\xleftrightarrow{\Pi^{\mathcal{R}_\mathsf{F}}(C_1, C_2, C_3)}$	If $\Pi^{\mathcal{R}_\mathsf{F}}(C_1, C_2, C_3)$ fails, return 0
		Return $\big(\mathsf{Verify}_{\mathcal{R}}(\mathsf{cred}', \mathsf{pk}) \wedge \mathsf{VerifySubset}(\mathsf{pp_{sc}}, C_1, \mathtt{D}, W')\big)$

Fig. 2. Scheme 2, a multi-show ABC system.

5.5. *Security*

The correctness of Scheme 2 follows by inspection.

Theorem 7. *Let $\Pi^{\mathcal{R}_\mathsf{F}}$, $\Pi^{\mathcal{R}_\mathsf{U}}$ and $\Pi^{\mathcal{R}_\mathsf{O}}$ be ZKPoKs. If the t-co-DL assumption holds,* SC *is subset-sound and* $\mathsf{SPS\text{-}EQ}$ *is EUF-CMA-secure, then Scheme 2 is unforgeable.*

In the proof of unforgeability, we distinguish whether the adversary wins the game by forging a signature, breaking subset-opening soundness of the commitment scheme or computing a discrete logarithm. We can efficiently determine which was the case since the knowledge extractor of the ZKPoK $\Pi^{\mathcal{R}_\mathsf{F}}$ lets us extract the credential used by the adversary.

Proof (of Theorem 7). We first introduce the following syntactic changes to the experiment, which let us distinguish different types of forgeries: (1) We include the value R in credentials cred output by Obtain (these belong to honest users and are now of the form $\mathsf{cred} = ((C, R), \sigma, r, O))$. (2) When the adversary makes a valid call to $\mathcal{O}_{\mathsf{Issue}}$, the experiment receives the values C, R and produces a signature σ; instead of appending $\perp$ to the list $\mathtt{CRED}$, the oracle now appends $((C, R), \sigma, \perp, \perp)$. Note that the adversary's view in the experiment remains unchanged.

Assume now an efficient adversary $\mathcal{A}$ wins the unforgeability game (Definition 28) with non-negligible probability, and let $((C_1^*, C_2^*, C_3^*), \sigma^*)$ be the message-signature pair it uses and W^* be the witness for an attribute set $\mathtt{D}^* \not\subseteq \mathtt{ATTR}[j]$, for all j with $\mathtt{OWNR}[j] \in \mathtt{CU}$; moreover, the ZKPoK $\Pi^{\mathcal{R}_\mathsf{F}}(C_1^*, C_2^*, C_3^*)$ verifies. We distinguish the following cases:

Type 1: $[(C_1^*, C_2^*, C_3^*)]_\mathcal{R} \neq [(C, R, P)]_\mathcal{R}$ for $((C, R), \sigma, *, *) = \mathtt{CRED}[j]$ for all issuance indices j (i.e., $\mathtt{OWNR}[j] \in \mathtt{HU} \cup \mathtt{CU}$). The pair $((C_1^*, C_2^*, C_3^*), \sigma^*)$ is thus a signature forgery and using $\mathcal{A}$ we construct an adversary $\mathcal{B}$ that breaks the EUF-CMA security of the SPS-EQ scheme.

Type 2: $[(C_1^*, C_2^*, C_3^*)]_\mathcal{R} = [(C, R, P)]_\mathcal{R}$ where $((C, R), \sigma, *, *) = \mathtt{CRED}[j]$ for some index j with $\mathtt{OWNR}[j] \in \mathtt{CU}$. Since $\mathcal{A}$ only wins if $\mathtt{D} \not\subseteq \mathtt{ATTR}[j]$, it must have broken subset soundness. We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that breaks subset soundness of the set-commitment scheme SC.

Type 3: $[(C_1^*, C_2^*, C_3^*)]_\mathcal{R} = [(C, R, P)]_\mathcal{R}$ where $((C, R), \sigma, r, O) = \mathtt{CRED}[j]$ for some index j with $\mathtt{OWNR}[j] \in \mathtt{HU}$. Then, we use $\mathcal{A}$ to break q-co-DL.

Type 1. This reduction is straightforward. $\mathcal{B}$ interacts with a challenger $\mathcal{C}$ in the EUF-CMA game for SPS-EQ and $\mathcal{B}$ simulates the ABC-unforgeability game for $\mathcal{A}$.

$\mathcal{C}$ runs $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_\mathcal{R}(\mathsf{BG}, 1^3)$ and gives pk to $\mathcal{B}$. Then, $\mathcal{B}$ picks $a \xleftarrow{R} \mathbb{Z}_p$, defines $\mathsf{pp}_\mathsf{sc} \leftarrow (\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]})$ and sets $(\mathsf{osk}, \mathsf{opk}) \leftarrow ((a, \perp), (\mathsf{pp}_\mathsf{sc}, \mathsf{pk}))$. It next runs $\mathcal{A}(\mathsf{opk})$ and simulates the environment and the oracles. All oracles are executed as in the real game, except for the following oracles, which use the signing oracle instead of the signing key sk:

$\mathcal{O}_{\mathsf{ObtIss}}(i, \mathtt{A})$: $\mathcal{B}$ computes $(C, O) \xleftarrow{R} \mathsf{Commit}(\mathsf{pp}_\mathsf{sc}, \mathtt{A}, \mathtt{USK}[i])$, picks $r \xleftarrow{R} \mathbb{Z}_p^*$ and then queries its oracle $\mathsf{Sign}_\mathcal{R}(\cdot, \mathsf{sk})$ on $(C, r \cdot C, P)$ to obtain σ; $\mathcal{B}$ appends $(i, ((C, r \cdot C), \sigma, r, O), \mathtt{A})$ to $(\mathtt{OWNR}, \mathtt{CRED}, \mathtt{ATTR})$.

$\mathcal{O}_{\mathsf{Issue}}(i, \mathtt{A})$: $\mathcal{B}$ runs this oracle by running the simulator $\mathcal{S}$ of ZKPoK $\Pi^{\mathcal{R}_\mathsf{O}}(\mathsf{opk})$ (as it does not know $\mathsf{sk} = \mathsf{osk}[2]$), and instead of signing (C, R, P), $\mathcal{B}$ obtains the signature σ from $\mathcal{C}$'s signing oracle. If successful, $\mathcal{B}$ appends $(i, ((C, R), \sigma, \perp, \perp), \mathtt{A})$ to $(\mathtt{OWNR}, \mathtt{CRED}, \mathtt{ATTR})$ and returns $\top$.

Note that by perfect zero-knowledge of $\Pi^{\mathcal{R}_{\mathsf{O}}}(\mathsf{opk})$ the simulation of $\mathcal{O}_{\mathsf{Issue}}$ is perfect, and so is that of $\mathcal{O}_{\mathsf{ObtIss}}$. When $\mathcal{A}$ outputs $(\mathtt{D}^*, \mathsf{st})$, $\mathcal{B}$ runs $\mathcal{A}(\mathsf{st})$ and interacts with $\mathcal{A}$ as verifier in a showing protocol. If $\mathcal{A}$ delivers a valid showing using $((C_1^*, C_2^*, C_3^*), \sigma^*)$ and conducting $\Pi^{\mathcal{R}_{\mathsf{F}}}(C_1^*, C_2^*, C_3^*)$, then $\mathcal{B}$ runs the knowledge extractor of $\Pi^{\mathcal{R}_{\mathsf{F}}}$ to obtain a witness $w = (r'', \mu)$ with $C_3^* = \mu P$. If there is a credential $\perp \neq ((C', R'), \sigma', *, *) \in \mathtt{CRED}$ such that $(C', R', P) = \mu^{-1} \cdot (C_1^*, C_2^*, C_3^*)$, then $\mathcal{B}$ aborts. (In this case, the forgery is not of Type 1.) Otherwise, $\mathcal{B}$ has never queried a signature for class $[(C_1^*, C_2^*, C_3^*)]_{\mathcal{R}}$ and outputs $((C_1^*, C_2^*, C_3^*), \sigma^*)$, which is a forgery. $\mathcal{B}$ breaks thus EUF-CMA of $\mathsf{SPS\text{-}EQ}$.

Type 2. $\mathcal{B}$ interacts with the challenger $\mathcal{C}$ in the subset-soundness game for SC for some $t > 0$. First, $\mathcal{C}$ generates set-commitment parameters $\mathsf{pp_{sc}} \leftarrow (\mathsf{BG}, (a^i P, a^i \hat{P})_{i \in [t]})$ with $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) = \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$ and sends $\mathsf{pp_{sc}}$ to $\mathcal{B}$. $\mathcal{B}$ generates a key pair $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^3)$, sets $(\mathsf{osk}, \mathsf{opk}) \leftarrow ((\perp, \mathsf{sk}), (\mathsf{pp_{sc}}, \mathsf{pk}))$ and runs $\mathcal{A}(\mathsf{opk})$, simulating the oracles. All oracles are as in the real game, except for $\mathcal{O}_{\mathsf{ObtIss}}$, in which $\mathcal{B}$ simply ignores the first two moves $\Pi^{\mathcal{R}_{\mathsf{U}}}$ and $\Pi^{\mathcal{R}_{\mathsf{O}}}$, and $\mathcal{O}_{\mathsf{Issue}}$, which is simulated as follows (as $\mathcal{B}$ does not know $a = \mathsf{osk}[1]$):

$\mathcal{O}_{\mathsf{Issue}}(i, \mathtt{A})$: The oracle is simulated as prescribed except for running the simulator for $\Pi^{\mathcal{R}_{\mathsf{O}}}(\mathsf{opk})$. When $\mathcal{A}$ conducts $\Pi^{\mathcal{R}_{\mathsf{U}}}(\mathsf{upk})$, $\mathcal{B}$ runs the extractor for $\Pi^{\mathcal{R}_{\mathsf{U}}}$ to extract usk and sets $\mathtt{USK}[i] \leftarrow \mathsf{usk}$.

By perfect zero-knowledge of $\Pi^{\mathcal{R}_{\mathsf{O}}}(\mathsf{opk})$, the simulation of the oracle $\mathcal{O}_{\mathsf{Issue}}$ is perfect. Moreover, note that $\mathcal{B}$ stores the secret keys of all users (all $i \in \mathtt{HU} \cup \mathtt{CU}$).

When $\mathcal{A}$ outputs $(\mathtt{D}^*, \mathsf{st})$, $\mathcal{B}$ runs $\mathcal{A}(\mathsf{st})$ and interacts with $\mathcal{A}$ as verifier in a showing protocol. Assume $\mathcal{A}$ delivers a valid showing using $((C_1^*, C_2^*, C_3^*), \sigma^*)$ and a witness W^* for the attribute set $\mathtt{D}^*$ such that $\mathtt{D}^* \not\subseteq \mathtt{ATTR}[j]$ for all j with $\mathtt{OWNR}[j] \in \mathtt{CU}$ and by conducting $\Pi^{\mathcal{R}_{\mathsf{F}}}(C_1^*, C_2^*, C_3^*)$. Then, $\mathcal{B}$ runs the knowledge extractor of $\Pi^{\mathcal{R}_{\mathsf{F}}}$ to obtain a witness $w = (r'', \mu)$ such that $C_3^* = \mu P$. Let $(C', R', P) = \mu^{-1} \cdot (C_1^*, C_2^*, C_3^*)$; if there is no credential $\perp \neq ((C', R'), *, *, *) \in \mathtt{CRED}$, then $\mathcal{B}$ aborts (the forgery was of Type 1). Otherwise, let j^* be such that $((C', R'), *, *, *) = \mathtt{CRED}[j^*]$. If $\mathtt{OWNR}[j^*] \in \mathtt{HU}$, then $\mathcal{B}$ aborts (the forgery was of Type 3). Else, we have $\mathtt{OWNR}[j^*] \in \mathtt{CU}$ and $\mathtt{D}^* \not\subseteq \mathtt{ATTR}[j^*]$. If for some $a' \in \mathtt{ATTR}[j^*] : a'P = aP$, then $\mathcal{B}$ sets $O^* \leftarrow (1, a')$. Else, $\mathcal{B}$ sets $O^* \leftarrow (0, \mu \cdot \mathtt{USK}[\mathtt{OWNR}[j^*]])$. $\mathcal{B}$ outputs $(C_1^*, \mathtt{ATTR}[j^*], O^*, \mathtt{D}^*, W^*)$, which satisfies $\mathtt{D}^* \not\subseteq \mathtt{ATTR}[j^*] \neq \perp$ and $\mathsf{VerifySubset}(\mathsf{pp_{sc}}, C_1^*, \mathtt{D}^*, W^*) = 1$. $\mathcal{B}$'s output breaks thus subset soundness of SC.

Type 3. We assume the forgery to be of Type 3 and use a sequence of games which are indistinguishable under q-co-DL. Henceforth, we denote the event that an adversary wins Game i by S_i.

Game 0: The original game, which only outputs 1 if the forgery is of Type 3.
Game 1: As Game 0, except for the following oracles:

$\mathcal{O}_{\mathsf{ObtIss}}(i, \mathtt{A})$: As in Game 0, except that the experiment aborts if set-commitment trapdoor a is contained in $\mathtt{A}$.
$\mathcal{O}_{\mathsf{Issue}}(i, \mathtt{A})$: Analogous to $\mathcal{O}_{\mathsf{ObtIss}}$.

Game 0 → Game 1: If $\mathcal{A}$ queries a set $\mathtt{A}$ with $a \in \mathtt{A}$ to one of the two oracles, then this breaks the q-co-DL assumption for $q = t$ and $\mathsf{BG} = \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$. Denoting by

$\epsilon_{qDL}(\kappa)$ the advantage of solving the q-co-DL assumption, we have thus

$$|\Pr[S_0] - \Pr[S_1]| \leq \epsilon_{qDL}(\kappa) . \tag{13}$$

Game 2: As Game 1, with the difference that the oracle $\mathcal{O}_{\mathsf{Show}}$ is run as follows:

$\mathcal{O}_{\mathsf{Show}}(j, \mathtt{D})$: As in Game 0, but the ZKPoK $\Pi^{\mathcal{R}_\mathsf{F}}(C_1, C_2, C_3)$ is simulated.

Game 1 → Game 2: By the perfect zero-knowledge property of $\Pi^{\mathcal{R}_\mathsf{F}}$, we have that

$$\Pr[S_1] = \Pr[S_2] . \tag{14}$$

Game 3: As Game 2, except that oracle $\mathcal{O}_{\mathrm{HU}}$ is run as follows:

$\mathcal{O}_{\mathrm{HU}}(i)$: As in Game 0, but when executing $\mathsf{UserKeyGen}(\mathsf{opk})$, the experiment draws $\mathsf{usk} \xleftarrow{R} \mathbb{Z}_p$ instead of $\mathsf{usk} \xleftarrow{R} \mathbb{Z}_p^*$ and it aborts if $\mathsf{usk} = 0$.

Game 2 → Game 3: Denoting by q_u the number of queries to $\mathcal{O}_{\mathrm{HU}}$, we have

$$|\Pr[S_2] - \Pr[S_3]| \leq \tfrac{q_u}{p} . \tag{15}$$

Game 4: As Game 3, except that when $\mathcal{A}$ eventually delivers a valid showing by conducting $\Pi^{\mathcal{R}_\mathsf{F}}(C_1^*, C_2^*, C_3^*)$, the experiment runs the knowledge extractor of $\Pi^{\mathcal{R}_\mathsf{F}}$ and extracts a witness w. If the extractor fails, we abort.

Game 3 → Game 4: The success probability in Game 4 is the same as in Game 3, unless the extractor fails, i.e., using knowledge soundness we have

$$|\Pr[S_3] - \Pr[S_4]| \leq \epsilon_{ks}(\kappa) . \tag{16}$$

Game 5: As Game 4, except that we pick an index $k \xleftarrow{R} [q_o]$, where q_o is the number of queries to $\mathcal{O}_{\mathsf{ObtIss}}$. (The game guesses that the adversary will use the kth issued credential in its Type 3 forgery.)

The extracted witness w is such that $w = (r, \mu) \in (\mathbb{Z}_p^*)^2$ and $C_2^* = rC_1^*$ and $C_3^* = \mu P$. If credential $((C', R'), \sigma', r', O') \leftarrow \mathtt{CRED}[k]$ is such that $(C', R', P) \neq \mu^{-1} \cdot (C_1^*, C_2^*, C_3^*)$, then the experiment aborts. Furthermore, we change the executions of the following oracle, by aborting should the adversary want to corrupt the user that owns the kth credential:

$\mathcal{O}_{\mathrm{CU}}(i)$: As in Game 0, except that the experiment aborts when $i = \mathtt{OWNR}[k]$.

Game 4 → Game 5: Note that when the forgery is of Type 3, then there exists some j s.t. for $\mathtt{CRED}[j] = ((C', R'), \sigma', r', O')$ we have $(C', R', P) = \mu^{-1} \cdot (C_1^*, C_2^*, C_3^*)$; moreover, $\mathtt{OWNR}[j] \in \mathtt{HU}$. With probability $\frac{1}{q_o}$, we have $k = j$, in which case the experiment does not abort, i.e., we have

$$\Pr[S_5] \geq \tfrac{1}{q_o} \Pr[S_4] . \tag{17}$$

We will now show that $\Pr[S_5] \leq \epsilon_{DL}(\kappa)$, where $\epsilon_{DL}(\kappa)$ is the advantage of solving the DLP. $\mathcal{B}$ plays the role of the challenger for $\mathcal{A}$ in Game 5 and obtains a $\mathbb{G}_1$-DLP instance

(BG, xP) with $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P}) = \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$, generates $\mathsf{pp_{sc}} \leftarrow (\mathsf{BG}, (a^i P, a^i \hat{P})_{i\in[t]})$ by picking $a \xleftarrow{R} \mathbb{Z}_p$, generates $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{R} \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 1^3)$ and sets $(\mathsf{osk}, \mathsf{opk}) \leftarrow ((a, \mathsf{sk}), (\mathsf{pp_{sc}}, \mathsf{pk}))$. Then, $\mathcal{B}$ runs $\mathcal{A}(\mathsf{opk})$ and simulates the oracles as in Game 5, except for $\mathcal{O}_{\mathsf{ObtIss}}$, whose simulation is as follows:

$\mathcal{O}_{\mathsf{ObtIss}}(i, \mathtt{A})$: Let this be the jth query. $\mathcal{B}$ first computes $C \leftarrow \mathtt{USK}[i] \cdot f_{\mathtt{A}}(a) \cdot P$. If $j = k$, then it sets $R \leftarrow \mathtt{USK}[i] \cdot f_{\mathtt{A}}(a) \cdot xP$ $(= x \cdot C)$, $O = (0, \mathtt{USK}[i])$ and appends $\mathsf{cred} = ((C, R), \sigma, \perp, O)$ to $\mathtt{CRED}$. Otherwise, $\mathcal{B}$ proceeds as in Game 5.

Note that since Game 2, the third component (r) of the credential is not required to simulate $\mathcal{O}_{\mathsf{Show}}$ queries. When $\mathcal{A}$ outputs $(\mathtt{D}^*, \mathsf{st})$, then $\mathcal{B}$ runs $\mathcal{A}(\mathsf{st})$ and interacts with $\mathcal{A}$ as verifier in a showing protocol. If $\mathcal{A}$ wins Game 5 using (C_1^*, C_2^*, C_3^*) and conducting $\Pi^{\mathcal{R}_\mathsf{F}}(C_1^*, C_2^*, C_3^*)$, then $\mathcal{B}$ runs the knowledge extractor of $\Pi^{\mathcal{R}_\mathsf{F}}$ and extracts a witness $w = (r', \mu) \in (\mathbb{Z}_p^*)^2$ such that $C_2^* = r'C_1^*$ and $C_3^* = \mu P$. Further, we have that $((C', R'), \sigma', \perp, O') = \mathtt{CRED}[k]$. In the end, $\mathcal{B}$ outputs r' as a solution to the DLP in $\mathbb{G}_1$. We thus have

$$\Pr[S_5] \leq \epsilon_{DL}(\kappa)\ . \tag{18}$$

Equations (13)–(18) together yield $\Pr[S_0] \leq q_o \cdot \epsilon_{DL}(\kappa) + \epsilon_{ks}(\kappa) + \frac{q_u}{p} + \epsilon_{qDL}(\kappa)$, where $q = t$ and q_o and q_u are the number of queries to $\mathcal{O}_{\mathsf{ObtIss}}$ and $\mathcal{O}_{\mathtt{HU}}$, respectively. □

Theorem 8. *Let $\Pi^{\mathcal{R}_\mathsf{F}}$, $\Pi^{\mathcal{R}_\mathsf{U}}$ and $\Pi^{\mathcal{R}_\mathsf{O}}$ be ZKPoKs. If the* SPS-EQ *has a class-hiding message space and perfectly adapts signatures, then Scheme 2 is anonymous.*

The proof proceeds by defining a sequence of indistinguishable games in the last of which the answers of oracle $\mathcal{O}_{LoR}$ are independent of the bit b. Such an answer contains (C_1, C_2, C_3), σ' and the proof $\Pi^{\mathcal{R}_\mathsf{F}}(C_1, C_2, C_3)$. We first replace the signature σ' by a fresh signature (Game 2) and simulate the proof $\Pi^{\mathcal{R}_\mathsf{F}}$ (Game 3). In Games 5 and 6, we replace C_1 and C_2 by random elements. Since $C_3 = \mu \cdot P$ for $\mu \xleftarrow{R} \mathbb{Z}_p^*$, in the final game the adversary receives a fresh signature σ' on a random tuple (C_1, C_2, C_3) and a simulated proof, resulting in a game that is independent of b.

Proof (of Theorem 8). We assume that adversary $\mathcal{A}$ at some point calls $\mathcal{O}_{LoR}$ for some $(j_0, j_1, \mathtt{D})$ with both $\mathtt{OWNR}[j_0], \mathtt{OWNR}[j_1] \in \mathtt{HU}$. This is w.l.o.g., as otherwise the bit b is perfectly hidden from $\mathcal{A}$. Henceforth, we denote the event that an adversary wins Game i by S_i.

Game 0: The original game as given in Definition 29.

Game 1: As Game 0, except for the oracle $\mathcal{O}_{\mathsf{Obtain}}$. On the first successful completion of the ZKPoK $\Pi^{\mathcal{R}_\mathsf{O}}(\mathsf{opk})$ (of which there must be at least one by the above assumption), the experiment runs the knowledge extractor for $\Pi^{\mathcal{R}_\mathsf{O}}$, which extracts a witness (w_1, w_2). If the extractor fails, we abort.

Game 0 → Game 1: The success probability in Game 1 is the same as in Game 0, unless the extractor fails, i.e., using knowledge soundness we have

$$|\Pr[S_0] - \Pr[S_1]| \leq \epsilon_{ks}(\kappa)\ . \tag{19}$$

Game 2: As Game 1, except that the experiment sets $a \leftarrow w_1$ and $\mathsf{sk} \leftarrow w_2$ and runs $\mathcal{O}_{LoR}$ as follows:

$\mathcal{O}_{LoR}(j_0, j_1, \mathtt{D})$: As in Game 0, except that all executions of $\mathsf{ChgRep}_{\mathcal{R}}((C, r \cdot C, P), \sigma, \mu, \mathsf{pk})$ for credential $(C, \sigma, r, O) \leftarrow \mathtt{CRED}[j_b]$ and $\mu \xleftarrow{R} \mathbb{Z}_p^*$ are replaced by $\mathsf{Sign}_{\mathcal{R}}(\mu \cdot (C, r \cdot C, P), \mathsf{sk})$).

Game 1 → Game 2: By knowledge soundness of $\Pi^{\mathcal{R}_{\mathsf{O}}}$, we have $\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk}) = 1$, and by perfect adaptation of signatures of SPS-EQ (Definition 19), $\mathsf{ChgRep}_{\mathcal{R}}(\vec{M}, \sigma, \mu, \mathsf{pk})$ and $\mathsf{Sign}_{\mathcal{R}}(\mu\vec{M}, \mathsf{sk})$ are identically distributed for all $\vec{M} \in (\mathbb{G}_1^*)^3$. We thus have $\Pr[S_1] = \Pr[S_2]$.

Game 3: As Game 2, except that the experiment runs $\mathcal{O}_{LoR}$ as follows:

$\mathcal{O}_{LoR}(j_0, j_1, \mathtt{D})$: As in Game 2, but the ZKPoK $\Pi^{\mathcal{R}_{\mathsf{F}}}(C_1^*, C_2^*, C_3^*)$ is simulated.

Game 2 → Game 3: By perfect zero-knowledge of $\Pi^{\mathcal{R}_{\mathsf{F}}}$, we have that $\Pr[S_2] = \Pr[S_3]$ and thus

$$\Pr[S_1] = \Pr[S_2] = \Pr[S_3] \,. \tag{20}$$

Game 4: As Game 3, except for the following changes. Let q_u be (an upper bound on) the number of queries made to $\mathcal{O}_{\mathtt{HU}}$. At the beginning, Game 4 picks $k \xleftarrow{R} [q_u]$ (it guesses that the user that owns the j_bth credential is registered at the kth call to $\mathcal{O}_{\mathtt{HU}}$) and runs $\mathcal{O}_{\mathtt{HU}}$, $\mathcal{O}_{\mathtt{CU}}$ and $\mathcal{O}_{LoR}$ as follows:

$\mathcal{O}_{\mathtt{HU}}(i)$: As in Game 3, except if this is the kth call to $\mathcal{O}_{\mathtt{HU}}$, then it additionally defines $i^* \leftarrow i$.

$\mathcal{O}_{\mathtt{CU}}(i, \mathsf{upk})$: If $i \in \mathtt{CU}$ or $i \in I_{LoR}$, it returns $\perp$ (as in the previous games). If $i = i^*$, then the experiment stops and outputs a random bit $b' \xleftarrow{R} \{0, 1\}$. Otherwise, if $i \in \mathtt{HU}$, it returns user i's usk and credentials and moves i from $\mathtt{HU}$ to $\mathtt{CU}$; and if $i \notin \mathtt{HU} \cup \mathtt{CU}$, it adds i to $\mathtt{CU}$ and sets $\mathtt{UPK}[i] \leftarrow \mathsf{upk}$.

$\mathcal{O}_{LoR}(j_0, j_1, \mathtt{D})$: As in Game 3, except that if $i^* \neq \mathtt{OWNR}[j_b]$, the experiment stops outputting $b' \xleftarrow{R} \{0, 1\}$.

Game 3 → Game 4: By assumption, $\mathcal{O}_{LoR}$ is called at least once with some input $(j_0, j_1, \mathtt{D})$ with $\mathtt{OWNR}[j_0], \mathtt{OWNR}[j_1] \in \mathtt{HU}$. If $i^* = \mathtt{OWNR}[j_b]$, then $\mathcal{O}_{LoR}$ does not abort and neither does $\mathcal{O}_{\mathtt{CU}}$ (it cannot have been called on $\mathtt{OWNR}[j_b]$ before that call to $\mathcal{O}_{LoR}$ (otherwise $\mathtt{OWNR}[j_b] \notin \mathtt{HU}$); if called afterward, it returns $\perp$, since $i^* \in I_{LoR}$). Since $i^* = \mathtt{OWNR}[j_b]$ with probability $\frac{1}{q_u}$, the probability that the experiment does not abort is at least $\frac{1}{q_u}$, and thus

$$\Pr[S_4] \geq \left(1 - \tfrac{1}{q_u}\right) \frac{1}{2} + \frac{1}{q_u} \cdot \Pr[S_3] \,. \tag{21}$$

Game 5: As Game 4, except for $\mathcal{O}_{LoR}$:

$\mathcal{O}_{LoR}(j_0, j_1, \mathtt{D})$: As in Game 4, except that in addition to $\mu \xleftarrow{R} \mathbb{Z}_p^*$, it picks $C_1 \xleftarrow{R} \mathbb{G}_1^*$ and performs the showing using $\mathsf{cred}' \xleftarrow{R} ((C_1, r \cdot C_1, \mu \cdot P), \mathsf{Sign}_{\mathcal{R}}((C_1, r \cdot C_1, \mu \cdot P), \mathsf{sk}))$, with $r \leftarrow \mathtt{CRED}[j_b][3]$, and $W \leftarrow \perp$ (if $a \in \mathtt{D}$) or $W \leftarrow f_{\mathtt{D}}(a)^{-1} \cdot C_1$ (if $a \notin \mathtt{D}$), where $a = w_1$ is the value extracted since Game 1.

Note that the only difference is the choice of C_1; W is distributed as in Game 4, in particular, if $a \notin \mathtt{D}$, it is the unique element satisfying $\mathsf{VerifySubset}(\mathsf{pp}, C, \mathtt{D}, W)$.

Game 4 → Game 5: Let $(\mathsf{BG}, xP, yP, zP)$ be a DDH instance for $\mathsf{BG} = \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$. After initializing the environment, the simulation initializes a list $L \leftarrow \emptyset$. The oracles are simulated as in Game 4, except for the subsequent oracles, which are simulated as follows:

$\mathcal{O}_{\mathtt{HU}}(i)$: As in Game 4, but if this is the kth call then, besides setting $i^* \leftarrow i$, it sets $\mathtt{USK}[i] \leftarrow \perp$ and $\mathtt{UPK}[i] \leftarrow xP$ (which implicitly sets $\mathsf{usk} \leftarrow x$).

$\mathcal{O}_{\mathsf{Obtain}}(i, \mathtt{A})$: As in Game 4, except for the computation of the following values if $i = i^*$. Let this be the jth call to this oracle. If $a \notin \mathtt{A}$, it computes C as $C \leftarrow f_{\mathtt{A}}(a) \cdot xP$ and sets $L[j] \leftarrow \perp$. If $a \in \mathtt{A}$, it picks $\rho \xleftarrow{R} \mathbb{Z}_p^*$, computes C as $C \leftarrow \rho \cdot xP$, sets $L[j] \leftarrow \rho$ and simulates the ZKPoK $\Pi^{\mathcal{R}_\mathsf{U}}(\mathsf{upk})$ (by the perfect ZK property of $\Pi^{\mathcal{R}_\mathsf{U}}(\mathsf{upk})$ the simulation is perfect). (In both cases, C is thus distributed as in the original game.)

$\mathcal{O}_{\mathsf{Show}}(j, \mathtt{D})$: As in Game 4, with the difference that if $\mathtt{OWNR}[j] = i^*$ and $a \notin \mathtt{D}$ it computes the witness $W \leftarrow \mu f_{\mathtt{A}\setminus\mathtt{D}}(a) \cdot xP$. ($W$ is thus distributed as in the original game.)

$\mathcal{O}_{LoR}(j_0, j_1, \mathtt{D})$: As in Game 4, with the following difference. Using self-reducibility of DDH, it picks $s, t \xleftarrow{R} \mathbb{Z}_p$ and computes $Y' \leftarrow t \cdot yP + sP = y'P$ with $y' \leftarrow ty + s$, and $Z' \leftarrow t \cdot zP + s \cdot xP = (t(z - xy) + xy')P$. (If $z \neq xy$, then Y' and Z' are independently random; otherwise, $Z' = y'X$.) It performs the showing using the following values (implicitly setting $\mu \leftarrow y'$):

- If $a \notin \mathtt{ATTR}[j_b]$: $C_1 \leftarrow f_{\mathtt{A}}(a) \cdot Z'$ and $W \leftarrow f_{\mathtt{D}}(a)^{-1} \cdot C_1$;
- If $a \in \mathtt{ATTR}[j_b]$ and $a \notin \mathtt{D}$: $C_1 \leftarrow \rho \cdot Z'$ with $\rho \leftarrow L[j_b]$ and $W \leftarrow f_{\mathtt{D}}(a)^{-1} \cdot C_1$;
- If $a \in \mathtt{D}$: $C_1 \leftarrow \rho \cdot Z'$ with $\rho \leftarrow L[j_b]$ and $W \leftarrow \perp$;

$C_2 \leftarrow r \cdot C_1$, $C_3 \leftarrow Y'$ and $r \leftarrow \mathtt{CRED}[j_b][3]$.

Apart from an error event happening with negligible probability, we have simulated Game 4 if the DDH instance was "real" and Game 5 otherwise. If $xP = 0_{\mathbb{G}_1}$, or if during the simulation of $\mathcal{O}_{LoR}$ it occurs that $Y' = 0_{\mathbb{G}_1}$ or $Z' = 0_{\mathbb{G}_1}$, then the distribution of values is not as in one of the two games. Otherwise, we have implicitly set $\mathsf{usk} \leftarrow x$ and $\mu \leftarrow y'$ (for a fresh value y' at every call of $\mathcal{O}_{LoR}$). In case of a DDH instance, we have (depending on the case) $C_1 \leftarrow \mathsf{usk}\mu f_{\mathtt{A}}(a) \cdot P$ (or $C_1 = \rho \cdot x\mu \cdot P = \mu \cdot C$); otherwise, C_1 is independently random. Letting $\epsilon_{DDH}(\kappa)$ denote the advantage of solving the DDH problem and q_l the number of queries to the $\mathcal{O}_{LoR}$, we have

$$|\Pr[S_4] - \Pr[S_5]| \leq \epsilon_{DDH}(\kappa) + (1 + 2q_l)\frac{1}{p} \,. \tag{22}$$

Game 6: As Game 5, except for $\mathcal{O}_{LoR}$:

$\mathcal{O}_{LoR}(j_0, j_1, \mathtt{D})$: As in Game 5, except that, in addition to μ and C_1, it also picks $C_2 \xleftarrow{R} \mathbb{G}_1^*$ and performs the showing using $\mathsf{cred}' \xleftarrow{R} ((C_1, C_2, \mu \cdot P), \mathsf{Sign}_{\mathcal{R}}((C_1, C_2, \mu \cdot P), \mathsf{sk}))$ and W as in Game 5.

Game 5 → Game 6: Let $(\mathsf{BG}, xP, yP, zP)$ be a DDH instance for $\mathsf{BG} = \mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$. After initializing the environment, the simulation initializes a list $L \leftarrow \emptyset$. The oracles are simulated as in Game 5, except for the subsequent oracles, which are simulated as follows:

$\mathcal{O}_{\mathsf{Obtain}}(i, \mathbb{A})$: As in Game 5, except for the computation of the following values if $i = i^*$. Let this be the jth call to this oracle. It first picks $u \xleftarrow{R} \mathbb{Z}_p$ and sets $X' \leftarrow xP + u \cdot P$ and $L[j] \leftarrow u$.

If $a \notin \mathbb{A}$, it computes $C \leftarrow f_{\mathbb{A}}(a) \cdot \mathtt{USK}[i] \cdot P$ and $R \leftarrow f_{\mathbb{A}}(a) \cdot \mathtt{USK}[i] \cdot X'$. If $a \in \mathbb{A}$, it picks $\rho \xleftarrow{R} \mathbb{Z}_p^*$ and computes $C \leftarrow \rho \cdot P$ and $R \leftarrow \rho \cdot X'$. In both cases, it sets $r \leftarrow \perp$ (r is implicitly set to $r \leftarrow x' := x + u$ and C and $R = r \cdot C$ are distributed as in the original game; unless $X' = 0_{\mathbb{G}_1}$). Note that, since the ZKPoK in $\mathcal{O}_{\mathsf{Show}}$ is simulated, r is not used anywhere in the game.

$\mathcal{O}_{LoR}(j_0, j_1, \mathtt{D})$: As in Game 5, with the difference that it fetches $u \leftarrow L[j_b]$, picks $s, t \xleftarrow{R} \mathbb{Z}_p$ and computes $Y' \leftarrow t \cdot yP + s \cdot P = y'P$ with $y' \leftarrow ty + s$, and $Z' \leftarrow t \cdot zP + s \cdot xP + ut \cdot yP + us \cdot P = (t(z - xy) + x'y')P$. It picks $\mu \xleftarrow{R} \mathbb{Z}_p^*$ and performs the showing using $C_1 \leftarrow Y'$, $C_2 \leftarrow Z'$ and $C_3 \leftarrow \mu \cdot P$. Witness W is computed from C_1 as in the previous simulation.

Apart from an error event happening with negligible probability, we have simulated Game 5 if the DDH instance was valid and Game 6 otherwise. If $X' = 0_{\mathbb{G}_1}$ during the simulation of $\mathcal{O}_{\mathsf{Obtain}}$, or if during the simulation of $\mathcal{O}_{LoR}$ it occurs that $Y' = 0_{\mathbb{G}_1}$ or $Z' = 0_{\mathbb{G}_1}$, then the distribution of values is not as in one of the two games. Otherwise, we have implicitly set $r \leftarrow x'$ (for a fresh value x' at every call of $\mathcal{O}_{\mathsf{Obtain}}$) and $C_1 \leftarrow Y'$ (for a fresh value Y' at every call of $\mathcal{O}_{LoR}$). In case of a DDH instance, we have $C_2 = r \cdot C_1$ (as in Game 5); otherwise, C_2 is independently random (as in Game 6). Letting $\epsilon_{DDH}(\kappa)$ denote the advantage of solving the DDH problem, and q_o and q_l be the number of queries to $\mathcal{O}_{\mathsf{Obtain}}$ and $\mathcal{O}_{LoR}$, respectively, we get

$$|\Pr[S_5] - \Pr[S_6]| \leq \epsilon_{DDH}(\kappa) + (q_o + 2q_l)\frac{1}{p} \ . \tag{23}$$

In Game 6, the $\mathcal{O}_{LoR}$ oracle returns a fresh signature σ on a random triple $(C_1, C_2, C_3) \xleftarrow{R} (\mathbb{G}_1^*)^3$ and a simulated proof; the bit b is thus information-theoretically hidden from $\mathcal{A}$ and we have $\Pr[S_6] = \frac{1}{2}$. From this and Eqs. (23)–(19), we have

$$\begin{aligned}
&\Pr[S_5] \leq \Pr[S_6] + \epsilon_{DDH}(\kappa) + (q_o + 2q_l)\tfrac{1}{p} = \tfrac{1}{2} + \epsilon_{DDH}(\kappa) + (q_o + 2q_l)\tfrac{1}{p} \ , \\
&\Pr[S_4] \leq \Pr[S_5] + \epsilon_{DDH}(\kappa) + (1 + 2q_l)\tfrac{1}{p} \leq \tfrac{1}{2} + 2 \cdot \epsilon_{DDH}(\kappa) + (1 + q_o + 4q_l)\tfrac{1}{p} \ , \\
&\Pr[S_3] \leq \tfrac{1}{2} + q_u \cdot \Pr[S_4] - \tfrac{1}{2} \cdot q_u \leq \tfrac{1}{2} + q_u \cdot \left(2 \cdot \epsilon_{DDH}(\kappa) + (1 + q_o + 4q_l)\tfrac{1}{p}\right) , \\
&\Pr[S_0] \leq \Pr[S_1] + \epsilon_{ks}(\kappa) \leq \tfrac{1}{2} + \epsilon_{ks}(\kappa) + q_u \cdot \left(2 \cdot \epsilon_{DDH}(\kappa) + (1 + q_o + 4q_l)\tfrac{1}{p}\right)
\end{aligned}$$

where $\Pr[S_1] = \Pr[S_3]$; q_u, q_o and q_l are the number of queries to the $\mathcal{O}_{\mathtt{HU}}$, $\mathcal{O}_{\mathsf{Obtain}}$ and the $\mathcal{O}_{LoR}$ oracle, respectively. Assuming security of the ZKPoKs and DDH, the adversary's advantage is thus negligible. □

Remark 1. (A Concurrently Secure Scheme Varian) We now sketch the idea of a more efficient and concurrently secure variant of our scheme, which uses a CRS (and is in particular, anonymous under malicious organization keys in the CRS model). Damgård [47] proposes a generic transformation of any Σ-protocol for an arbitrary NP-relation $\mathcal{R}$ into a 3-move concurrent ZKPoK (without any timing constraints), under the assumption of one-way functions and using a CRS. This requires the introduction of a setup algorithm and replacing the ZKPoKs used in our construction with those from [47] (the statements proven stay the same). It uses four moves during issuing and only three moves during showing (when interleaving the ZKPoK moves with the other protocol moves).

The introduction of system parameters pp allows us to move the set-commitment parameters from the organization keys to pp, which reduces the size of organization public keys.

5.6. *Efficiency Analysis and Comparison*

We provide a brief comparison with other ABC approaches. As other candidates for multi-show ABCs, we consider the Camenisch-Lysyanskaya schemes [40–42] as well as schemes from BBS$^+$ signatures [13,16], which cover a broad class of ABC schemes from randomizable signature schemes with efficient proofs of knowledge. Furthermore, we look at two alternative multi-show ABC constructions [43,44], the recent self-blindable scheme in [82], as well as Brands' approach [28] (for which there is a tweaked provable secure version [24]) for the sake of completeness, although the latter only provides one-show ABCs. We omit a comparison with approaches that only support a single attribute per credential, e.g., [12], as our focus is on schemes supporting an arbitrary number of attributes. We also omit approaches that achieve more efficient showings for existing ABC systems only in very special cases such as for attribute values that come from a very small set (and are, thus, hard to compare).[6] Finally, we also include the recent approach in [31] that has the same asymptotic parameter sizes as our approach. They achieve strong security in the UC framework [30], but far less efficient constructions when it comes to concrete instantiations. Their approach is equally expressive as ours (selective disclosure), but additionally supports pseudonyms and context-specific pseudonyms for showings. For our comparison in Table 1, we take their most efficient instantiation (which does not provide secret key extractability) and note that our showings require less than 10 group elements (when instantiated with Scheme 1 and the ZKPoK protocol from [32]), whereas the cheapest variant in [31] requires around 100 group elements.

Table 1 gives an overview of these systems, where BG denotes a bilinear-group setting; $\mathbb{G}_q$ denotes a group of prime order q (e.g., a subgroup of large order q of $\mathbb{Z}_p^*$ or an elliptic curve group of order q) and $\mathbb{Z}_N$ an RSA group. By $|\mathbb{G}|$, denote the bitlength of the representation of an element from group $\mathbb{G}$, by MK we indicate whether anonymity (privacy) holds with respect to maliciously generated issuer keys and by P we indicate whether the schemes support selective disclosure (s) or also proving relations about attributes (r). We note that $\circ$ indicates that the most efficient construction from [31] used in Table 1 does not consider malicious keys, while the other less efficient ones in [31] do. The required assumptions for the schemes include the strong RSA (sRSA) [27], LRSW [77], SXDH (cf. Definition 6), XDLIN (a decision linear [16] variant of SXDH),

[6] For instance, the approach in [35] for CL credentials in the RSA setting (encoding attributes as prime numbers) or in a pairing-based setting using BBS$^+$ credentials [85] (encoding attributes using accumulators) where the latter additionally requires very large public parameters (one F-secure BB signature [19] for every possible attribute value).

Table 1. Comparison of various approaches to ABC systems.

Scheme	Parameter size (L attr.)				
	Setting	Assumption	|opk|		|cred|
[41]	$\mathbb{Z}_N$	sRSA	$O(L)$	$O(1)$	$3\|\mathbb{Z}_N\|$
[42]	BG	LRSW	$O(L)$	$O(L)$	$(2L+2)\|\mathbb{G}_1\|$
[13,16]	BG	q-SDH	$O(L)$	$O(1)$	$\|\mathbb{G}_1\|+2\|\mathbb{Z}_q\|$
[43]	BG	q-ADHSDH	$O(1)$	$O(L)$	$L\|\mathbb{G}_1\|+\|\mathbb{G}_2\|$
[44]	BG	q-SDH,XDH	$O(L)$	$O(L)$	$(2L+2)(\|\mathbb{G}_1\|+\|\mathbb{Z}_p\|)$
[28]	$\mathbb{G}_q$	?	$O(L)$	$O(1)$	$2(\|\mathbb{G}_q\|+\|\mathbb{Z}_q\|)$
[82]	BG	LRSW	$O(L)$	$O(L)$	$(2L+4)(\|\mathbb{G}_1\|+\|\mathbb{Z}_q\|)$
[31]	BG	SXDH, J-RootDH, n-BSDH q-SDH, XDLIN, co-CDH, DBP	$O(L)$	$O(1)$	$6\|\mathbb{G}_1\|+2\|\mathbb{G}_2\|+\|\mathbb{Z}_p\|$
Scheme 2	BG	GGM	$O(L)$	$O(1)$	$3\|\mathbb{G}_1\|+\|\mathbb{G}_2\|+2\|\mathbb{Z}_p\|$

Scheme	Issuing			Showing (k-of-L attr.)			MK	P
	Issuer	User	Comm	Verifier	User	Comm		
[41]	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L-k)$	×	r
[42]	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	×	r
[13,16]	$O(L)$	$O(L)$	$O(1)$	$O(L)$	$O(L)$	$O(L)$	×	r
[43]	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(1)$	$O(1)$	×	s
[44]	$O(L)$	$O(L)$	$O(L)$	$O(k)$	$O(k)$	$O(k)$	×	s
[28]	$O(L)$	$O(L)$	$O(1)$	$O(k)$	$O(k)$	$O(L-k)$	×	r
[82]	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	×	r
[31]	$O(L)$	$O(L)$	$O(1)$	$O(k)$	$O(L-k)$	$O(1)$	✓	s
Scheme 2	$O(L)$	$O(L)$	$O(1)$	$O(k)$	$O(L-k)$	$O(1)$	✓	s

DBP [3], q-SDH [14], q-ADHSDH [58], n-BSDH [64], J-RootDH [31], the generic-group model (GGM), and we write ? when no security proof is given.

We emphasize that, in contrast to other approaches, such as [42,44,82], our construction only requires a small and constant number of pairing evaluations in all protocol steps. Finally, we want to mention that the model introduced in [37] allows to instantiate constructions, for instance based on [41], that can deal with malicious organization keys (although at the cost of efficiency).

6. Future Work

Some challenging issues with respect to SPS-EQ remain open. Primarily, the construction of an instantiation secure in the standard model (or CRS model) that relies on simple assumptions and perfectly adapts signatures (under malicious keys) is an open problem. A first step was [55], which gives a standard-model construction of SPS-EQ under a q-type assumption, but which only provides a weaker form of privacy, which is too weak for any of the considered applications of SPS-EQ. A further step was [52], which gives a construction of SPS-EQ from standard assumptions, but achieving a weaker form of unforgeability where the adversary must reveal the logarithms of the message vector for which it queries a signature. This notion is not sufficient for the construction of round-optimal blind signatures from SPS-EQ [54,55].[7]

Another interesting question is whether such signatures when built for other more general equivalence relations yield alternative and further applications.

Acknowledgements

Work started while the first author was at IST Austria and supported by the European Research Council, ERC Starting Grant (259668-PSPC); now supported by the French ANR EfTrEC project (ANR-16-CE39-0002). Work has been done while the second and third authors were at IAIK, Graz University of Technology. The second author has been supported by the European Commission through projects FP7-MATTHEW (GA No. 610436) and FP7-FutureID (GA No. 318424). The work of the last author has been supported by the European Commission through project FP7-FutureID (GA No. 318424) and by EU HORIZON 2020 through project PRISMACLOUD (GA NO. 644962).

References

[1] J.H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, A. Shelat, B. Waters, Computing on authenticated data, in Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*. (Springer, Heidelberg, March 2012), pp. 1–20

[2] M. Abe, M. Chase, B. David, M. K., R. Nishimaki, M. Ohkubo, Constant-size structure-preserving signatures: Generic constructions and simple assumptions, in Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*. (Springer, Heidelberg, 2012), pp. 4–24

[3] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo, Structure-preserving signatures and commitments to group elements, in Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*. (Springer, Heidelberg, August 2010), pp. 209–236

[4] M. Abe, J. Groth, K. Haralambiev, M. Ohkubo, Optimal structure-preserving signatures in asymmetric bilinear groups, in Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*. (Springer, Heidelberg, August 2011), pp. 649–666

[5] M. Abe, J. Groth, M. Ohkubo, M. Tibouchi, Structure-preserving signatures from type II pairings, in Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*. (Springer, Heidelberg, August 2014), pp. 390–407

[6] M. Abe, J. Groth, M. Ohkubo, M. Tibouchi, Unified, minimal and selectively randomizable structure-preserving signatures, in Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*. (Springer, Heidelberg, February 2014), pp. 688–712

[7] M. Abe, D. Hofheinz, R. Nishimaki, M. Ohkubo, J. Pan, Compact structure-preserving signatures with almost tight security, in Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*. (Springer, Heidelberg, August 2017), pp. 548–580

[7] In ABC schemes, one can add interactive proofs of knowledge of the logarithms when obtaining a signature; the reduction can then make signing queries using the logarithms instead of the group elements itself, as required by the security model in [52]. However, round-optimality of blind signatures precludes adding interaction; it is also not possible to add NIZKs of knowledge, as they require a CRS, which is not compatible with the strong security model (malicious-signer anonymity) for blind signatures considered in [55].

[8] M. Abe, K. Haralambiev, M. Ohkubo, Signing on elements in bilinear groups for modular protocol design. Cryptology ePrint Archive, Report 2010/133, (2010). http://eprint.iacr.org/2010/133

[9] M. Abe, M. Kohlweiss, M. Ohkubo, M. Tibouchi, Fully structure-preserving signatures and shrinking commitments, in Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*. (Springer, Heidelberg, April 2015), pp. 35–65

[10] N. Attrapadung, B. Libert, T. Peters, Computing on authenticated data: New privacy definitions and constructions, in Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*. (Springer, Heidelberg, December 2012), pp. 367–385

[11] N. Attrapadung, B. Libert, T. Peters, Efficient completely context-hiding quotable and linearly homomorphic signatures, in K. Kurosawa and G. Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*. (Springer, Heidelberg, February/March 2013), pp. 386–404

[12] N. Akagi, Y. Manabe, T. Okamoto, An efficient anonymous credential system, in G. Tsudik, editor, *FC 2008*, volume 5143 of *LNCS*. (Springer, Heidelberg, January 2008), pp. 272–286

[13] M.H. Au, W. Susilo, Y. Mu, Constant-size dynamic k-TAA, in R. De Prisco and M. Yung, editors, *SCN 06*, volume 4116 of *LNCS*. (Springer, Heidelberg, September 2006), pp. 111–125

[14] D. Boneh, X. Boyen, Short signatures without random oracles, in C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*. (Springer, Heidelberg, May 2004), pp. 56–73

[15] D. Boneh, X. Boyen, E.-J. Goh, Hierarchical identity based encryption with constant size ciphertext, in R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*. (Springer, Heidelberg, May 2005), pp. 440–456

[16] D. Boneh, X. Boyen, H. Shacham, Short group signatures, in M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*. (Springer, Heidelberg, August 2004), pp. 41–55

[17] D. Boneh, H. Corrigan-Gibbs, Bivariate polynomials modulo composites and their applications, in P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*. (Springer, Heidelberg, December 2014), pp. 42–62

[18] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, H. Shacham, Randomizable proofs and delegatable anonymous credentials, in S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*. (Springer, Heidelberg, August 2009), pp. 108–125

[19] M. Belenkiy, M. Chase, M. Kohlweiss, A. Lysyanskaya, P-signatures and noninteractive anonymous credentials, in R. Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*. (Springer, Heidelberg, March 2008), pp. 356–374

[20] G. Barthe, E. Fagerholm, D. Fiore, A. Scedrov, B. Schmidt, M. Tibouchi, Strongly-optimal structure preserving signatures from type II pairings: Synthesis and lower bounds, in J. Katz, editor, *PKC 2015*, volume 9020 of *LNCS*. (Springer, Heidelberg, March/April 2015), pp. 355–376

[21] D. Boneh, D. Freeman, J. Katz, B. Waters, Signing a linear subspace: Signature schemes for network coding, in S. Jarecki and G. Tsudik, editors, *PKC 2009*, volume 5443 of *LNCS*. (Springer, Heidelberg, March 2009), pp. 68–87

[22] O. Blazy, G. Fuchsbauer, D. Pointcheval, D. Vergnaud, Signatures on randomizable ciphertexts, in D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*. (Springer, Heidelberg, March 2011), pp. 403–422

[23] M. Bellare, G. Fuchsbauer, A. Scafuro, NIZKs with an untrusted CRS: Security in the face of parameter subversion, in J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*. (Springer, Heidelberg, December 2016), pp. 777–804

[24] F. Baldimtsi, A. Lysyanskaya, Anonymous credentials light, in A.-R. Sadeghi, V.D. Gligor, and M. Yung, editors, *ACM CCS 13*. (ACM Press, November 2013), pp. 1087–1098

[25] P.S.L.M. Barreto, M. Naehrig, Pairing-friendly elliptic curves of prime order, in B. Preneel and S. Tavares, editors, *SAC 2005*, volume 3897 of *LNCS*. (Springer, Heidelberg, August 2006), pp. 319–331

[26] X. Boyen, The uber-assumption family (invited talk), in S.D. Galbraith and K.G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*. (Springer, Heidelberg, 2008), pp. 39–56

[27] N. Bari, B. Pfitzmann, Collision-free accumulators and fail-stop signature schemes without trees, in W. Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*. (Springer, Heidelberg, May 1997), pp. 480–494

[28] S. Brands, *Rethinking public-key Infrastructures and Digital Certificates: Building in Privacy*. (MIT Press, 2000)

[29] M. Bellare, H. Shi, C. Zhang, Foundations of group signatures: The case of dynamic groups, in A. Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*. (Springer, Heidelberg, February 2005), pp. 136–153

[30] R. Canetti, Universally composable security: A new paradigm for cryptographic protocols, in *42nd FOCS*. IEEE Computer Society Press, (October 2001), pp. 136–145

[31] J. Camenisch, M. Dubovitskaya, K. Haralambiev, M. Kohlweiss, Composable and modular anonymous credentials: definitions and practical constructions, in T. Iwata and J.H. Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*. (Springer, Heidelberg, November/December 2015), pp. 262–288

[32] R. Cramer, I. Damgård, P.D. MacKenzie, Efficient zero-knowledge proofs of knowledge without intractability assumptions, in H. Imai and Y. Zheng, editors, *PKC 2000*, volume 1751 of *LNCS*. (Springer, Heidelberg, January 2000), pp. 354–372

[33] D. Catalano, D. Fiore, Vector commitments and their applications. In K. Kurosawa and G. Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*. (Springer, Heidelberg, February / March 2013), pp. 55–72

[34] D. Catalano, D. Fiore, B. Warinschi, Efficient network coding signatures in the standard model, in M. Fischlin, J. Buchmann, and M. Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*. (Springer, Heidelberg, 2012), pp. 680–696

[35] J. Camenisch, T. Groß, Efficient attributes for anonymous credentials. *ACM Transactions on Information and System Security*, **15**(1), 4, (2012)

[36] M. Chase, C. Ganesh, P. Mohassel, Efficient zero-knowledge proof of algebraic and non-algebraic statements with applications to privacy preserving credentials, in M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*. (Springer, Heidelberg, 2016), pp. 499–530

[37] J. Camenisch, S. Krenn, A. Lehmann, G.L. Mikkelsen, G. Neven, M.Ø. Pedersen, Formal treatment of privacy-enhancing credential systems, in O. Dunkelman and L. Keliher, editors, *SAC 2015*, volume 9566 of *LNCS*. (Springer, Heidelberg, August 2016), pp. 3–24

[38] M. Chase, M. Kohlweiss, A. Lysyanskaya, S. Meiklejohn. Malleable proof systems and applications, in D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*. (Springer, Heidelberg, April 2012), pp. 281–300

[39] M. Chase, M. Kohlweiss, A. Lysyanskaya, S. Meiklejohn. Malleable signatures: New definitions and delegatable anonymous credentials, in *IEEE 27th Computer Security Foundations Symposium, CSF 2014*, (2014), pp. 199–213

[40] J. Camenisch, A. Lysyanskaya, An efficient system for non-transferable anonymous credentials with optional anonymity revocation, in B. Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*. (Springer, Heidelberg, May 2001), pp. 93–118

[41] J. Camenisch, A. Lysyanskaya, A signature scheme with efficient protocols, in S. Cimato, C. Galdi, and G. Persiano, editors, *SCN 02*, volume 2576 of *LNCS*. (Springer, Heidelberg, September 2003), pp. 268–289

[42] J. Camenisch, A. Lysyanskaya, Signature schemes and anonymous credentials from bilinear maps, in M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*. (Springer, Heidelberg, August 2004), pp. 56–72

[43] S. Canard, R. Lescuyer, Anonymous credentials from (indexed) aggregate signatures, in *DIM'11, Proceedings of the 2013 ACM Workshop on Digital Identity Management, Chicago, IL, USA - October 21, 2011*, (2011), pp. 53–62

[44] S. Canard, R. Lescuyer, Protecting privacy by sanitizing personal data: a new approach to anonymous credentials, in K. Chen, Q. Xie, W. Qiu, N. Li, and W.-G. Tzeng, editors, *ASIACCS 13*. (ACM Press, May 2013), pp. 381–392

[45] S. Chatterjee, A. Menezes, On cryptographic protocols employing asymmetric pairings - the role of Ψ revisited. *Discrete Applied Mathematics* **159**(13), 1311–1322, (2011)

[46] D. Chaum, T.P. Pedersen, Wallet databases with observers, in E.F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*. (Springer, Heidelberg, 1993), pp. 89–105

[47] I. Damgård, Efficient concurrent zero-knowledge in the auxiliary string model, in B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*. (Springer, Heidelberg, May 2000), pp. 418–430

[48] I. Damgård, H. Haagh, C. Orlandi, Access control encryption: Enforcing information flow with cryptography, in M. Hirt and A.D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*. (Springer, Heidelberg, October/November 2016), pp. 547–576

[49] D. Derler, C. Hanser, D. Slamanig, A new approach to efficient revocable attribute-based anonymous credentials, in J. Groth, editor, *15th IMA International Conference on Cryptography and Coding*, volume 9496 of *LNCS*. (Springer, Heidelberg, 2015), pp. 57–74

[50] D. Derler, C. Hanser, D. Slamanig, Revisiting cryptographic accumulators, additional properties and relations to other primitives, in K. Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*. (Springer, Heidelberg, April 2015), pp. 127–144

[51] D. Derler, D. Slamanig, Fully-anonymous short dynamic group signatures without encryption. *IACR Cryptology ePrint Archive*, 2016:154, (2016)

[52] G. Fuchsbauer, R. Gay, Weakly secure equivalence-class signatures from standard assumptions, in M. Abdalla, editor, *PKC 2018*, LNCS. (Springer, 2018)

[53] G. Fuchsbauer, R. Gay, L. Kowalczyk, C. Orlandi, Access control encryption for equality, comparison, and more, in S. Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*. (Springer, Heidelberg, 2017), pp. 88–118

[54] G. Fuchsbauer, C. Hanser, C. Kamath, D. Slamanig, Practical round-optimal blind signatures in the standard model from weaker assumptions, in V. Zikas and R. De Prisco, editors, *SCN 16*, volume 9841 of *LNCS*. (Springer, Heidelberg, August/September 2016), pp. 391–408

[55] G. Fuchsbauer, C. Hanser, D. Slamanig, Practical round-optimal blind signatures in the standard model, in R. Gennaro and M.J.B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pp. 233–253. (Springer, Heidelberg, August 2015)

[56] E. Fujisaki, T. Okamoto, A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In K. Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*. (Springer, Heidelberg, May/June 1998), pp. 32–46

[57] D.M. Freeman, Improved security for linearly homomorphic signatures: A generic framework, in M. Fischlin, J. Buchmann, and M. Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*. (Springer, Heidelberg, May 2012), pp. 697–714

[58] G. Fuchsbauer, Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. Cryptology ePrint Archive, Report 2009/320 (2009). http://eprint.iacr.org/2009/320.

[59] G. Fuchsbauer, Commuting signatures and verifiable encryption, in K.G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*. (Springer, Heidelberg, May 2011), pp. 224–245

[60] G. Fuchsbauer, Breaking existential unforgeability of a signature scheme from asiacrypt 2014. Cryptology ePrint Archive, Report 2014/892, (2014). http://eprint.iacr.org/2014/892

[61] E. Ghadafi, Short structure-preserving signatures, in K. Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*. (Springer, Heidelberg, February / March 2016), pp. 305–321

[62] S. Goldwasser, S. Micali, R.L. Rivest, A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing* **17**(2), 281–308, (1988)

[63] O. Goldreich, *The Foundations of Cryptography - Volume 1, Basic Techniques*. (Cambridge University Press, 2001)

[64] V. Goyal, Reducing trust in the PKG in identity based cryptosystems, in A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*. (Springer, Heidelberg, August 2007), pp. 430–447

[65] J. Groth, Short pairing-based non-interactive zero-knowledge arguments, in M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*. (Springer, Heidelberg, December 2010), pp. 321–340

[66] J. Groth, Efficient fully structure-preserving signatures for large messages, in T. Iwata and J.H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*. (Springer, Heidelberg, November / December 2015), pp. 239–259

[67] J. Groth, A. Sahai, Efficient non-interactive proof systems for bilinear groups, in N.P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*. (Springer, Heidelberg, 2008), pp. 415–432

[68] C. Hanser, M. Rabkin, D. Schröder, Verifiably encrypted signatures: Security revisited and a new construction, in G. Pernul, P.Y.A. Ryan, and E.R. Weippl, editors, *ESORICS 2015, Part I*, volume 9326 of *LNCS*. (Springer, Heidelberg, September 2015), pp. 146–164

[69] C. Hanser, D. Slamanig, Structure-preserving signatures on equivalence classes and their application to anonymous credentials, in P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*. (Springer, Heidelberg, December 2014), pp. 491–511

[70] M. Izabachène, B. Libert, D. Vergnaud, Block-wise P-signatures and non-interactive anonymous credentials with efficient attributes, in L. Chen, editor, *13th IMA International Conference on Cryptography and Coding*, volume 7089 of *LNCS*. (Springer, Heidelberg, December 2011), pp. 431–450

[71] R. Johnson, D. Molnar, D.X. Song, D. Wagner, Homomorphic signature schemes, in B. Preneel, editor, *CT-RSA 2002*, volume 2271 of *LNCS*. (Springer, Heidelberg, February 2002), pp. 244–262

[72] C.S. Jutla, A. Roy, Improved structure preserving signatures under standard bilinear assumptions, in S. Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*. (Springer, Heidelberg, March 2017), pp. 183–209

[73] E. Kiltz, J. Pan, H. Wee, Structure-preserving signatures from standard assumptions, revisited, in R. Gennaro and M.J.B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*. (Springer, Heidelberg, August 2015), pp. 275–295

[74] A. Kate, G.M. Zaverucha, I. Goldberg, Constant-size commitments to polynomials and their applications, in M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*. (Springer, Heidelberg, December 2010), pp. 177–194

[75] H. Lipmaa, Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments, in R. Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*. (Springer, Heidelberg, March 2012), pp. 169–189

[76] B. Libert, T. Peters, M. Joye, M. Yung, Linearly homomorphic structure-preserving signatures and their applications, in R. Canetti and J.A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*. (Springer, Heidelberg, August 2013), pp. 289–307

[77] A. Lysyanskaya, R.L. Rivest, A. Sahai, S. Wolf, Pseudonym systems, in H.M. Heys and C.M. Adams, editors, *SAC 1999*, volume 1758 of *LNCS*. (Springer, Heidelberg, August 1999), pp. 184–199

[78] R.C. Merkle, A digital signature based on a conventional encryption function, in C. Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*. (Springer, Heidelberg, August 1988), pp. 369–378

[79] S. Micali, M.O. Rabin, J. Kilian, Zero-knowledge sets. In *44th FOCS*. (IEEE Computer Society Press, October 2003), pp. 80–91

[80] T.P. Pedersen, Non-interactive and information-theoretic secure verifiable secret sharing, in J. Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*. (Springer, Heidelberg, 1992), pp. 129–140

[81] D. Pointcheval, O. Sanders, Short randomizable signatures, in K. Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*. (Springer, Heidelberg, February / March 2016), pp. 111–126

[82] S. Ringers, E.R. Verheul, J.-H. Hoepman, An efficient self-blindable attribute-based credential scheme. *IACR Cryptology ePrint Archive*, **2017**, 115, (2017). (to appear at Financial Crypto 2017)

[83] R. Steinfeld, L. Bull, Y. Zheng, Content extraction signatures, in K. Kim, editor, *ICISC 01*, volume 2288 of *LNCS*. (Springer, Heidelberg, December 2002), pp. 285–304

[84] V. Shoup, Lower bounds for discrete logarithms and related problems, in W. Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*. (Springer, Heidelberg, May 1997), pp. 256–266

[85] A. Sudarsono, T. Nakanishi, N. Funabiki, Efficient proofs of attributes in pairing-based anonymous credential system, in *Privacy Enhancing Technologies - 11th International Symposium, PETS 2011, Waterloo, ON, Canada, July 27-29, 2011. Proceedings*, pp. 246–263 (2011)

[86] E.R. Verheul, Self-blindable credential certificates from the Weil pairing, in C. Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*. (Springer, Heidelberg, December 2001), pp. 533–551

[87] B.R. Waters, Efficient identity-based encryption without random oracles, in R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*. (Springer, Heidelberg, May 2005), pp. 114–127