Improved Constructions of Anonymous Credentials From Structure-Preserving Signatures on Equivalence Classes

Aisling Connolly, Pascal Lafourcade, Octavio Perez-Kempner

Improved Constructions of Anonymous Credentials From Structure-Preserving Signatures on Equivalence Classes

Aisling Connolly[1], Pascal Lafourcade[2], and Octavio Perez Kempner[3,4]

[1] Worldline Global
aislingmconnolly@gmail.com
[2] LIMOS, University Clermont Auvergne, France
pascal.lafourcade@uca.fr
[3] DIENS, École normale supérieure, CNRS, PSL University, Paris, France
[4] be-ys Research, France
octavio.perez.kempner@ens.fr

Abstract. Anonymous attribute-based credentials (ABCs) are a powerful tool allowing users to authenticate while maintaining privacy. When instantiated from structure-preserving signatures on equivalence classes (SPS-EQ) we obtain a controlled form of malleability, and hence increased functionality and privacy for the user. Existing constructions consider equivalence classes on the message space, allowing the joint randomization of credentials and the corresponding signatures on them. In this work, we additionally consider equivalence classes on the signing-key space. In this regard, we obtain a *signer-hiding* notion, where the issuing organization is not revealed when a user shows a credential. To achieve this, we instantiate the ABC framework of Fuchsbauer, Hanser, and Slamanig (FHS, Journal of Cryptology '19) with a recent SPS-EQ scheme (ASIACRYPT '19) modified to support a fully adaptive NIZK from the framework of Couteau and Hartmann (CRYPTO '20). We also show how to obtain Mercurial Signatures (CT-RSA, 2019), extending the application of our construction to anonymous delegatable credentials.
To further increase functionality and efficiency, we augment the set-commitment scheme of FHS19 to support openings on attribute sets disjoint from those possessed by the user, while integrating a proof of exponentiation to allow for a more efficient verifier. Instantiating in the CRS model, we obtain an efficient credential system, anonymous under malicious organization keys, with increased expressiveness and privacy, proven secure in the standard model.

Keywords: Anonymous credentials · Mercurial signatures · SPS-EQ

1 Introduction

Considering access to online services, designing protocols to manage the information users can be requested to present is of utmost importance to protect the user. A first step in the literature developed the concept of *attribute-based*

credentials (ABC), to model how users could show a credential, containing a set of attributes, to access different services.

Subsequently, the development of *anonymous* attribute-based credentials made it possible protect the holders identity when showing a credential. Users could present a credential disclosing no information other than that revealed by the attributes they choose to show (*anonymity*), while also ensuring that the provided information is authentic (*unforgeability*). Proposed alternatives consider a third property *unlinkability* which ensures that multiple showings of the same credential cannot be linked. Credential systems that support an arbitrary number of unlinkable showings are said to be *multi-show*. In contrast, those that only allow a single use of an issued credential in an unlinkable fashion are called *one-show*.

Initial progress was made with respect to one-show constructions. Here, *blind signatures* are issued on commitments to attributes so that users can later show the signature and disclose some of the attributes, while proving knowledge of those left unrevealed. Examples include [9, 4], and [30].

In the multi-show setting, pioneering constructions (based on Camenisch and Lysyanskaya's (CL) signatures [12, 13]) such as the one underlying the Idemix credential system [53] rely on randomizing the signature to then prove in zero-knowledge the correspondence between the set of attributes (disclosed and undisclosed), and the signature.

A major drawback from such an approach is that the zero-knowledge proof used during showings is of variable-length and may require multiple sub-proofs On the other hand, more recent constructions (*e.g.,* [14, 11, 47, 39, 50, 24, 33]) apply other techniques based on different lines of work to adapt the signature and the message without using Zero-Knowledge Proofs of Knowledge (ZKPoK), providing constant-size showings.

The concept of ABC has been recently extended to consider multi-authority credentials (*e.g.,* [39, 49, 23]), where users obtain a single credential for a set of attributes not necessarily issued by a single authority. In this work we consider the *classical* setting (single authority issuance).

1.1 Limitations of state-of-the-art ABCs

Constructions in the classical setting differentiate from each other by the expressiveness they provide, their efficiency, on whether or not they provide non-interactive features, on their security model, and on how and if they manage revocation features. Achieving all these properties simultaneously has been challenging and tends to rely on complex or non-standard assumptions.

When considering state-of-the-art credential systems, there are five lines of work with respect to the underlying signature scheme that is used to build them; (1) CL signatures [13]: Idemix [53] and [50]. (2) Aggregatable signatures: [14] and [39]. (3) Sanitizable signatures: [15]. (4) Redactable signatures: [11] and [47]. (5) Structure-Preserving Signatures on Equivalence Classes (SPS-EQ): [33].

Proof settings. All previous work with the exception of [50] rely on security proofs in the Generic Group Model (GGM) [48]. Our first motivation is to provide an alternative to [50], building on [33] without relying on the GGM.

Scheme	$\lvert\sigma\rvert$	$\lvert\mathsf{pk}\rvert$	Sign	Verify	ChgRep	Assumptions
[41]	$8\lvert\mathbb{G}_1\rvert + 9\lvert\mathbb{G}_2\rvert$	$(2+\ell)\lvert\mathbb{G}_2\rvert$	29E	11P	19P+38E	SXDH
Section 5	$9\lvert\mathbb{G}_1\rvert + 4\lvert\mathbb{G}_2\rvert$	$(2+\ell)\lvert\mathbb{G}_2\rvert$	10E	11P	19P+21E	extKerMDH, SXDH

Table 1: Signatures comparison including pairings and exponentiations.

Signer-hiding properties. Showing protocols of previous constructions (including [50]), verify signatures with a key that belongs to the authority that issued the credential. This restricts the use of ABC in scenarios where one would like to verify a valid credential without linking it to a particular authority.

Concrete efficiency. Most alternatives provide similar efficiency at the asymptotic level. Yet, an up-to-date fine-grained analysis on their concrete efficiency lacks in the literature.

1.2 Summary of contributions

We follow the ABC and SPS-EQ line of work from Fuchsbauer, Hanser and Slamanig [33], improving over prior work in the following ways:

1. We extend the set-commitment scheme from [33] to build a more expressive credential system allowing the generation of witnesses for disjoint sets ([33] allows only selective disclosure of attributes).
2. We instantiate the ABC from [33], with a new SPS-EQ scheme based on the one from [41] also using a CRS, a tight reduction, and under weaker assumptions. Thus, we move away from a security proof in the GGM when compared to the work from [33], and obtain a more efficient ABC than the one resulting from instantiating [33] with [41] (see Table 1).
3. We incorporate a proof of exponentiation to outsource part of computational cost from the verifier to the prover, which can be useful in some settings.
4. We adapt the signature scheme to build an SPS-EQ where one not only can randomize the message together with the signature, but also the corresponding public key used to verify the signature using a proof of well-formedness. Thus, users can hide the identity of the *signer* during showings.

By doing so, the verifier can check a signature using a randomized public key, knowing that it comes from a valid authority but not which one. Unlike solutions using ring signatures where it is the signer (credential issuer) who chooses the ring size, we let users do it independently (relying on SPS-EQ and an efficient proof of correct randomization alone). Hence, once users get a credential from a valid authority they can decide on the anonymity set themselves whenever they use their credential. This approach is better aligned with the concept of self sovereign identity and related applications that seek to empower users giving them full control on their identity.

Along the way, we also describe how to build *mercurial signatures* [20] with security proofs in the standard model (assuming a CRS). All the previous ones [20, 21] have security proofs in the GGM. Consequently, our signature construction can also be used to build delegatable anonymous credentials [17, 5] as well.

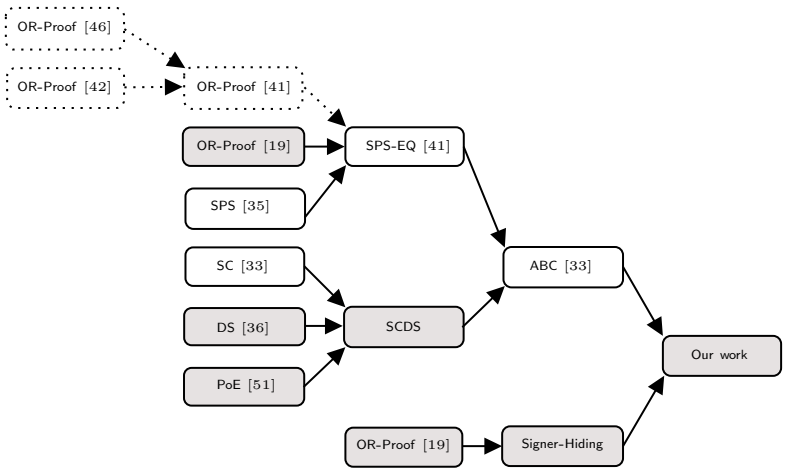

Fig. 1: Summary of building blocks used in this work. Dashed boxes represent replaced building blocks while grey boxes are used to highlight our contributions. When applicable, references inside each box indicate the related previous work.

1.3 Roadmap

We begin by presenting related work with a focus on the development of SPS-EQ and set-commitment schemes (Section 2) followed by the required cryptographic background in Section 3. Our *first* contribution, extending the set-commitment scheme (SC) in [33] to support non-membership proofs for disjoint sets (DS), is presented in Section 4. We also define here the proof of exponentiation (PoE), which can be seen as an optional *plug-in* to gain efficiency in this new set-commitment scheme (SCDS).

In Section 5 we present our SPS-EQ scheme. It uses a new malleable NIZK argument based on a recent work from Couteau and Hartmann [19], which we use to replace the one underlying [41].

In [33] the authors discuss a concurrently secure variant of their ABC based on a trapdoor commitment scheme to implement ZKPoK, assuming the existence of one-way functions and a CRS. Since our SPS-EQ makes use of a CRS, we instantiate the previous variant with it, incorporate a Pedersen commitment scheme to compute the relevant ZKPoK, and adapt the rest to our set-commitment scheme and the proof of exponentiation (*second* and *third* contributions). Thus, we dedicate Section 6 to present the resulting ABC.

Subsequently, we extend the previous construction to support another NIZK argument that allows to hide the identity of the signer during showings. This allows us to build another ABC as our *fourth* contribution. Furthermore, we also outline in this section how to perform revocation and build mercurial signatures.

In Figure 1 we summarize the dependencies between the different building blocks used in the previously mentioned sections highlighting our contributions.

Finally, a detailed comparison on the concrete efficiency of our constructions when compared to other state-of-the-art alternatives is provided in Section 8, while the conclusions of this work are presented in Section 9.

2 Background and Related Work

2.1 Structure-Preserving Signatures on Equivalence Classes

In [37], Hanser and Slamanig introduced a novel structure preserving signature (SPS) scheme that allowed joint randomization of messages and their corresponding signatures, coining Structure-Preserving Signatures on Equivalence Classes (SPS-EQ). They observed that if one considers a prime-order group $\mathbb{G}$ and defines the projective vector space $(\mathbb{G}^*)^\ell$, there is a partition into equivalence classes given by the following relation $\mathcal{R}$: $\mathbf{m} \in (\mathbb{G}^*)^\ell \sim_{\mathcal{R}} \mathbf{m}^* \in (\mathbb{G}^*)^\ell \iff \exists\ \mu \in \mathbb{Z}_p^* : \mathbf{m}^* = \mu\mathbf{m}$. If the discrete logarithm problem is hard in $\mathbb{G}$ and one restricts the vector components to be non-zero, given two vectors $\mathbf{m}$ and $\mathbf{m}^*$, it is difficult to distinguish whether they were randomly sampled or if they belong to the same equivalence class. Hence, Hanser and Slamanig defined SPS-EQ as SPS that produce signatures on an equivalence class instead of messages alone. Given a message and its corresponding signature, SPS-EQ provides a *controlled form of malleability* in which one can publicly (without requiring access to the secret key) adapt a signature to change the representative (message). The equivalence relation provides indistinguishability on the message space if the DDH assumption holds. If additionally, updated signatures are distributed like fresh signatures, message-signature pairs falling into the same class are unlinkable. For unlinkability to hold, signatures should also be randomized when adapting them to a new representative of the class. As described in [33], given a representative and its corresponding signature, a random representative of the same class with an adapted signature are indistinguishable from a random message-signature pair.

Since their introduction, SPS-EQ have been used to build several cryptographic protocols (*e.g.,* [2, 32, 31, 3, 26, 10, 29]). They have been used in anonymous credentials [37, 24, 33], and delegatable anonymous credential systems, in this case under the name of mercurial signatures [20, 21], which are an extension of the equivalence classes to the signing keys. State-of-the-art constructions focus on building schemes under weaker assumptions and with tight security. The first step was the work from Fuchsbauer and Gay [28]. Subsequently, Khalili *et al.* [41] proposed a new SPS-EQ which is, to the best of our knowledge, the only one under standard assumptions and with a tight security reduction to date.

The construction of [28] is based on the family of Matrix-Diffie-Hellman assumptions [27]. They first modify an affine MAC from [6] to obtain a linear structure-preserving MAC, which is made publicly verifiable using a known technique in the context of SPS [42]. This allows to use a *tag* to randomize both the signature and message.

The resulting scheme is secure under a weaker notion of unforgeability (EUF-CoMA). In [41], authors observe that using a structure-preserving MAC such as the one from [28] has an inherent problem in the security game. As messages and Matrix Decision Diffie-Hellman challenges belong to the same source group of the bilinear group, one cannot do better than EUF-CoMA security following this approach. Consequently, they proposed to use an OR-Proof based on that in [35] to then construct tightly secure structure-preserving MACs based on the key encapsulation mechanism of Gay *et al.* in [34]. This allows to circumvent

the previous issue and obtain the first EUF-CMA secure SPS-EQ scheme with a tight security reduction under standard assumptions.

In this work, we present an SPS-EQ scheme where the OR-based proof in [41] is replaced by the one in [19], while adapting other building blocks accordingly.

2.2 Accumulators and Set-Commitments.

In [25], Derler, Hanser and Slamanig revisited the notion of cryptographic accumulators and proposed a unified formal model which included the notions of undeniability and indistinguishability for accumulators, complementing the classical ones of correctness and collision-freeness. They showed how to construct a commitment scheme using an indistinguishable accumulator in a black-box manner. The relation stems from the fact that indistinguishability and collision-freeness of accumulators resemble those of hiding and binding for commitments.

In subsequent work [37], Hanser and Slamanig built an ABC with constant-size credentials and constant-size showings (for selective disclosure of attributes) based on a polynomial commitment scheme with factor openings. They departed from the work of Kate *et al.* on constant-size polynomial commitments [40] with the following observations; (1) If a credential is seen as a set of attributes mapped to roots of a monic polynomial, then one can generate a polynomial commitment of constant-size to represent the credential using the approach from [40]. (2) Instead of evaluating the polynomial at certain points, what is important to prove possession of an attribute is to open factors of the polynomial instead. (3) If one can open multiple factors in constant-size, a showing involving a selective disclosure of attributes can be done in constant-size as well.
As a result they proposed an indistinguishable bilinear accumulator ([44]) with batch membership proofs (*i.e,* factor opening), which was subsequently re-stated as a *set-commitment* scheme in a follow-up work [33].

A drawback of the ABC from [33] is that the achieved level of *expressiveness* is limited. It allows only to show proofs for the conjunction of attributes in arbitrary subsets of attributes encoded in the credential (selective disclosure). Another potential issue is that verification involves a number of exponentiations that are linear in the size of the subset to be verified. This is undesirable when verification of the credential should be fast.

Thakur [51] proposed a series of protocols for batch membership and non-membership proofs for bilinear accumulators using *proofs of exponentiation* (an idea previously introduced for accumulators in groups of unknown order by Boneh et. al [8] and by Wesolowski [52]) to shift the computational cost from the verifier to the prover. The main idea is to replace some of the exponentiations by a single polynomial division and to use of a non-interactive proof obtained via the Fiat-Shamir transform.

Batch proofs in the bilinear accumulator setting can be traced back to the works by Papamanthou *et al.* [45] and by Ghosh *et al.* [36]. The latter presents the same underlying ideas of the (non)membership proofs provided by Thakur, and a *Zero-Knowledge Dynamic Universal Accumulator*, which strengthens the notion of indistinguishability using the randomization ideas from [25].

Scheme	[13]	[14]	[15]	[11] & [33]	[50]	[47]	[39]	Section 6
	Issuing n-attr. credential							
Comm.	$O(n)$	$O(n)$	$O(n)$	$\boldsymbol{O(1)}$	$O(n)$	$\boldsymbol{O(1)}$	$O(n)$	$\boldsymbol{O(1)}$
User	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Issuer	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
	Showing k-of-n attributes (selective disclosure)							
$\lvert\mathsf{ek}\rvert$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n^2)$	$O(n)$	$O(n)$
Comm.	$O(n)$	$O(1)$	$O(k)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$
User	$O(n)$	$O(n)$	$O(k)$	$O(n-k)$	$O(n-k)$	$O(n-k)$	$\boldsymbol{O(1)}$	$O(\max\{n-k,k\})$
Verifier	$O(n)$	$O(n)$	$O(k)$	$O(k)$	$O(k)$	$O(k)$	$O(n)$	$\boldsymbol{O(1)}$

Table 2: Asymptotic complexities of ABC systems where n is the number of attributes in the credential and k the number of disclosed ones during a showing.

More recently, a new set-commitment scheme including set intersection and set difference operations was proposed in [50]. It provides more expressiveness when compared to the one from [33] but under a weaker hiding notion.

We incorporate the previous ideas from [25, 36], and [51] to extend the set-commitment scheme from [33] to support disjoint sets (batch non-membership proofs), while also allowing a faster verification and a stronger hiding notion. Thus, we obtain a set-commitment scheme that is more expressive than the one in [33] and almost as expressive as [50] (but better in efficiency and strength).

2.3 Attribute-based Credentials

We recall in Table 2 the asymptotic complexities for the issuing and showing protocols, considering recent credential systems from each of the lines of work mentioned in the introduction, and our construction in Section 6 . For showing protocols we consider the selective disclosure of attributes (*i.e.,* the ability to show multiple attributes while hiding others during a showing). While the work from [39] (based on aggregatable signatures) is the only one with $O(1)$ complexity for the user during a showing, this is obtained at the cost of a more expensive verifier. Our work achieves $O(1)$ complexity for the verifier but keeping better asymptotics for the user. A more detailed comparison on the concrete efficiency of ABC's (as well as an implementation benchmark) was provided in [50], but the recent works from [47] and [39] were not included. Therefore, we provide an updated comparison for the most efficient ones in Section 8.

2.4 Signer-Hiding

Independent and concurrent work by Bobolz *et al.* [7] also addressed the problem of hiding the identity of a credential issuer/signer under the notion of *issuer-hiding*. There, the authors propose a sligthly different setting to avoid using an OR-like proof as done in this work. In brief, the authors consider access policies of the form $\{\sigma_i, \mathsf{pk}_i\}_{i\in[n]}$, where σ_i is a signature on a given authority's public key pk_i produced by the verifier. As a result, users can prove the correspondence between a public key (defined in the policy) and the credential verification under that public key in zero knowledge, using a NIZK independent to the number

of public keys defined in the policy. In this regard, we note that our work is compatible with their formalization and, furthermore, under the previous setting such NIZK can be avoided in our case. Since we use mercurial signatures, it would suffice to randomize the access policy and the user credential consistently.

3 Preliminaries

NOTATION. Let BGGen be a p.p.t algorithm that on input 1^λ with λ the security parameter, returns a description $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e)$ of an asymmetric bilinear group where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are cyclic groups of prime order p with $\lceil \log_2 p \rceil = \lambda$, P_1 and P_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable (non-degenerate) bilinear map. BG is said to be of Type-3 if no efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$ are known. For all $a \in \mathbb{Z}_p$, we denote by $[a]_s = aP_s \in \mathbb{G}_s$ the implicit representation of a in $\mathbb{G}_s$ for $s \in \{1, 2, T\}$. For matrices (or vectors) $\mathbf{A}$, $\mathbf{B}$ we extend the pairing notation to $e([\mathbf{A}]_1, [\mathbf{B}]_2) := [\mathbf{AB}]_T \in \mathbb{G}_T$. Sampling r from set $\mathcal{S}$ uniformly at random is denoted by $r \xleftarrow{\$} \mathcal{S}$. Finally, we use the notation $\mathcal{A}(x; y)$ to indicate that a value y (usually computed internally by $\mathcal{A}$), is being passed directly to $\mathcal{A}$ on input x.

ASSUMPTIONS. We recall the Diffie-Hellman assumptions in the billinear group setting and the algebraic framework from [27] and [43], including a generalization of the Strong Diffie-Hellman assumption from [33], in the full version (Appendix A from [18]). Besides, we will also use the following generalization of the KerMDH assumption introduced in [19]. It allows an adversary to extend the given matrix but requiring it to output multiple, linearly independent vectors in the kernel.

$\mathcal{D}_k$-$\mathsf{extKerMDH}$ASSUMPTION. Let $\mathcal{D}_k$ be a matrix distribution, $l, k \in \mathbb{N}$, and $s \in \{1, 2\}$. We say that the $\mathcal{D}_k$-$\mathsf{extKerMDH}$ *assumption* holds in $\mathbb{G}_s$ relative to BGGen, if for every $\mathsf{BG} \xleftarrow{\$} \mathsf{BGGen}(1^\lambda)$, $\mathbf{D} \xleftarrow{\$} \mathcal{D}_k$, and all p.p.t. adversaries $\mathcal{A}$ the following probability is negligible.

$$\mathsf{Pr}\left[\begin{array}{c|c} [\mathbf{C}]_{3-s} \in \mathbb{G}_3^{l+1\times k+l+1} \wedge [\mathbf{B}]_s \in \mathbb{G}_s^{l\times k} & \mathsf{BG} \xleftarrow{\$} \mathsf{BGGen}(1^\lambda); \mathbf{D} \xleftarrow{\$} \mathcal{D}_k \\ \wedge\ [\mathbf{C}]_{3-s}[\mathbf{D'}]_s = 0 & ([\mathbf{C}]_{3-s}, [\mathbf{B}]_s) \xleftarrow{\$} \mathcal{A}(\mathsf{BG}, [\mathbf{D}]_s) \\ \wedge\ \mathsf{rank}(\mathbf{C}) \geq l+1 & [\mathbf{D'}]_s := \left[\begin{smallmatrix}\mathbf{D}\\ \mathbf{B}\end{smallmatrix}\right]_s \end{array}\right]$$

CHARACTERISTIC POLYNOMIAL. For a set $\mathcal{X}$ with elements in $\mathbb{Z}_p$, we refer to $\mathsf{Ch}_\mathcal{X}(X) = \prod_{x\in\mathcal{X}}(X + x) = \sum_{i=0}^{i=n} c_i \cdot X^i$ (a monic polynomial of degree $n = |\mathcal{X}|$ and defined over $\mathbb{Z}_p[X]$) as its *characteristic polynomial*. For a group generator P, $\mathsf{Ch}_\mathcal{X}(s)P$ can be efficiently computed (*e.g.,* using the Fast Fourier Transform) when given $(s^i P)_{i=0}^{|\mathcal{X}|}$ but not s. This is because $\mathsf{Ch}_\mathcal{X}(s)P = \sum_{i=0}^{i=n}(c_i \cdot s^i)P$.

In addition to exploiting properties of characteristic polynomials, we will also use the Schwartz-Zippel lemma and the Extended Euclidean Algorithm (EEA) in our constructions following the ideas from [36].

Lemma 1 (Schwartz-Zippel) Let $q_1(x), q_2(x)$ be two d-degree polynomials from $\mathbb{Z}_p[X]$ with $q_1(x) \neq q_2(x)$, then for $s \xleftarrow{\$} \mathbb{Z}_p$, the probability that $q_1(x) = q_2(x)$ is at most d/p, and the equality can be tested in time $O(d)$.

3.1 Non-interactive Zero-Knowledge Arguments and Malleable Proof Systems

We next define fully adaptive NIZK arguments (*i.e.,* the crs does not depend on the language distribution or language parameters), and the notions of malleable proof systems given in [16] and [41] respectively.

NIZK Syntax. A fully adaptive NIZK Π for a family of language distribution $\{\mathcal{D}_{\mathsf{pp}}\}_{\mathsf{pp}}$ consists of four probabilistic algorithms:

$\mathsf{PGen}(1^\lambda)$: On input 1^λ generates public parameters pp, a crs and a trapdoor td.
$\mathsf{Prove}(\mathsf{crs}, \rho, x, w)$: On input a crs, a language description $\rho \in \mathcal{D}_{\mathsf{pp}}$ and a statement x with witness w, outputs a proof π for $x \in \mathcal{L}_\rho$.
$\mathsf{Verify}(\mathsf{crs}, \rho, x, \pi)$: On input a crs, a language description $\rho \in \mathcal{D}_{\mathsf{pp}}$, a statement x and a proof π, accepts or rejects the proof.
$\mathsf{SimProve}(\mathsf{crs}, \mathsf{td}, \rho, x)$: Given a crs, the trapdoor td, a language description $\rho \in \mathcal{D}_{\mathsf{pp}}$ and a statement x, outputs a simulated proof for the statement $x \in \mathcal{L}_\rho$.

The following properties need to hold for NIZK arguments with respect to a family of language distributions $\mathcal{D}_{\mathsf{pp}}$.

Perfect Completeness.

$$\mathsf{Pr}\left[\mathsf{Verify}(\mathsf{crs}, \rho, x, \pi) = 1 \;\middle|\; \begin{array}{c}(\mathsf{pp}, \mathsf{crs}, \mathsf{td}) \xleftarrow{\$} \mathsf{PGen}(1^\lambda); \rho \in \mathsf{Supp}(\mathcal{D}_{\mathsf{pp}}); \\ (x, w) \in R_\rho; \pi \xleftarrow{\$} \mathsf{Prove}(\mathsf{crs}, \rho, x, w)\end{array}\right] = 1$$

Computational Soundness. For every efficient adversary $\mathcal{A}$,

$$\mathsf{Pr}\left[\begin{array}{c}\mathsf{Verify}(\mathsf{crs}, \rho, x, \pi) = 1 \\ \wedge\ x \notin \mathcal{L}_\rho\end{array} \;\middle|\; \begin{array}{c}(\mathsf{pp}, \mathsf{crs}, \mathsf{td}) \xleftarrow{\$} \mathsf{PGen}(1^\lambda); \\ \rho \in \mathsf{Supp}(\mathcal{D}_{\mathsf{pp}}); (\pi, x) \xleftarrow{\$} \mathcal{A}(\mathsf{crs}, \rho)\end{array}\right] \approx 0$$

where the probability is taken over PGen.

Perfect Zero-Knowledge. For all λ, all $(\mathsf{pp}, \mathsf{crs}, \mathsf{td}) \in \mathsf{Supp}(\mathsf{PGen}(1^\lambda))$, all $\rho \in \mathsf{Supp}(\mathcal{D}_{\mathsf{pp}})$ and all $(x, w) \in R_\rho$, the distributions $\mathsf{Prove}(\mathsf{crs}, \rho, x, w)$ and $\mathsf{SimProve}(\mathsf{crs}, \mathsf{td}, \rho, x)$ are identical.

Let $\mathcal{R}_\mathcal{L}$ be the witness relation associated to a language $\mathcal{L}$, then a controlled malleable proof system is accompanied by a family of efficiently computable n-ary transformations $T = (T_x, T_w)$ such that for any n-tuple $\{(x_1, w_1), \ldots, (x_n, w_n)\}$ $\in \mathcal{R}_\mathcal{L}^n$ it holds that $(T_x(x_1, \ldots, x_n), T_w(w_1, \ldots, w_n)) \in \mathcal{R}_\mathcal{L}$. Intuitively, such a proof system allows when given valid proofs $\{\Omega_i\}_{i\in[n]}$ for words $\{x_i\}_{i\in[n]}$ with associated witnesses $\{w_i\}_{i\in[n]}$ to publicly compute a valid proof Ω for word $x := T_x(x_1, \ldots, x_n)$ corresponding to witness $w := T_w(w_1, \ldots, w_n)$ using an additional algorithm ZKEval which is defined as follows:

$\mathsf{ZKEval}(\mathsf{crs}, \mathcal{T}, (x_i, \Omega_i)_{i\in[n]})$ takes as input a common reference string crs, a transformation $T \in \mathcal{T}$, words $x_1, \ldots, x_n$ and their corresponding proofs $\Omega_1, \ldots, \Omega_n$, and outputs a new word $x' := T_x(x_1, \ldots, x_n)$ and proof Ω'.

Proofs computed by ZKEval should be indistinguishable from freshly computed proofs for the resulting word x' and corresponding witness w'. This notion is captured by the following definition.

Derivation Privacy. A NIZK proof system Π, malleable with respect to a set of transformations $\mathcal{T}$ defined on some relation $\mathcal{R}$ is *derivation private*, if for all p.p.t adversaries $\mathcal{A}$, the following probability is negligible,

$$\mathsf{Pr}\left[\begin{array}{l}\mathsf{crs} \xleftarrow{\$} \mathsf{PGen}(1^\lambda), b \xleftarrow{\$} \{0,1\} \\ (\mathsf{st}, ((x_i, w_i), \Omega_i)_{i\in[q]}, T) \xleftarrow{\$} \mathcal{A}(\mathsf{crs}), \\ \textbf{if } (T \notin \mathcal{T} \vee (\exists\ i \in [q] : (\mathsf{Verify}(\mathsf{crs}, x_i, \Omega_i) = 0) \vee (x_i, w_i) \notin \mathcal{R}) \\ \textbf{return } \bot, \\ \textbf{else if } b = 0 : \Omega \leftarrow \mathsf{Prove}(\mathsf{crs}, T_x((x_i)_{i\in[q]}), T_w((w_i)_{i\in[q]})), \\ \textbf{else if } b = 1 : \Omega \leftarrow \mathsf{ZKEval}(\mathsf{crs}, T, (x_i, \pi_i)_{i\in[q]}), \\ b' \xleftarrow{\$} \mathcal{A}(\mathsf{st}, \Omega)\end{array} : b = b'\right]$$

4 A Set-Commitment Scheme supporting Disjoint Sets

We extend the set-commitment scheme in [33] to support *non-membership proofs* for disjoint sets, while also including an optional *proof of exponentiation* to replace most of the exponentiations in the verifier (outsourcing them to the prover) with a single polynomial division. To do so, we borrow the previously mentioned ideas in [25], [36] and [51], and adapt them to the Type-3 setting.

SCDS Syntax. A *set-commitment scheme supporting disjoint sets* (SCDS) consists of the following p.p.t algorithms:

$\mathsf{Setup}(1^\lambda, 1^q)$ is a probabilistic algorithm which takes as input a security parameter λ and an upper bound q for the cardinality of committed sets, both in unary form. It outputs public parameters pp (including an evaluation key ek), and discards the trapdoor key s used to generate them. $\mathbb{Z}_p^* \setminus \{s\}$ defines the domain of set elements for sets of maximum cardinality q.

$\mathsf{TSetup}(1^\lambda, 1^q)$ is equivalent to Setup but also returns the trapdoor key.

$\mathsf{Commit}(\mathsf{pp}, \mathcal{X})$ is a probabilistic algorithm which takes as input pp and a set $\mathcal{X}$ with $1 \leq |\mathcal{X}| \leq q$. It outputs a commitment C on set $\mathcal{X}$ and opening information O.

$\mathsf{Open}(\mathsf{pp}, C, \mathcal{X}, O)$ is a deterministic algorithm which takes as input pp, a commitment C, a set $\mathcal{X}$, and opening information O. It outputs 1 if and only if O is a valid opening of C on $\mathcal{X}$.

$\mathsf{OpenSS}(\mathsf{pp}, C, \mathcal{X}, O, \mathcal{S})$ is a deterministic algorithm which takes as input pp, a commitment C, a set $\mathcal{X}$, opening information O, and a non-empty set $\mathcal{S}$. If $\mathcal{S}$ is a subset of $\mathcal{X}$ committed to in C, OpenSS outputs a witness wit that attests to it. Otherwise, outputs $\bot$.

$\mathsf{OpenDS}(\mathsf{pp}, C, \mathcal{X}, O, \mathcal{D})$ is a deterministic algorithm which takes as input pp, a commitment C, a set $\mathcal{X}$, opening information O, and a non-empty set $\mathcal{D}$. If $\mathcal{D}$ is disjoint from $\mathcal{X}$ committed to in C, OpenDS outputs a witness $\underline{\mathsf{wit}}$ that attests to it. Otherwise, outputs $\bot$.

$\mathsf{VerifySS}(\mathsf{pp}, C, \mathcal{S}, \mathsf{wit})$ is a deterministic algorithm which takes as input pp, a commitment C, a non-empty set $\mathcal{S}$, and a witness wit. If wit is a valid witness for $\mathcal{S}$ a subset of the set committed to in C, it outputs 1 and otherwise $\bot$.

$\mathsf{VerifyDS}(\mathsf{pp}, C, \mathcal{D}, \underline{\mathsf{wit}})$ takes as input pp, a commitment C, a non-empty set $\mathcal{D}$, and a witness wit. If $\underline{\mathsf{wit}}$ is a valid witness for $\mathcal{D}$ being disjoint from the set committed to in C, it outputs 1 and otherwise $\perp$.

$\mathsf{PoE}(\mathsf{pp}, \mathcal{X}, \alpha)$ takes as input pp, a non-empty set $\mathcal{X}$, and a randomly-chosen value α. It computes a proof of exponentiation for the characteristic polynomial of $\mathcal{X}$ and outputs a proof π_Q and a witness Q.

A SCDS scheme is *secure* if it satisfies the properties of correctness, binding, hiding, and soundness. These notions are defined next, modified to suit the scheme, but following the usual convention.

CORRECTNESS. An SCDS scheme is *correct* if for all $q > 0$, all $\lambda > 0$, all $\mathsf{pp} \in [\mathsf{Setup}(1^\lambda, 1^q)]$, all non-empty $\mathcal{S} \subseteq \mathcal{X}$ and all non-empty $\mathcal{D} : \mathcal{D} \cap \mathcal{X} = \emptyset$, the following probabilities equal 1:

1. $\mathsf{Pr}\left[(C, O) \xleftarrow{\$} \mathsf{Commit}(\mathsf{pp}, \mathcal{X}) \ : \ \mathsf{Open}(\mathsf{pp}, C, \mathcal{X}, O) = 1\right]$
2. $\mathsf{Pr}\left[\begin{matrix}(C, O) \xleftarrow{\$} \mathsf{Commit}(\mathsf{pp}, \mathcal{X}); \\ \mathsf{wit} \leftarrow \mathsf{OpenSS}(\mathsf{pp}, C, \mathcal{X}, O, \mathcal{S})\end{matrix} \ : \ \mathsf{VerifySS}(\mathsf{pp}, C, \mathcal{S}, \mathsf{wit}) = 1\right]$
3. $\mathsf{Pr}\left[\begin{matrix}(C, O) \xleftarrow{\$} \mathsf{Commit}(\mathsf{pp}, \mathcal{X}); \\ \underline{\mathsf{wit}} \leftarrow \mathsf{OpenDS}(\mathsf{pp}, C, \mathcal{X}, O, \mathcal{D})\end{matrix} \ : \ \mathsf{VerifyDS}(\mathsf{pp}, C, \mathcal{D}, \underline{\mathsf{wit}}) = 1\right]$

BINDING. An SCDS scheme is *binding* if for all $q > 0$ and all p.p.t adversaries $\mathcal{A}$, the following probability is negligible,

$$\mathsf{Pr}\left[\begin{matrix}\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, 1^q), \\ (C, \mathcal{X}, O, \mathcal{X}', O') \xleftarrow{\$} \mathcal{A}(\mathsf{pp})\end{matrix} \ : \ \begin{matrix}\mathsf{Open}(\mathsf{pp}, C, \mathcal{X}, O) = 1 \ \wedge \\ \mathsf{Open}(\mathsf{pp}, C, \mathcal{X}', O') = 1 \ \wedge \ \mathcal{X} \neq \mathcal{X}'\end{matrix}\right]$$

HIDING. We say that an SCDS scheme is *hiding* if for all $q > 0$ and all p.p.t adversaries $\mathcal{A}$ with access to $\mathcal{O}_{\mathsf{SS}}$, an opening oracle which allows queries for sets $\mathcal{X}' \subseteq \mathcal{X}_0 \cap \mathcal{X}_1$, and to $\mathcal{O}_{\mathsf{DS}}$, for sets $\mathcal{X}'$ s.t. $\mathcal{X}' \cap \{\mathcal{X}_0 \cup \mathcal{X}_1\} = \emptyset$, there is a negligible function $\epsilon(\cdot)$ such that:

$$\mathsf{Pr}\left[\begin{matrix}b \xleftarrow{\$} \{0,1\}; \mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, 1^q); \\ (\mathcal{X}_0, \mathcal{X}_1, \mathsf{st}) \xleftarrow{\$} \mathcal{A}(\mathsf{pp}); \\ (C, O) \xleftarrow{\$} \mathsf{Commit}(\mathsf{pp}, \mathcal{X}_b); \\ b^* \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{SS}}(\mathsf{pp}, C, \mathcal{X}_b, O, \cdot), \mathcal{O}_{\mathsf{DS}}(\mathsf{pp}, C, \mathcal{X}_b, O, \cdot)}(\mathsf{st}, C)\end{matrix} \ : \ b^* = b\right] - \frac{1}{2} \leq \epsilon(k).$$

where $\mathcal{X}_0$ and $\mathcal{X}_1$ are two distinct sets s.t. $1 \leq |\mathcal{X}_b| \leq q$.
If the above holds for $\epsilon \equiv 0$, the scheme is said to be perfectly hiding.

SOUNDNESS. An SCDS scheme is sound if for all $q > 0$ and all p.p.t adversaries $\mathcal{A}$, the following probabilities are negligible,

1. $\mathsf{Pr}\left[\begin{matrix}\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, 1^q); \\ (C, \mathcal{X}, O, \mathcal{S}, \mathsf{wit}) \xleftarrow{\$} \mathcal{A}(\mathsf{pp})\end{matrix} : \begin{matrix}\mathcal{S} \nsubseteq \mathcal{X} \ \wedge \ \mathsf{OpenSS}(C, \mathcal{X}, O) = 1 \\ \wedge \ \mathsf{VerifySS}(C, \mathcal{S}, \mathsf{wit}) = 1\end{matrix}\right]$
2. $\mathsf{Pr}\left[\begin{matrix}\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, 1^q); \\ (C, \mathcal{X}, O, \mathcal{D}, \underline{\mathsf{wit}}) \xleftarrow{\$} \mathcal{A}(\mathsf{pp})\end{matrix} : \begin{matrix}\mathcal{D} \cap \mathcal{X} \neq \emptyset \ \wedge \ \mathsf{OpenDS}(C, \mathcal{X}, O) = 1 \\ \wedge \ \mathsf{VerifyDS}(C, \mathcal{D}, \underline{\mathsf{wit}}) = 1\end{matrix}\right]$

$\underline{\mathsf{SCDS.Setup}(1^\lambda, 1^q)}$:
$\mathsf{BG} \xleftarrow{\$} \mathsf{BGGen}(1^\lambda)$; $s \xleftarrow{\$} \mathbb{Z}_p^*$
$\mathsf{pp} \leftarrow (\mathsf{BG}, (s^i P_1, s^i P_2)_{i\in[q]})$
return pp

$\underline{\mathsf{SCDS.TSetup}(1^\lambda, 1^q)}$:
$\mathsf{BG} \xleftarrow{\$} \mathsf{BGGen}(1^\lambda)$; $s \xleftarrow{\$} \mathbb{Z}_p^*$
$\mathsf{pp} \leftarrow (\mathsf{BG}, (s^i P_1, s^i P_2)_{i\in[q]})$
return (pp, s)

$\underline{\mathsf{SCDS.PoE}(\mathsf{pp}, \mathcal{X}, \alpha)}$:
$Q \leftarrow \mathsf{Ch}_\mathcal{X}(s)P_2$; Let $h(X)$ and β s.t.
$\mathsf{Ch}_\mathcal{X}(X)=(X+\alpha)\cdot h(X)+\beta$; $\pi_Q \leftarrow h(s)P_2$
return (π_Q, Q)

$\underline{\mathsf{SCDS.Commit}(\mathsf{pp}, \mathcal{X})}$:
if $|\mathcal{X}| > q$ **return** $\perp$; $r \xleftarrow{\$} \mathbb{Z}_p^*$
if $\exists\ s' \in \mathcal{X} : s'P_1 = sP_1$
  $C \leftarrow rP_1$; $O \leftarrow (1, (r, s'))$
else $C \leftarrow r \cdot \mathsf{Ch}_\mathcal{X}(s)P_1$; $O \leftarrow (0, r)$
return (C, O)

$\underline{\mathsf{SCDS.Open}(\mathsf{pp}, C, \mathcal{X}, O)}$:
if $O = (1, (r, s')) \wedge\ s'P_1 = sP_1$
  if $C = rP_1$ **return** 1 **else** 0
if $O = (0, r)$
  if $C = r \cdot \mathsf{Ch}_\mathcal{X}(s)P_1$ **return** 1 **else** 0

$\underline{\mathsf{SCDS.OpenSS}(\mathsf{pp}, C, \mathcal{X}, O, \mathcal{S})}$:
if $\underline{\mathsf{SCDS.Open}(C, \mathcal{X}, O) = 0 \vee}$
  $\mathcal{S} \nsubseteq \mathcal{X} \vee \mathcal{S} = \emptyset$ **return** $\perp$
if $O = (1, (r, s'))$
  if $s' \notin \mathcal{S}$ **return** $\mathsf{Ch}_\mathcal{S}(s')^{-1}C$
if $O = (0, r)$ **return** $r \cdot \mathsf{Ch}_{\mathcal{X}\setminus\mathcal{S}}(s)P_1$
else return $\perp$

$\underline{\mathsf{SCDS.VerifySS}(\mathsf{pp}, C, \mathcal{S}, \mathsf{wit}, [\mathsf{PoE}])}$:
if $(\mathcal{S} = \emptyset \wedge \mathsf{wit} = \perp)$ **return** 1
if $\exists\ s' \in \mathcal{S} : s'P_1 = sP_1$
  if $\mathsf{wit} = \perp$ **return** 1 **else** 0
if $\mathsf{PoE} = \perp$
  return $e(\mathsf{wit}, \mathsf{Ch}_\mathcal{S}(s)P_2) = e(C, P_2)$
else
  parse $\mathsf{PoE} = (\alpha, \pi_Q, Q)$
  $\beta \leftarrow \mathsf{Ch}_\mathcal{S}(X) (\mathrm{mod}\ (X + \alpha))$
  return $e(sP_1+\alpha P_1, \pi_Q)+e(\beta P_1, P_2)$
  $= e(P_1, Q) \wedge e(\mathsf{wit}, Q) = e(C, P_2)$

$\underline{\mathsf{SCDS.OpenDS}(\mathsf{pp}, C, \mathcal{X}, O, \mathcal{D})}$:
if $(t = 0 \vee |\mathcal{D} \cap \mathcal{X}| > 0)$ **return** $\perp$
if $O = (1, (r, s'))$
  if $s' \in \mathcal{D}$ **return** $\perp$ **else**
  $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$; $(w_0, w_1) \leftarrow (\gamma P_2, \frac{1-\gamma\cdot r}{\mathsf{Ch}_\mathcal{D}(s)} P_1)$
if $O = (0, r)$
  $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$; Let $q_1(X)$ and $q_2(X)$ s.t.
  $\mathsf{Ch}_\mathcal{X}(X) \cdot q_1(X) + \mathsf{Ch}_\mathcal{D}(X) \cdot q_2(X) = 1$
  $q_1'(s) \leftarrow q_1(s) + \gamma \cdot \mathsf{Ch}_\mathcal{D}(s)$
  $q_2'(s) \leftarrow q_2(s) - \gamma \cdot \mathsf{Ch}_\mathcal{X}(s)$
  $(w_0, w_1) \leftarrow ((r^{-1} \cdot q_1'(s))P_2, q_2'(s)P_1)$
return (w_0, w_1)

$\underline{\mathsf{SCDS.VerifyDS}(\mathsf{pp}, C, \mathcal{D}, \underline{\mathsf{wit}}, [\mathsf{PoE}])}$:
if $(\mathcal{D} = \emptyset \wedge \underline{\mathsf{wit}} = \perp)$ **return** 1
if $\exists\ s' \in \mathcal{D} : s'P_1 = sP_1$
  if $\underline{\mathsf{wit}} = \perp$ **return** 1 **else** 0
parse $\underline{\mathsf{wit}} = (w_0, w_1)$
if $\mathsf{PoE} = \perp$ **return**
  $e(C, w_0)+e(w_1, \mathsf{Ch}_\mathcal{D}(s)P_2)=e(P_1, P_2)$
else
  parse $\mathsf{PoE} = (\alpha, \pi_Q, Q)$
  $\beta \leftarrow \mathsf{Ch}_\mathcal{D}(X) (\mathrm{mod}\ (X + \alpha))$
  return $e(sP_1+\alpha P_1, \pi_Q)+e(\beta P_1,P_2)$
  $=e(P_1, Q) \wedge e(C, w_0)+e(w_1, Q)=e(P_1, P_2)$

Fig. 2: Our SCDS construction

4.1 Construction

Our construction is presented in Figure 2. As in [33] we use a special opening for the case in which the commited set contains the trapdoor to achieve perfect correctness and perfect hiding. To prove that a given set is disjoint with respect to the commited set, the EEA is computed to obtain the Bézout coefficients. This way, equality is checked randomizing q_1, q_2 and using a single PPE. Finally, the PoE computes a polynomial division, and produces the corresponding proof.

Theorem 1. The SCDS construction from Figure 2 is correct and perfectly hiding. Furthermore, if the q-co-DL (resp. q-co-GSDH) assumption holds, SCDS is computationally binding (resp. sound).

Proof. The proof strategy follows closely that of [33]. We extend these proofs in a similar manner to consider disjoint sets. The full proof is provided in [18] (Appendix B).

5 Our SPS-EQ construction

The starting point for the SPS-EQ construction in [41] was the tightly secure SPS from [35], which builds on a structure-preserving MAC (based on the works from [34] and [38]) and a NIZK OR-Proof from [46]. To couple with equivalence classes, the authors proposed a way to adapt the OR-Proof so that it could be randomized and malleable. Unfortunately, as the CRS used in the OR-Proof from [46] was incompatible with the required randomization properties, the authors were forced to build a QA-NIZK on top to overcome the limitation.

In this section we introduce a new SPS-EQ scheme based on the one from [41], which we obtain replacing the underlying OR-Proof from [46] with one given in [19], while adapting accordingly. As a result we obtain a more efficient signature scheme based on a new malleable OR-NIZK argument. Before giving the intuition of our construction, we recall the syntax and security properties for SPS-EQ introduced in [33] and [41].

SPS-EQ Syntax. An SPS-EQ consists of the following p.p.t algorithms:

$\mathsf{ParGen}(1^\lambda)$is a probabilistic algorithm which takes as input a security parameter λ and returns public parameters pp including an asymmetric bilinear group, but without the related trapdoor.

$\mathsf{TParGen}(1^\lambda)$is like the ParGen algorithm but it also returns the trapdoor.

$\mathsf{KGen}(\mathsf{pp}, \ell)$is a probabilistic algorithm which takes as input pp and a vector length $\ell > 1$, and outputs a key pair $(\mathsf{sk}, \mathsf{pk})$.

$\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, \mathbf{m})$is a probabilistic algorithm which takes as input pp, a representative $\mathbf{m} \in (\mathbb{G}_i^*)^\ell$ for class $[\mathbf{m}]_\mathcal{R}$, a secret key sk, and outputs a signature $\sigma' = (\sigma, \tau)$ (potentially including a tag τ) on the message $\mathbf{m}$.

$\mathsf{ChgRep}(\mathsf{pp}, \mathbf{m}, (\sigma, \tau), \mu, \mathsf{pk})$is a probabilistic algorithm which takes as input pp, a representative message $\mathbf{m} \in (\mathbb{G}_i^*)^\ell$, a signature σ (and potentially a tag τ), a scalar μ and a public key pk. It computes an updated signature σ' on new representative $\mathbf{m}^* = \mu\mathbf{m}$ and returns $(\mathbf{m}^*, \sigma')$.

$\mathsf{Verify}(\mathsf{pp}, \mathbf{m}, (\sigma, \tau), \mathsf{pk})$is a deterministic algorithm which takes as input pp, a representative message $\mathbf{m}$, a signature σ (potentially including a tag τ) and public key pk. If σ is a valid signature on $\mathbf{m}$ it outputs 1 and 0 otherwise.

Correctness. An SPS-EQ scheme over $(\mathbb{G}_i^*)^\ell$ is correct if for any $\lambda \in \mathbb{N}$, any $\ell > 1$, any $\mathsf{pp} \xleftarrow{\$} \mathsf{ParGen}(1^\lambda)$, any pair $(\mathsf{sk}, \mathsf{pk})$, any message $\mathbf{m} \in (\mathbb{G}_i^*)^\ell$, and any $\mu \in \mathbb{Z}_p^*$, the following holds:

$$\mathsf{Pr}\big[\mathsf{Verify}(\mathbf{m}, \mathsf{Sign}(\mathsf{sk}, \mathbf{m}), \mathsf{pk}) = 1\big] = 1, \text{ and}$$
$$\mathsf{Pr}\big[\mathsf{Verify}(\mathsf{ChgRep}(\mathbf{m}, \mathsf{Sign}(\mathsf{sk}, \mathbf{m}), \mu, \mathsf{pk}), \mathsf{pk}) = 1\big] = 1.$$

EUF-CMA. An SPS-EQ scheme over $(\mathbb{G}_i^*)^\ell$ is existentially unforgeable under adaptively chosen-message attacks, if for all $\ell > 1$ and p.p.t adversaries $\mathcal{A}$ with access to a signing oracle Sign, the following probability is negligible,

$$\mathsf{Pr}\left[\begin{array}{l}\mathsf{pp} \xleftarrow{\$} \mathsf{ParGen}(1^\lambda),\\ (\mathsf{sk}, \mathsf{pk}) \xleftarrow{\$} \mathsf{KGen}(\mathsf{pp}, \ell),\\ ([\mathbf{m}]_i^*, \sigma^*) \xleftarrow{\$} \mathcal{A}^{\textsc{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})\end{array} : \begin{array}{r}[\mathbf{m}^*]_\mathcal{R} \neq [\mathbf{m}]_\mathcal{R} \;\forall\; [\mathbf{m}]_i \in \mathsf{Q} \;\wedge\\ \mathsf{Verify}([\mathbf{m}]_i^*, \sigma^*, \mathsf{pk}) = 1\end{array}\right],$$

where Q is the set of queries that $\mathcal{A}$ has issued to the signing oracle Sign. Note that in the tag-based case this oracle returns (σ_i, τ_i).

The following notion is based on Definition 10 from [41], which defines perfect adaption of signatures in the CRS model. Perfect adaption mandates that signatures output by the algorithm ChgRep are distributed identically to new signatures on the respective representative. When this notion is defined considering adversaries who could maliciously generate signing keys, one obtains the strongest possible notion for perfect adaption. Unlike [41], we opt to explicitly state that perfect adaption is defined with respect to the message space. We do this, as later on we will introduce a new a definition for perfect adaption with respect to the *key space*.

Perfect adaption of signatures (under malicious keys in the honest parameters model) with respect to the *message space*: An SPS-EQ over $\mathcal{S}_\mathbf{m}$ perfectly adapts signatures with respect to the message space if for all tuples $(\mathsf{pp}, \mathsf{pk}, [\mathbf{m}]_i, \sigma, \mu)$ where $\mathsf{pp} \xleftarrow{\$} \mathsf{ParGen}(1^\lambda)$, $[\mathbf{m}]_i \in \mathcal{S}_\mathbf{m}$, $\mu \in \mathbb{Z}_p^*$, and $\mathsf{Verify}([\mathbf{m}]_i, \sigma, \mathsf{pk}) = 1$, we have that the output of $\mathsf{ChgRep}([\mathbf{m}]_i, (\sigma, \tau), \mu, \mathsf{pk})$ is $([\mu \cdot \mathbf{m}]_i, \sigma^*)$, with σ^* being a uniformly random element in the space of signatures, conditioned on $\mathsf{Verify}([\mu \cdot \mathbf{m}]_i, \sigma^*, \mathsf{pk}) = 1$.

5.1 Our Malleable NIZK argument

Our malleable NIZK argument is based solely on the fully-adaptive OR-Proof from [19]. This allows us to circumvent the randomization problem in the OR-Proof from [46], and to avoid the need to build a QA-NIZK atop.

As a result, we reduce the number of exponentiations required in the proving and ZKEval algorithms, which leads to a more efficient signature scheme. This comes at the cost of relying on the $\mathcal{L}_1$-1-$\mathsf{extKerMDH}$ assumption. We argue that the change is justified as the $\mathsf{extKerMDH}$ is a natural extension of the KerMDH assumption and in this case, the assumption is also falsifiable.

Intuition. We look for a NIZK proof which can be randomizable and malleable so that randomized proofs look like fresh proofs, while the malleability allows to update the proof statements. The goal is to obtain derivation privacy, which is crucial to perform the change of representative in the signature scheme.

The fully-adaptive NIZK argument from [19] is based on a challenge $z = z_0 + z_1$, where z is in the CRS, and z_0 and z_1 are elements of the proof and

$\underline{\mathsf{PGen}(1^\lambda)}$:
$\mathsf{BG} \xleftarrow{\$} \mathsf{BGGen}(1^\lambda)$; $z \xleftarrow{\$} \mathbb{Z}_p$
return $((\mathsf{BG}, [z]_2), z)$

$\underline{\mathsf{PPro}(\mathsf{crs}, [\mathbf{x}_1]_1, \mathbf{w}_1, [\mathbf{x}_2]_1, \mathbf{w}_2)}$:
// $[\mathbf{x}_j]_1 = \mathbf{A}_i\mathbf{w}_j$ with $\mathbf{A} \in \mathcal{M}^{2k\times k}$
$\mathbf{s}_j \xleftarrow{\$} \mathbb{Z}_p^k$; $z_{1-i} \xleftarrow{\$} \mathbb{Z}_p^*$; $\delta \xleftarrow{\$} \mathbb{Z}_p^*$
$[z_i]_2 \leftarrow \delta[z]_2 - [z_{1-i}]_2$
$[\mathbf{d}_i^j]_2 \leftarrow [z_i]_2\mathbf{w}_j + [\mathbf{s}_j]_2$
$[\mathbf{a}_i^j]_1 \leftarrow [\mathbf{A}_i]\mathbf{s}_j$
$\mathbf{d}_{1-i}^j \xleftarrow{\$} \mathbb{Z}_p^k$
$[\mathbf{a}_{1-i}^j]_1 \leftarrow \mathbf{A}_{1-i}\mathbf{d}_{1-i}^j - z_{1-i}\mathbf{x}_j$
return $([\mathbf{a}_i^j]_1, [\mathbf{d}_i^j]_2, [z_i]_2, \delta P_1)_{i\in\{0,1\}}^{j\in\{1,2\}}$

$\underline{\mathsf{PSim}(\mathsf{crs}, z, [\mathbf{x}_1]_1, [\mathbf{x}_2]_1)}$:
$z_0 \xleftarrow{\$} \mathbb{Z}_p$; $\delta \xleftarrow{\$} \mathbb{Z}_p^*$; $z_1 \leftarrow \delta z - z_0$
for all $i \in \{0,1\}$, $j \in \{1,2\}$ **do**
  $\mathbf{d}_i^j \xleftarrow{\$} \mathbb{Z}_p^k$; $[\mathbf{a}_i^j]_1 \leftarrow \mathbf{A}_i\mathbf{d}_i^j - z_i\mathbf{x}_j$
return $([\mathbf{a}_i^j]_1, [\mathbf{d}_i^j]_2, [z_i]_2, \delta P_1)_{i\in\{0,1\}}^{j\in\{1,2\}}$

$\underline{\mathsf{PRVer}(\mathsf{crs}, [\mathbf{x}]_1, \pi)}$:
parse $\pi = ([\mathbf{a}_i]_1, [\mathbf{d}_i]_2, [z_i]_2, Z_1)_{i\in\{0,1\}}$
check $e(Z_1, [z]_2) = e([1]_1, [z_0]_2 + [z_1]_2)$
for all $i \in \{0,1\}$ **check**
  $e([\mathbf{A}_i]_1, [\mathbf{d}_i]_2) = e([\mathbf{x}]_1, [z_i]_2) + e([\mathbf{a}_i]_1, [1]_2)$

$\underline{\mathsf{PVer}(\mathsf{crs}, [\mathbf{x}_1]_1, [\mathbf{x}_2]_1, \Omega)}$:
parse $\Omega = ([\mathbf{a}_i^j]_1, [\mathbf{d}_i^j]_2, [z_i]_2, Z_1)_{i\in\{0,1\}}^{j\in\{1,2\}}$
check $e(Z_1, [z]_2) = e([1]_1, [z_0]_2 + [z_1]_2)$
for all $i \in \{0,1\}$, $j \in \{1,2\}$ **check**
  $e([\mathbf{A}_i]_1, [\mathbf{d}_i^j]_2) = e([\mathbf{x}_j]_1, [z_i]_2) + e([\mathbf{a}_i^j]_1, [1]_2)$

$\underline{\mathsf{ZKEval}(\mathsf{crs}, [\mathbf{x}_1]_1, [\mathbf{x}_2]_1, \Omega)}$:
parse $\Omega = ([\mathbf{a}_i^j]_1, [\mathbf{d}_i^j]_2, [z_i]_2, Z_1)_{i\in\{0,1\}}^{j\in\{1,2\}}$
check $\mathsf{PVer}(\mathsf{crs}, [\mathbf{x}_1]_1, [\mathbf{x}_2]_1, \Omega)$
$\alpha, \beta \xleftarrow{\$} \mathbb{Z}_p^*$; $Z_1' \leftarrow \alpha Z_1$
for all $i \in \{0,1\}$
  $[z_i']_2 \leftarrow \alpha[z_i]_2$; $[\mathbf{a}_i']_1 \leftarrow \alpha[\mathbf{a}_i^1]_1 + \alpha\beta[\mathbf{a}_i^2]_1$
  $[\mathbf{d}_i']_2 \leftarrow \alpha[\mathbf{d}_i^1]_2 + \alpha\beta[\mathbf{d}_i^2]_2$
return $([\mathbf{a}_i']_1, [\mathbf{d}_i']_2, [z_i']_2, Z_1')$

Fig. 3: Malleable NIZK argument for language $\mathcal{L}^\vee_{\mathbf{A}_0,\mathbf{A}_1}$

chosen such that the equation holds. To randomize a proof we need to randomize z_0 and z_1 and so, instead of checking the original equation we will check for linear combinations of the equation $\alpha z = z_0 + z_1$. We modify the original proof to compute a random α and add an extra element $Z = \alpha P_1$ to the proof. Consequently, the verification algorithm will now check an extra pairing.

As observed in [41], the malleability of the OR-NIZK proof can be achieved by using a tag and a second NIZK for that tag with shared randomness. We follow the same approach. The resulting malleable NIZK argument for the OR-language (for fixed $\mathbf{A}_0$ and $\mathbf{A}_1$) is defined below and presented in Figure 3.

$$\mathcal{L}^\vee_{\mathbf{A}_0,\mathbf{A}_1} = \{[\mathbf{x}]_1 \in \mathbb{G}_1^{2k} | \exists\ \mathbf{w} \in \mathbb{Z}_p^k : [\mathbf{x}]_1 = [\mathbf{A}_0]_1 \cdot \mathbf{w} \vee [\mathbf{x}]_1 = [\mathbf{A}_1]_1 \cdot \mathbf{w}\},$$

Theorem 2. The protocol in Figure 3 is a fully adaptive NIZK argument for the OR-language $\mathcal{L}^\vee_{\mathbf{A}_0,\mathbf{A}_1}$ if the falsifiable $\mathcal{L}_1$-$(4k+1)$-extKerMDH assumption holds in $\mathbb{G}_2$.

Proof. The proof follows [19] and is provided in [18] (Appendix C).

5.2 Signature Construction

Our construction is shown in Figure 4, where the higlighted sections note the main differences to the scheme presented in [41]. In [18] (Appendix H), we also show how to extend it to obtain mercurial signatures (later explained in Section 7.1).

SPS-EQ.ParGen(1^λ):
$\mathsf{BG} \xleftarrow{\$} \mathsf{BGGen}(1^\lambda)$; $\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1 \xleftarrow{\$} \mathcal{D}_1$
$(\mathsf{crs}, \mathsf{td}) \xleftarrow{\$} \mathsf{PGen}(1^\lambda; \mathsf{BG})$
return $(\mathsf{BG}, [\mathbf{A}]_2, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1, \mathsf{crs})$

SPS-EQ.TParGen(1^λ):
$\mathsf{BG} \xleftarrow{\$} \mathsf{BGGen}(1^\lambda)$; $\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1 \xleftarrow{\$} \mathcal{D}_1$
$(\mathsf{crs}, \mathsf{td}) \xleftarrow{\$} \mathsf{PGen}(1^\lambda; \mathsf{BG})$
$\mathsf{pp} \leftarrow (\mathsf{BG}, [\mathbf{A}]_2, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1, \mathsf{crs})$
return $(\mathsf{pp}, \mathsf{td})$

SPS-EQ.KGen($\mathsf{pp}, 1^\lambda$):
$\mathbf{K}_0 \xleftarrow{\$} \mathbb{Z}_p^{2\times 2}$; $\mathbf{K} \xleftarrow{\$} \mathbb{Z}_p^{\ell\times 2}$
$[\mathbf{B}]_2 \leftarrow [\mathbf{K}_0]_2[\mathbf{A}]_2$; $[\mathbf{C}]_2 \leftarrow [\mathbf{K}]_2[\mathbf{A}]_2$
$\mathsf{sk} \leftarrow (\mathbf{K}_0, \mathbf{K})$; $\mathsf{pk} \leftarrow ([\mathbf{B}]_2, [\mathbf{C}]_2)$
return $(\mathsf{sk}, \mathsf{pk})$

SPS-EQ.Sign($\mathsf{pp}, \mathsf{sk}, [\mathbf{m}]_1$):
$r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p$
$[\mathbf{t}]_1 \leftarrow [\mathbf{A}_0]_1 r_1$; $[\mathbf{w}]_1 \leftarrow [\mathbf{A}_0]_1 r_2$
$\Omega \leftarrow \mathsf{PPro}(\mathsf{crs}, [\mathbf{t}]_1, r_1, [\mathbf{w}]_1, r_2)$
parse $\Omega = (\Omega_1, \Omega_2, [z_0]_2, [z_1]_2, Z_1)$
$\mathbf{u}_1 \leftarrow \mathbf{K}_0^\top [\mathbf{t}]_1 + \mathbf{K}^\top [\mathbf{m}]_1$; $\mathbf{u}_2 \leftarrow \mathbf{K}_0^\top [\mathbf{w}]_1$
$\sigma \leftarrow ([\mathbf{u}_1]_1, [\mathbf{t}]_1, \Omega_1, [z_0]_2, [z_1]_2, Z_1)$
$\tau \leftarrow ([\mathbf{u}_2]_1, [\mathbf{w}]_1, \Omega_2)$
return (σ, τ)

SPS-EQ.Verify($\mathsf{pp}, [\mathbf{m}]_1, (\sigma, \tau), \mathsf{pk}$):
parse $\sigma = ([\mathbf{u}_1]_1, [\mathbf{t}]_1, \Omega_1, [z_0]_2, [z_1]_2, Z_1)$
parse $\tau \in \{([\mathbf{u}_2]_1, [\mathbf{w}]_1, \Omega_2) \cup \bot\}$
check $\mathsf{PRVer}(\mathsf{crs}, [\mathbf{t}]_1, \Omega_1, [z_0]_2, [z_1]_2, Z_1)$
check $e([\mathbf{u}_1]_1^\top, [\mathbf{A}]_2) =$
$e([\mathbf{t}]_1^\top, [\mathbf{B}]_2) + e([\mathbf{m}]_1^\top, [\mathbf{C}]_2)$
if $\tau \neq \bot$ **check**
$\mathsf{PRVer}(\mathsf{crs}, [\mathbf{w}]_1, \Omega_2, [z_0]_2, [z_1]_2, Z_1)$
$e([\mathbf{u}_2]_1^\top, [\mathbf{A}_2]) = e([\mathbf{w}]_1^\top, [\mathbf{B}]_2)$

SPS-EQ.ChgRep($\mathsf{pp}, [\mathbf{m}]_1, \sigma, \tau, \mu, \mathsf{pk}$):
parse $\sigma = ([\mathbf{u}_1]_1, [\mathbf{t}]_1, \Omega_1, [z_0]_2, [z_1]_2, Z_1)$
parse $\tau \in \{([\mathbf{u}_2]_1, [\mathbf{w}]_1, \Omega_2) \cup \bot\}$
$\Omega \leftarrow (\Omega_1, \Omega_2, [z_0]_2, [z_1]_2, Z_1)$
check $\mathsf{PVer}(\mathsf{crs}, [\mathbf{t}]_1, [\mathbf{w}]_1, \Omega)$
check $e([\mathbf{u}_2]_1^\top, [\mathbf{A}]_2) \neq e([\mathbf{w}]_1^\top, [\mathbf{B}]_2)$
check $e([\mathbf{u}_1]_1^\top, [\mathbf{A}]_2) \neq$
$e([\mathbf{t}]_1^\top, [\mathbf{B}]_2) + e([\mathbf{m}]_1^\top, [\mathbf{C}]_2)$
$\alpha, \beta, \xleftarrow{\$} \mathbb{Z}_p^*$
$[\mathbf{u}_1']_1 \leftarrow \mu[\mathbf{u}_1]_1 + \beta[\mathbf{u}_2]_1$
$[\mathbf{t}']_1 \leftarrow \mu[\mathbf{t}]_1 + \beta[\mathbf{w}]_1 = [\mathbf{A}_0]_1(\mu r_1 + \beta r_2)$
for all $i \in \{0, 1\}$
$[z_i']_2 \leftarrow \alpha[z_i]_2$
$[\mathbf{a}_i']_2 \leftarrow \alpha\mu[\mathbf{a}_i^1]_2 + \alpha\beta[\mathbf{a}_i^2]_2$
$[d_i']_1 \leftarrow \alpha\mu[d_i^1]_1 + \alpha\beta[d_i^2]_1$
$\Omega' \leftarrow ((([\mathbf{a}_i']_1, [d_i']_2, [z_i']_2)_{i\in\{0,1\}}, \alpha Z_1)$
$\sigma' \leftarrow ([\mathbf{u}_1']_1, [\mathbf{t}']_1, \Omega')$
return $(\mu[\mathbf{m}]_1, \sigma')$

Fig. 4: Our SPS-EQ scheme.

Theorem 3. The SPS-EQ in Figure 4 perfectly adapts signatures (under malicious keys in the honest parameter model) with respect to the message space.

To prove Theorem 3 we follow almost verbatim the original proof from [41].

Proof. For all $[\mathbf{m}]_1$ and $\mathsf{pk} = ([\mathbf{K}_0\mathbf{A}]_2, [\mathbf{K}\mathbf{A}]_2)$, a signature σ=($[\mathbf{u}_1]_1$, $[\mathbf{t}]_1$, Ω_1 ,$[z_0]_2$, $[z_1]_2$, Z_1) generated according to the CRS $([\mathbf{A}]_2, [\mathbf{A}_0]_1, [\mathbf{A}_1]_1, [z]_2)$ satisfying the verification algorithm must be of the form: σ=($\mathbf{K}_0^\top[\mathbf{A}_0]_1 r_1$+$\mathbf{K}^\top[\mathbf{m}]_1$, $[\mathbf{A}_0]_1 r_1$, $[\mathbf{A}_0]s_1$, $[\mathbf{A}_1]d_1^1 - z_1[\mathbf{A}_0]_1 r_1$, $[z_0]_2 r_1$+$[s_1]_2$, $[d_1^1]_2$, $[z_0]_2$, $[z_1]_2$, Z_1). A signature output by ChgRep has the form σ=($\mathbf{K}_0^\top[\mathbf{A}_0]_1(\mu r_1 + \beta r_2)$+$\mathbf{K}^\top[\mu\mathbf{m}]_1$, $[\mathbf{A}_0]_1(\mu r_1 + \beta r_2)$, $[\mathbf{A}_0]\alpha(\mu s_1 + \beta s_2)$, $[\mathbf{A}_1]\alpha(\mu d_1^1 + \beta d_1^2) - z_1[\mathbf{A}_0]_1\alpha(\mu r_1 + \beta r_2)$, $\alpha([z_0]_2(\mu r_1 + \beta r_2)$+$\mu[s_1]_2$+$\beta[s_2]_2)$, $\alpha(\mu[d_1^1]_2 + \beta[d_1^2]_2)$, $\alpha[z_0]_2$, $\alpha[z_1]_2$, αZ_1), for new independent randomness α, β and μ so is a random element in the space of all signatures. Furthermore, the signature output by ChgRep is distributed identically to a fresh signature on message $[\mathbf{m}]_1$ output by Sign. □

Theorem 4. If the KerMDH and MDDH assumptions hold, the SPS-EQ in Figure 4 is unforgeable.

Proof. The proof is provided in [18] (Appendix D).

6 Extending the ABC Model from [33]

In this section we present a new ABC model which extends [33] to consider NAND showing proofs and the use of a CRS (denoted as pp). A NAND showing proof allows users to demonstrate that a given set of attributes is not present in their credential. The core differences in this extended ABC model follow naturally from (1) the addition of disjoint sets in the SCDS scheme in section 4, and (2) the removal of the key verification algorithm (as we work with a CRS).

ABC Syntax. An ABC scheme consists of the following p.p.t algorithms:

$\mathsf{Setup}(1^\lambda, 1^q)$ takes a security parameter λ and an upper bound q for the size of attribute sets, and outputs public parameters pp discarding any trapdoor.

$\mathsf{TSetup}(1^\lambda, 1^q)$ similar to Setup but it also returns a trapdoor (if any).

$\mathsf{OrgKeyGen}(\mathsf{pp})$ takes pp as input and outputs an organization key pair $(\mathsf{osk}, \mathsf{opk})$.

$\mathsf{UserKeyGen}(\mathsf{pp})$ takes pp as input and outputs a user key pair $(\mathsf{usk}, \mathsf{upk})$.

$\mathsf{Obtain}(\mathsf{pp}, \mathsf{usk}, \mathsf{opk}, \mathcal{X})$ and $\mathsf{Issue}(\mathsf{pp}, \mathsf{upk}, \mathsf{osk}, \mathcal{X})$ are run by a user and the organization respectively, who interact during execution. Obtain takes as input pp, the user's secret key usk, an organization's public key opk, and an attribute set $\mathcal{X}$ of size $|\mathcal{X}| < t$. Issue takes as input pp, a user public key upk, the organization's secret key osk, and an attribute set $\mathcal{X}$ of size $|\mathcal{X}| < t$. At the end of this protocol, Obtain outputs a credential cred on $\mathcal{X}$ for the user or $\perp$ if the execution failed.

$\mathsf{Show}(\mathsf{pp}, \mathsf{opk}, \mathcal{X}, \mathcal{S}, \mathcal{D}, \mathsf{cred})$ and $\mathsf{Verify}(\mathsf{pp}, \mathsf{opk}, \mathcal{S}, \mathcal{D})$ are run by a user and a verifier respectively, who interact during execution. Show takes as input pp, an organization public key opk, a credential cred for the attribute set $\mathcal{X}$, potentially non-empty sets $\mathcal{S} \subseteq \mathcal{X}$, $\mathcal{D} \nsubseteq \mathcal{X}$ representing attributes sets being a subset ($\mathcal{S}$) or disjoint ($\mathcal{D}$) to the attribute set ($\mathcal{X}$) committed in the credential. Verify takes as input pp, an organization public key opk, the sets $\mathcal{S}$ and $\mathcal{D}$. At the end, Verify outputs 1 or 0 indicating whether or not the credential showing was accepted.

6.1 Security Properties

The following notions are based on the security model from [33] (Section 5.1), which we adapt to consider the use of a crs (pp) and NAND showing proofs. Informally, an ABC scheme is secure if it has the following properties:

Correctness. A showing of a credential with respect to a non-empty sets $\mathcal{S}$ and $\mathcal{D}$ of attributes always verify if the credential was issued honestly on some attribute set $\mathcal{X}$ with $\mathcal{S} \subset \mathcal{X}$ and $\mathcal{D} \nsubseteq \mathcal{X}$.

Unforgeablility. Given at least one non-empty set $\mathcal{S} \subset \mathcal{X}$ or $\mathcal{D} \not\subseteq \mathcal{X}$, a user in possession of a credential for the attribute set $\mathcal{X}$ cannot perform a valid showing for $\mathcal{D} \subset \mathcal{X}$ nor for $\mathcal{S} \not\subseteq \mathcal{X}$. 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 (thus, covering 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.

To introduce the corresponding formal definitions, the following global variables and oracles are listed below.

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 which issuance indices and corresponding users have been set during the first call to the left-or-right oracle in the anonymity game.

Oracles. Considering an adversary $\mathcal{A}$ 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 $(\mathtt{USK}[i], \mathtt{UPK}[i]) \xleftarrow{\$} \mathsf{UsrKGen}(\mathsf{opk})$, adding i to the honest user list $\mathtt{HU}$ 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, 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 $i \in I_{\mathsf{LoR}}$ (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 $\mathtt{UPK}[i] \leftarrow \mathsf{upk}$.

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

$$(\mathsf{cred}, \top) \xleftarrow{\$} \mathsf{Obtain}(\mathsf{pp}, \mathtt{USK}[i], \mathsf{opk}, \mathcal{X}), \mathsf{Issue}(\mathsf{pp}, \mathtt{UPK}[i], \mathsf{osk}, \mathcal{X}).$$

If $\mathsf{cred} = \perp$, it returns $\perp$. Else, it appends $(i, \mathsf{cred}, \mathcal{X})$ to $(\mathtt{OWNR}, \mathtt{CRED}, \mathtt{ATTR})$ and returns $\top$.

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

$$(\mathsf{cred}, \cdot) \xleftarrow{\$} \mathsf{Obtain}(\mathsf{pp}, \mathtt{USK}[i], \mathsf{opk}, \mathcal{X}), \cdot),$$

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

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

$$(\cdot, I) \xleftarrow{\$} (\cdot, \mathsf{Issue}(\mathsf{pp}, \mathtt{UPK}[i], \mathsf{osk}, \mathcal{X})),$$

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

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

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

where the Verify part is executed by $\mathcal{A}$.

$\mathcal{O}_{\mathsf{LoR}}(j_0, j_1, \mathcal{S}, \mathcal{D})$ is the challenge oracle in the anonymity game where $\mathcal{A}$ 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 attribute sets $\mathcal{S}$ and $\mathcal{D}$. If $J_{\mathsf{LoR}} \neq \emptyset$ and $J_{\mathsf{LoR}} \neq \{j_0, j_1\}$, it returns $\bot$. Let $i_0 \xleftarrow{\$} \mathtt{OWNR}[j_0]$ and $i_1 \xleftarrow{\$} \mathtt{OWNR}[j_1]$. If $J_{\mathsf{LoR}} \neq \emptyset$ then it sets $J_{\mathsf{LoR}} \xleftarrow{\$} \{j_0, j_1\}$ and $I_{\mathsf{LoR}} \xleftarrow{\$} \{i_0, i_1\}$. If $i_0, i_1 \neq \mathtt{HU} \vee \mathcal{S} \nsubseteq \mathtt{ATTR}[j_0] \cap \mathtt{ATTR}[j_1] \vee \mathcal{D} \cap \{\mathtt{ATTR}[j_0] \cup \mathtt{ATTR}[j_1]\} \neq \emptyset$, it returns $\bot$. Else, it runs

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

(with b set by the experiment) where the Verify part is executed by $\mathcal{A}$.

Correctness. An ABC system is correct, if for all $\lambda > 0$, all $t > 0$, all $\mathcal{X}$ with $0 < |\mathcal{X}| \leq t$ and all $\emptyset \neq \mathcal{S} \subset \mathcal{X}$ and $\emptyset \neq \mathcal{D} \nsubseteq \mathcal{X}$ with $0 < |\mathcal{D}| \leq t$ it holds that:

$$\mathsf{Pr}\left[\begin{array}{l} \mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, 1^q); \\ (\mathsf{osk}, \mathsf{opk}) \xleftarrow{\$} \mathsf{OrgKGen}(\mathsf{pp}); \\ (\mathsf{usk}, \mathsf{upk}) \xleftarrow{\$} \mathsf{UsrKGen}(\mathsf{pp}); \\ (\mathsf{cred}, \top) \xleftarrow{\$} (\mathsf{Obtain}(\mathsf{pp}, \mathsf{usk}, \mathsf{opk}, \mathcal{X}), \\ \qquad\qquad \mathsf{Issue}(\mathsf{pp}, \mathsf{upk}, \mathsf{osk}, \mathcal{X})) \end{array} : \begin{array}{l} (\top, 1) \xleftarrow{\$} (\mathsf{Show}(\mathsf{pp}, \mathsf{opk}, \mathcal{X}, \mathcal{S}, \\ \mathcal{D}, \mathsf{cred}), \mathsf{Verify}(\mathsf{pp}, \mathsf{opk}, \mathcal{S}, \mathcal{D})) \end{array}\right] = 1.$$

Unforgeability. An ABC system is unforgeable, if for all $\lambda > 0$, all $q > 0$ and p.p.t adversaries $\mathcal{A}$ having oracle access to $\mathcal{O} := \{\mathcal{O}_{\mathtt{HU}}, \mathcal{O}_{\mathtt{CU}}, \mathcal{O}_{\mathsf{ObtIss}}, \mathcal{O}_{\mathsf{Issue}}, \mathcal{O}_{\mathsf{Show}}\}$ the following probability is negligible.

$$\mathsf{Pr}\left[\begin{array}{l} \mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, 1^q); \\ (\mathsf{osk}, \mathsf{opk}) \xleftarrow{\$} \mathsf{OrgKGen}(\mathsf{pp}); \\ (\mathcal{S}, \mathcal{D}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{opk}); \\ (\cdot, b^*) \xleftarrow{\$} (\mathcal{A}(\mathsf{st}), \mathsf{Verify}(\mathsf{pp}, \mathsf{opk}, \mathcal{S}, \mathcal{D})) \end{array} : \begin{array}{l} b^* = 1 \wedge \\ \forall\, j : \mathtt{OWNR}[j] \in \mathtt{CU} \implies \\ (\mathcal{S} \notin \mathtt{ATTR}[j] \vee \mathcal{D} \in \mathtt{ATTR}[j]) \end{array}\right]$$

Anonymity. An ABC system is anonymous, if for all $\lambda > 0$, all $q > 0$ and all p.p.t adversaries $\mathcal{A}$ having oracle access to $\mathcal{O} := \{\mathcal{O}_{\mathtt{HU}}, \mathcal{O}_{\mathtt{CU}}, \mathcal{O}_{\mathsf{Obtain}}, \mathcal{O}_{\mathsf{Issue}}, \mathcal{O}_{\mathsf{Show}}, \mathcal{O}_{\mathsf{LoR}}\}$ the following probability is negligible.

$$\mathsf{Pr}\left[\begin{array}{l} \mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, 1^q); b \xleftarrow{\$} \{0,1\}; (\mathsf{opk}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}(\mathsf{pp}); \\ b^* \xleftarrow{\$} \mathcal{A}^{\mathcal{O}}(\mathsf{st}) \end{array} : b^* = b\right] - \tfrac{1}{2}$$

7 Our ABC construction

As previously explained in Section 1.3, our ABC scheme is based on the one from [33]. The main changes are the following:

- As we use a signature scheme that relies on a CRS, we move the parameters of the set-commitment scheme from the organization's key pair to the public parameters pp that include the previous CRS. Furthermore, we instantiate the ZKPoK's using Pedersen commitments and the construction from [22], as suggested in [33] (Remark 1).
- Our showing protocol can be instantiated with two sets $\mathcal{S}$ and $\mathcal{D}$, one to compute AND proofs (selective disclosure) and one to compute NAND proofs.
- We integrate the proof of exponentiation to the showing protocol [5].

Intuition. We begin explaining the difference to [33] with respect to malicious organizations as it clarifies the changes introduced in the issuing protocol. We recall that in this context the term *malicious organizations* refers to organizations whose key-pairs are generated in a way that trapdoor information is included. Such trapdoor information could later be used by an organization to break anonymity, provided that extra information (a transcript of a given showing protocol containing a credential issued by the organization) is available. The ABC scheme from [33] defines a ZKPoK in the issuing protocol ($\Pi^{\mathcal{R}_O}$) for which the organization needs to prove knowledge of the corresponding secret key to avoid the previous scenario. Since the signing keys in our SPS-EQ need to be generated using the CRS (which includes the matrix $\mathbf{A}$), we do not need to request a ZKPoK from the organization in the issuing protocol as the signature's verification algorithm a pairing involving the matrix $\mathbf{A}$ and the organization's public key $\mathsf{opk} = (\mathbf{B}, \mathbf{C})$ is used to check the signature. Hence, a signature that verifies rules out that 1) someone impersonated the issuer signing with a different secret key, and 2) that the public key was maliciously generated. Regarding the showing protocol, the only changes are the addition of NAND and exponentiation proofs. For the latter, we require the verifier to randomly pick the challenge and send it to the user.

For ease of exposition, we present the resulting construction (Scheme 1) in Figure 5 considering selective disclosures only. We highlight in gray the required changes to do NAND proofs, but both types of proofs could be computed while executing a single showing. If so, a NAND proof increases bandwidth by 4 elements (two from $\mathbb{G}_1$ and two from $\mathbb{G}_2$), as the PoE can reuse the same challenge.

Theorem 5. Scheme 1 is correct.

Theorem 6. If the q-co-DL assumption holds, the ZKPoK's have perfect ZK, SCDS is sound, and $\mathsf{SPS\text{-}EQ}$ is EUF-CMA secure, then Scheme 1 is unforgeable.

Theorem 7. If the DDH assumption holds, the ZKPoK's have perfect ZK, and the $\mathsf{SPS\text{-}EQ}$ perfectly adapts signatures, then Scheme 1 is anonymous.

[5] The security of this integration is discussed in [18] (Appendix J).

Proof. Proof of Theorem 6 follows closely to that presented in [33] but extended to include disjoint sets. Proof of Theorem 7 also follows that in [33] with the exception that we work with a CRS and an accordingly modified definition of perfect adaption. All proofs are provided in [18] (Appendix E).

7.1 Revocation strategies

The natural approach to revocation would be to follow that described in [24] where they use the fact that randomization of a credential is compatible with the randomization of the accumulator and its corresponding witness. This approach requires the revocation authority to compute and maintain the witness list. As it uses the accumulator from [1], the cost of non-membership proofs is linear in the size of the accumulator (*i.e.,* revoked users), and this should be done at least once by the manager for every user. If, instead, the dynamic variant is used (as discussed in [24]), then users could be given their non-membership witness once and subsequently update it with a single constant size operation. Other approaches for revocation are discussed in [18] (Appendix F).

7.2 Signer-Hiding

We recall that our signature scheme is based on the one from [41] and that we are using the credential framework of [33]. Therefore, as we have $k = 1$ and $\ell = 3$, the public keys consist of two vectors $[\mathbf{B}]_2 \in (\mathbb{G}_2^*)^2$ and $[\mathbf{C}]_2 \in (\mathbb{G}_2^*)^3$, where the secret keys have the form $\mathsf{sk} = (\mathbf{K}_0, \mathbf{K})$ with $\mathbf{K}_0 \xleftarrow{\$} (\mathbb{Z}_p^*)^{2\times 2}$ and $\mathbf{K} \xleftarrow{\$} (\mathbb{Z}_p^*)^{3\times 2}$. With this in mind, we can naturally define equivalence relationships on the key spaces $\mathcal{S}_{\mathsf{sk}} = \{(\mathbb{Z}_p^*)^{2\times 2} \times (\mathbb{Z}_p^*)^{3\times 2}\}$ and $\mathcal{S}_{\mathsf{pk}} = \{(\mathbb{G}_2^*)^2 \times (\mathbb{G}_2^*)^3\}$ as follows:

$$\mathcal{R}_{\mathsf{sk}} = \{(\mathsf{sk}, \tilde{\mathsf{sk}}) \in \mathcal{S}_{\mathsf{sk}} \times \mathcal{S}_{\mathsf{sk}} \mid \exists\ \rho \in \mathbb{Z}_p^* \text{ s.t } \tilde{\mathsf{sk}} = \rho \cdot \mathsf{sk}\}$$
$$\mathcal{R}_{\mathsf{pk}} = \{(\mathsf{pk}, \tilde{\mathsf{pk}}) \in \mathcal{S}_{\mathsf{pk}} \times \mathcal{S}_{\mathsf{pk}} \mid \exists\ \rho \in \mathbb{Z}_p^* \text{ s.t } \tilde{\mathsf{pk}} = \rho \cdot \mathsf{pk}\}$$

If we have a list of public keys $(\mathbf{B}_1, \mathbf{C}_1), ..., (\mathbf{B}_n, \mathbf{C}_n)$ and define the equivalence class of each public key as before $((\mathbf{B}'_i, \mathbf{C}'_i) = (\mathbf{B}_i, \mathbf{C}_i) \cdot \rho)$, we can efficiently prove that a given public key $(\mathbf{B}'_i, \mathbf{C}'_i)$ belongs to the equivalence class of one of the public keys $(\mathbf{B}_1, \mathbf{C}_1), ..., (\mathbf{B}_n, \mathbf{C}_n)$ for some $(\mathbf{B}_i, \mathbf{C}_i)$. The idea is to use a generalized version of the OR-Proof from [19], and building a generalized NIZK OR-Proof for the AND statements of the two components. The new language is defined as follows (remember we use $\ell = 3$):

$$\mathcal{L}_{\bigvee(\mathbf{B}_i \wedge \mathbf{C}_i)_{i\in[n]}} = \{(\mathbf{B}'_i, \mathbf{C}'_i) \in \mathbb{G}_2^{2\times\ell} \mid \exists\ \rho \in \mathbb{Z}_p^* : \vee\ (\mathbf{B}'_i = \mathbf{B}_i \cdot \rho \wedge \mathbf{C}'_i = \mathbf{C}_i \cdot \rho)_{i\in[n]}\}$$

The resulting NIZK argument is given in Figure 6.

Theorem 8. The proof system given in Figure 6 is a fully-adaptive NIZK argument for the language $\mathcal{L}_{\bigvee(\mathbf{B}_i \wedge \mathbf{C}_i)_{i\in[n]}}$.

Proof. The proof follows from Theorem 19 in [19]. The only difference is that we rely on the AND composition for sigma protocols to compile the one in [19] using the same challenge for both proofs.

ABC.Setup$(1^\lambda, 1^q)$:
$(\mathsf{BG}, \mathsf{scds}_{\mathsf{pp}}) \xleftarrow{\$} \mathsf{SCDS.Setup}(1^\lambda, q)$; $(\mathsf{sps}_{\mathsf{pp}}) \xleftarrow{\$} \mathsf{SPS\text{-}EQ.ParGen}(1^\lambda; \mathsf{BG})$;
$r \xleftarrow{\$} \mathbb{Z}_p^*$; $\mathsf{ck} \leftarrow (P_1, rP_1)$; **return** $(\mathsf{BG}, \mathsf{scds}_{\mathsf{pp}}, \mathsf{sps}_{\mathsf{pp}}, \mathsf{ck})$

ABC.TSetup$(1^\lambda, 1^q)$:
$(\mathsf{BG}, \mathsf{scds}_{\mathsf{pp}}, \mathsf{scds}_{\mathsf{td}}) \xleftarrow{\$} \mathsf{SCDS.TSetup}(1^\lambda, q)$; $(\mathsf{sps}_{\mathsf{pp}}, \mathsf{sps}_{\mathsf{td}}) \xleftarrow{\$} \mathsf{SPS\text{-}EQ.TParGen}(1^\lambda; \mathsf{BG})$;
$r \xleftarrow{\$} \mathbb{Z}_p^*$; $\mathsf{ck} \leftarrow (P_1, rP_1)$; $\mathsf{ck}_{\mathsf{td}} \leftarrow r$; **return** $((\mathsf{BG}, \mathsf{scds}_{\mathsf{pp}}, \mathsf{sps}_{\mathsf{pp}}, \mathsf{ck}), (\mathsf{scds}_{\mathsf{td}}, \mathsf{sps}_{\mathsf{td}}, \mathsf{ck}_{\mathsf{td}}))$

ABC.OrgKGen(pp):
return $\mathsf{SPS\text{-}EQ.KGen}(\mathsf{BG}, \mathsf{sps}_{\mathsf{pp}}, 3)$

ABC.UsrKGen(pp):
$\mathsf{usk} \xleftarrow{\$} \mathbb{Z}_p^*$; $\mathsf{upk} \leftarrow \mathsf{usk}P_1$
return $(\mathsf{usk}, \mathsf{upk})$

ABC.Obtain(pp, usk, opk, $\mathcal{X}$)		ABC.Issue(pp, upk, osk, $\mathcal{X}$)
$r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p^*$; $a \leftarrow r_1 P_1$		
$c \leftarrow \mathsf{Commit}(\mathsf{ck}, a, r_2)$	$\xrightarrow{c}$	
$z \leftarrow r_1 + e \cdot \mathsf{usk}$	$\xleftarrow{e}$	$e \xleftarrow{\$} \mathbb{Z}_p^*$
$(C, O) \leftarrow \mathsf{SCDS.Commit}(\mathsf{scds}_{\mathsf{pp}}, \mathcal{X}; \mathsf{usk})$	$\xrightarrow{C,\ R,\ z,a,r_2}$	**if** $(zP_1 \neq a + e \cdot \mathsf{upk} \vee$
$r_3 \xleftarrow{\$} \mathbb{Z}_p^*$; $R \leftarrow r_3 C$		$c \neq \mathsf{Commit}(\mathsf{ck}, a, r_2))$ **return** $\perp$
		if $(e(C, P_2) \neq e(\mathsf{upk}, \mathsf{Ch}_{\mathcal{X}}(s) P_2)$
		$\wedge \forall\, x \in \mathcal{X} : xP_1 \neq \mathsf{ek}_1^0)$ **return** $\perp$
	$\xleftarrow{(\sigma,\tau)}$	$(\sigma, \tau) \leftarrow \mathsf{SPS\text{-}EQ.Sign}((C, R, P_1), \mathsf{osk})$
check $\mathsf{SPS\text{-}EQ.Verify}(\mathsf{sps}_{\mathsf{pp}}, (C, R, P_1), (\sigma, \tau), \mathsf{opk})$		
return $\mathsf{cred} = (C, (\sigma, \tau), r_3, O)$		

ABC.Show(pp, usk, opk, ek, $\mathcal{S}$, cred)		ABC.Verify(pp, opk, $\mathcal{S}$)
parse $\mathsf{cred} = (C, \sigma, r, O)$; $\mu \xleftarrow{\$} \mathbb{Z}_p^*$		
if $O = (1, (o_1, o_2))$ **then**		
$O' = (1, (\mu \cdot o_1, o_2))$ **else** $O' = \mu O$		
$\sigma' \xleftarrow{\$} \mathsf{SPS\text{-}EQ.ChgRep}(\mathsf{sps}_{\mathsf{pp}}, (C, rC, P_1), \sigma, \tau, \mu, \mathsf{opk})$		
$(C_1, C_2, C_3) \leftarrow \mu \cdot (C, rC, P_1)$		
$\mathsf{cred}' \leftarrow (C_1, C_2, C_3, \sigma')$		
$\mathsf{wit} \leftarrow \mathsf{SCDS.OpenSS}(\mathsf{scds}_{\mathsf{pp}}, \mu C, \mathcal{S}, \mathsf{ek}, O')$		
$r_1, r_2, r_3, r_4 \xleftarrow{\$} \mathbb{Z}_p^*$; $a_1 \leftarrow r_1 C_1$; $a_2 \leftarrow r_3 P_1$		
$c_1 \leftarrow \mathsf{Commit}(\mathsf{ck}, a_1, r_2)$	$\xrightarrow{\mathsf{cred}',\ \mathsf{wit},\ c_1,c_2}$	**parse** $\mathsf{cred}' = (C_1, C_2, C_3, \sigma)$
$c_2 \leftarrow \mathsf{Commit}(\mathsf{ck}, a_2, r_4)$		
$\pi \leftarrow \mathsf{SCDS.PoE}(\mathsf{scds}_{\mathsf{pp}}, \mathcal{S}, \tilde{e})$	$\xleftarrow{e,\tilde{e}}$	$e, \tilde{e} \xleftarrow{\$} \mathbb{Z}_p^*$
$z_1 \leftarrow r_1 + e \cdot (r \cdot \mu)$; $z_2 \leftarrow r_3 + e \cdot \mu$		
$\Omega = ((z_i, a_i, r_i)_{i \in \{1,2\}}, \pi)$	$\xrightarrow{\Omega}$	**parse** $\Omega = ((z_i, a_i, r_i)_{i \in \{1,2\}}, \pi)$
		check
		$z_1 C_1 = a_1 + eC_2$; $z_2 P_1 = a_2 + eC_3$
		$c_1 = \mathsf{Commit}(\mathsf{ck}, a_1, r_2)$
		$c_2 = \mathsf{Commit}(\mathsf{ck}, a_2, r_4)$
		$\mathsf{SPS\text{-}EQ.Verify}(\mathsf{sps}_{\mathsf{pp}}, \mathsf{cred}', \mathsf{opk})$
		$\mathsf{SCDS.VerifySS}(\mathsf{scds}_{\mathsf{pp}}, C_1, \mathcal{S}, \mathsf{wit}; \pi, \tilde{e})$

Fig. 5: Scheme 1.

$\underline{\mathsf{PGen}(1^\lambda)}$:
$\mathsf{BG} \xleftarrow{\$} \mathsf{BGGen}(1^\lambda)$; $z \xleftarrow{\$} \mathbb{Z}_p$; $\mathsf{td} \leftarrow z$
return $(\mathsf{BG}, \mathsf{crs}, [z]_1)$

$\underline{\mathsf{PPrv}(\mathsf{crs}, (\mathbf{B}_i, \mathbf{C}_i)_{i\in[n]}, (\mathbf{B}'_i, \mathbf{C}'_i), \rho)}$:
// $\mathbf{B}'_i = \mathbf{B}_i \cdot \rho \wedge \mathbf{C}'_i = \mathbf{C}_i \cdot \rho$
$s_1, s_2, z_1, ..., z_{n-1} \xleftarrow{\$} \mathbb{Z}_p$
$[z_n]_1 \leftarrow [z]_1 - \sum_{j=1}^{j=n-1} [z_j]_1$
$[\mathbf{a}_i^1]_2 \leftarrow s_1\mathbf{B}_i$; $[\mathbf{a}_i^2]_2 \leftarrow s_2\mathbf{C}_i$
$[d_i^1]_1 \leftarrow \rho[z_i]_1 + [s_1]_1$; $[d_i^2]_1 \leftarrow \rho[z_i]_1 + [s_2]_1$
for all $j \neq i \in [n]$ **do**
  $d_j^1, d_j^2 \xleftarrow{\$} \mathbb{Z}_p$
  $[\mathbf{a}_j^1]_2 \leftarrow d_j^1\mathbf{B}_j - z_j\mathbf{B}'_i$; $[\mathbf{a}_j^2]_2 \leftarrow d_j^2\mathbf{C}_j - z_j\mathbf{C}'_i$
return $((([\mathbf{a}_n^k]_2, [d_n^k]_1)_{n\in[n]}^{k\in[2]}, ([z_j]_1)_{j\in[n-1]})$

$\underline{\mathsf{PSim}(\mathsf{crs}, \mathsf{td}, (\mathbf{B}_i, \mathbf{C}_i)_{i\in[n]}, (\mathbf{B}'_i, \mathbf{C}'_i))}$:
$z_1, ..., z_{n-1} \xleftarrow{\$} \mathbb{Z}_p$
$[z_n]_1 \leftarrow [\mathsf{td}]_1 - \sum_{j=1}^{j=n-1} [z_j]_1$
for all $\mathbf{i} \in [n]$ **do**
  $d_i^1, d_i^2 \xleftarrow{\$} \mathbb{Z}_p$
  $[\mathbf{a}_i^1]_2 \leftarrow d_i^1\mathbf{B}_i - z_i\mathbf{B}'_i$; $[\mathbf{a}_i^2]_2 \leftarrow d_i^2\mathbf{C}_i - z_i\mathbf{C}'_i$
return $((([\mathbf{a}_n^k]_2, [d_n^k]_1)_{n\in[n]}^{k\in[2]}, ([z_j]_1)_{j\in[n-1]})$

$\underline{\mathsf{PVer}(\mathsf{crs}, (\mathbf{B}_i, \mathbf{C}_i)_{i\in[n]}, (\mathbf{B}'_i, \mathbf{C}'_i), \pi)}$:
parse $\pi = ((([\mathbf{a}_n^k]_2, [d_n^k]_1)_{n\in[n]}^{k\in[2]}, ([z_j]_1)_{j\in[n-1]})$
$[z_n]_1 = [z]_1 - \sum_{j=1}^{j=n-1} [z_j]_1$
for all $\mathbf{i} \in [n]$ **check**
  $e([d_i^1]_1, \mathbf{B}_i) = e([z_i]_1, \mathbf{B}'_i) + e([1]_1, [\mathbf{a}_i^1]_2)$
  $e([d_i^2]_1, \mathbf{C}_i) = e([z_i]_1, \mathbf{C}'_i) + e([1]_1, [\mathbf{a}_i^2]_2)$

Fig. 6: Fully adaptive NIZK argument for $\mathcal{L}^{\vee}_{(\mathbf{B}_i \wedge \mathbf{C}_i)_{i\in[n]}}$

We now explain how the above NIZK can be used to hide the identity of a signer. First, we need to consider a scenario in which n-authorities can issue credentials to different sets of users. As we are in the classical setting, we also assume that every user gets a credential from one of the n-authorities and that the organization keys are certified and publicly available.

When showing a credential, the verifier needs to check the signature using the corresponding public key. The idea is to use the above NIZK proof so that a user can randomize the public key and present this randomized key to the verifier, which in turn will check the NIZK to verify that the public key is valid (*i.e.,* it belongs to the equivalence class of one of the n-authorities).

Signatures need to be adapted by the users so that they can be verified with the randomized public key. Therefore, we consider the definition of mercurial signatures [20], which includes algorithms ConvertPK, ConvertSK and ConvertSig, and introduce the following notion.

Perfect adaption of signatures (under malicious keys in the honest parameters model) with respect to the *key space*; An SPS-EQ over a message space $\mathcal{S}_{\mathbf{m}}$ perfectly adapts signatures with respect to the key space $\mathcal{S}_{\mathsf{pk}}$ if for all tuples $(\mathsf{pp}, [\mathsf{pk}]_j, [\mathbf{m}]_i, (\sigma, \tau), \rho)$ where $\mathsf{pp} \xleftarrow{\$} \mathsf{ParGen}(1^\lambda)$, $[\mathsf{pk}]_j \in \mathcal{S}_{\mathsf{pk}}$, $[\mathbf{m}]_i \in \mathcal{S}_{\mathbf{m}}$, $\mathsf{Verify}([\mathbf{m}]_i, (\sigma, \tau), [\mathsf{pk}]_j) = 1$ and $\rho \in \mathbb{Z}_p^*$, we have that the output of $\mathsf{ConvertSig}([\mathbf{m}]_i, (\sigma, \tau), \rho, [\mathsf{pk}]_j)$ is σ^*, with σ^* being a random element in the space of signatures, conditioned on $\mathsf{Verify}([\mathbf{m}]_i, \sigma^*, \mathsf{ConvertPK}([\mathsf{pk}]_j, \rho)) = 1$.

ConvertSig is analogous to the ChgRep algorithm, but restricted to act on the equivalence class defined by the key space. The algorithms ConvertPK and ConvertSK are just defined to abstract the computation of new representatives.

As our signature construction is compatible with the joint executions of the algorithms ChgRep and ConvertSig, we define below a general notion for perfect adaption where the ChgRep algorithm acts on all the equivalence classes.

Perfect adaption of signatures (under malicious keys in the honest parameters model); An SPS-EQ over $\mathcal{S}_{\mathbf{m}}$ perfectly adapts signatures with respect to the key space $\mathcal{S}_{\mathsf{pk}}$ if for all tuples $(\mathsf{pp}, [\mathsf{pk}]_j, [\mathbf{m}]_i, (\sigma, \tau), \mu, \rho)$ where $\mathsf{pp} \xleftarrow{\$} \mathsf{ParGen}(1^\lambda)$, $[\mathsf{pk}]_j \in \mathcal{S}_{\mathsf{pk}}$, $[\mathbf{m}]_i \in \mathcal{S}_{\mathbf{m}}$, $\mathsf{Verify}([\mathbf{m}]_i, (\sigma, \tau), [\mathsf{pk}]_j) = 1$ and $\mu, \rho \in \mathbb{Z}_p^*$ we have that the output of $\mathsf{ChgRep}([\mathbf{m}]_i, (\sigma, \tau), \mu, \rho, [\mathsf{pk}]_j)$ is $([\mu \cdot \mathbf{m}]_i, \sigma^*)$, with σ^* being a random element in the space of signatures, conditioned on $\mathsf{Verify}([\mu \cdot \mathbf{m}]_i, \sigma^*, \mathsf{ConvertPK}([\mathsf{pk}]_j, \rho)) = 1$.

Theorem 9. The above extension applied to the SPS-EQ from Figure 4 perfectly adapts signatures (under malicious keys in the honest parameter model).

Proof. It follows from the security of SPS-EQ and the definition of perfect adaption for mercurial signatures (Appendix D and H from [18]).

We now formalize the signer-hiding notion and show that our construction satisfies it as it perfectly adapts signatures.

Signer-Hiding. An ABC system supports signer-hiding if for all $\lambda > 0$, all $q > 0$, all $n > 0$, all $t > 0$, all $\mathcal{X}$ with $0 < |\mathcal{X}| \leq t$, all $\emptyset \neq \mathcal{S} \subset \mathcal{X}$ and $\emptyset \neq \mathcal{D} \not\subseteq \mathcal{X}$ with $0 < |\mathcal{D}| \leq t$, and p.p.t adversaries $\mathcal{A}$, the following holds.

$$\mathsf{Pr}\left[\begin{array}{l} \mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, 1^q); \\ \forall\ i \in [n] : (\mathsf{osk}_i, \mathsf{opk}_i) \xleftarrow{\$} \mathsf{OrgKGen}(\mathsf{pp}); \\ (\mathsf{usk}, \mathsf{upk}) \xleftarrow{\$} \mathsf{UsrKGen}(\mathsf{pp}); j \xleftarrow{\$} [n]; \\ (\mathsf{cred}, \top) \xleftarrow{\$} (\mathsf{Obtain}(\mathsf{usk}, \mathsf{opk}_j, \mathcal{X}), \mathsf{Issue}(\mathsf{upk}, \mathsf{osk}_j, \mathcal{X})); \\ j^* \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{Show}}}(\mathsf{pp}, \mathcal{S}, \mathcal{D}, (\mathsf{opk}_i)_{i \in [n]}) \end{array} : j^* = j \right] \leq \frac{1}{n}$$

where the oracle $\mathcal{O}_{\mathsf{Show}}$ is defined as in Section 6.

Theorem 10. If the underlying signature scheme is a SPS-EQ which perfectly adapts signatures (under malicious keys in the honest parameter model), the resulting ABC from Section 7.2 supports signer-hiding.

Proof. Let us first observe that the adversary can guess the bit j^* with probability $1/n$. By definition of perfect adaption, for all tuples $(\mathsf{pp}, [\mathsf{opk}]_j, [\mathbf{m}]_i, (\sigma, \tau), \mu, \rho)$ s.t $(\sigma, \tau) \xleftarrow{\$} \mathsf{Sign}(\mathsf{pp}, \mathsf{osk}_j, [\mathbf{m}]_i)$, we have that $[\mu \cdot \mathbf{m}]_i$ and $[\rho \cdot \mathsf{opk}]_j$ are identically distributed in the message and key spaces, where $([\mu \cdot \mathbf{m}]_i, \sigma^*) \leftarrow \mathsf{ChgRep}([\mathbf{m}]_i, (\sigma, \tau), \mu, \rho, [\mathsf{opk}]_j)$ and $[\rho \cdot \mathsf{opk}]_j \leftarrow \mathsf{ConvertPK}(\mathsf{opk}_j, \rho)$. Furthermore, we also have that σ^* is a random element in the space of signatures conditioned on $\mathsf{Verify}([\mu \cdot \mathbf{m}]_i, \sigma^*, [\rho \cdot \mathsf{opk}]_j) = 1$. Therefore, an adversary with access to $[\mu \cdot \mathbf{m}]_i$, σ^* and $[\rho \cdot \mathsf{opk}]_j$ can only guess the bit j^* with probability at most $1/n$. □

Integration with our ABC scheme. As our NIZK argument is fully adaptive, users can choose the size of the anonymity set (*i.e.,* the set of public keys in the OR-Proof). We find this approach much simpler than using delegatable credentials to achieve a similar result as users do not need to interact with the organizations to compute the NIZK proof nor to adapt the signature. Moreover, there is no need to use pseudonyms for public and secret keys. We essentially compute public key's pseudonyms "on-the-fly" guaranteeing that the signature

adaption is done with respect to a valid public key. In other words, our NIZK argument is a proof of correct randomization, where the same randomizer is used to adapt the signature and generate a pseudonymous public key. A complete figure for the proposed ABC, including the signer-hiding extension as well as NAND proofs, is given in [18] (Appendix I).

Efficiency analysis. As the proof size is $9n - 1$ for an anonymity set of n-authorities, communication bandwidth will no longer be constant. Nevertheless, given the previously mentioned advantages we believe that this is a fair trade-off for the added functionality. In terms of computational cost, it is also substantially more efficient than similar variants (see, for instance, Table 2 from [19]).

8 Comparison of state-of-the-art ABC

We provide comparisons on the efficiency of state-of-the-art ABC and ours (Section 7) on Table 3. For ease of exposition, we list the work from [33] next to ours, and consider an instantiation of it in the CRS model, and using the same ZKPoK's as the ones used in Section 7.

When looking at a whole, the work from Sanders [47] presents very good results while also allowing showings to prove relationships between attributes and to consider malicious keys. Nevertheless, security of the related construction is proven in the GGM model and thus, falls short in that aspect. The same also applies to the works from [39] and [33].

While for the comparisons only the classical setting (credentials are issued by a single authority) was considered, it is worth to mention that [39] does consider multi-authorities. As authors point out, in order to allow multi-authorities they base their construction on aggregate signatures, and obtain the most efficient showing for the users. Their security model follows the game-based approach from [33] but because of the multi-authority setting, they also consider malicious credential issuers, with adaptive corruptions, and collusions with malicious users. Unfortunately, this is done assuming that the keys are *honestly* generated.

[50] uses a set-commitment scheme which alongside an SDH-based signature, leads to a credential system that supports a variety of show proofs for complex statements among which AND and NAND are included. For this reason, we also compare our work with the one from [50] considering NAND showings. In terms of security models, authors provide a formalization for impersonation attacks and prove their scheme secure against impersonation under active and concurrent attacks. The security of their ABC scheme is proven in the standard model and providing a tight reduction.

Considering the different trade-offs, our ABC provides very similar performance when compared to [33] and it is not too distant from the most efficient ones either. Unlike the rest, it can be adapted to different scenarios in case that reducing the verification cost is not needed, and it can also be efficiently adapted to provide revocation features. Furthermore, as for many practical applications the ability to perform AND and NAND showings suffices, we also achieve a good level of expressiveness too. Finally, the signer-hiding feature makes it suitable for scenarios in which the rest of the alternatives struggle.

ABC	[47]	[39]	[50]	[33]	Section 7
Parameters size (n-attributes)					
ek	$(\frac{n^2+n+2}{2})_{\mathbb{G}_1}+n_{\mathbb{G}_2}$	$(2n+2)_{\mathbb{G}_2}$	$(n+1)_{\mathbb{G}_1}+$ $(n+1)_{\mathbb{G}_2}$	$(n+1)_{\mathbb{G}_1}+$ $(n+1)_{\mathbb{G}_2}$	$(n+1)_{\mathbb{G}_1}+$ $(n+1)_{\mathbb{G}_2}$
Cred	$2_{\mathbb{G}_2}$	$4_{\mathbb{G}_1}$	$1_{\mathbb{G}_1}+6_{\mathbb{Z}_p}$	$3_{\mathbb{G}_1}+1_{\mathbb{G}_2}+2_{\mathbb{Z}_p}$	$18_{\mathbb{G}_1}+6_{\mathbb{G}_2}+3_{\mathbb{Z}_p}$
Bandwidth					
Issue	$4_{\mathbb{G}_2}+2_{\mathbb{Z}_p}$	$n_{\mathbb{G}_1}$	$3_{\mathbb{G}_1}+(n+3)_{\mathbb{Z}_p}$	$12_{\mathbb{G}_1}+1_{\mathbb{G}_2}+8_{\mathbb{Z}_p}$	$14_{\mathbb{G}_1}+11_{\mathbb{G}_2}+7_{\mathbb{Z}_p}$
Show	$2_{\mathbb{G}_1}+2_{\mathbb{G}_2}+1_{\mathbb{G}_T}+2_{\mathbb{Z}_p}$	$3_{\mathbb{G}_1}+1_{\mathbb{Z}_p}$	$3_{\mathbb{G}_1}+5_{\mathbb{Z}_p}$	$10_{\mathbb{G}_1}+1_{\mathbb{G}_2}+8_{\mathbb{Z}_p}$	$18_{\mathbb{G}_1}+14_{\mathbb{G}_2}+4_{\mathbb{Z}_p}$
k-of-n attributes (AND)					
Usr	$(2(n\text{-}k)+2)_{\mathbb{G}_1}$,$2_{\mathbb{G}_2}$, **1**	$6_{\mathbb{G}_1}$	$(6+n\text{-}k)_{\mathbb{G}_1}$	$(11+n\text{-}k)_{\mathbb{G}_1}$,$1_{\mathbb{G}_2}$, **8**	$(20+n\text{-}k)_{\mathbb{G}_1}$, $(k\text{-}1)_{\mathbb{G}_2}$,**19**
Ver	$(k+1)_{\mathbb{G}_1}$,$1_{\mathbb{G}_T}$,**5**	$4_{\mathbb{G}_1}$,$2n_{\mathbb{G}_2}$,**3**	$5_{\mathbb{G}_1}$,$(k+1)_{\mathbb{G}_2}$,**3**	$4_{\mathbb{G}_1}$,$(k+1)_{\mathbb{G}_2}$,**10**	$10_{\mathbb{G}_1}$,**16**
k-of-n attributes (NAND)					
Usr	N/A	N/A	$(6+n)_{\mathbb{G}_1}$	N/A	$(31+n)_{\mathbb{G}_1}$, $(9+2k)_{\mathbb{G}_2}$,**19**
Ver	N/A	N/A	$(2k+5)_{\mathbb{G}_1}$, $(k+3)_{\mathbb{G}_2}$,**3**	N/A	$10_{\mathbb{G}_1}$,**17**

Table 3: Efficiency of ABCs considering issuing and showing interactions (the number of pairings is marked in bold).

9 Conclusions and Future Work

Our results explore multiple paths to extend the ABC framework of [33] to include more applications and scenarios where it can be used. In order to improve expressiveness of the set-commitment scheme in [33] we allow openings on sets of attributes disjoint from those possessed by a user. We also enhance efficiency by employing the trick of allowing the prover to compute a proof of exponentiation leaving the verifier only to compute a polynomial division.

Our signature scheme is based on [41] where we adapt the SPS-EQ scheme by alleviating the need to build a QA-NIZK incorporating results from the recent framework of [19]. With this fully adaptive NIZK, we find further interesting applications by looking at equivalence classes on the key-space. We develop a signer-hiding notion to allow a credential-bearing user to hide their issuing organization upon presentation of the credential. As we increasingly see cases of (algorithmic) bias against users, notions such as this are of growing importance. Moreover, we also present interesting directions to integrate revocation features.

We worked in the classical setting where each credential is issued by a single authority. It would be interesting to follow the related work on aggregatable signatures to see if we could lift SPS-EQ to the multi-authority setting.

While our set-commitment scheme is more expressive than [33] it is still less expressive than [50]. Hence, it would be interesting to see if the set-commitment scheme introduced there would yield greater expressiveness to the ABC scheme from this work. Likewise, to verify if the stronger security notions presented here, could enhance the construction in [50].

Acknowledgements. We thank the anonymous reviewers for their valuable feedback. The European Commission partially supported Octavio Perez Kempner's work as part of the CUREX project (H2020-SC1-FA-DTS-2018-1 under grant agreement No 826404).

References

1. Au, M.H., Tsang, P.P., Susilo, W., Mu, Y.: Dynamic universal accumulators for ddh groups and their application to attribute-based anonymous credential systems. In: Fischlin, M. (ed.) Topics in Cryptology – CT-RSA 2009. pp. 295–308. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
2. Backes, M., Hanzlik, L., Kluczniak, K., Schneider, J.: Signatures with Flexible Public Key: Introducing Equivalence Classes for Public Keys. In: Peyrin, T., Galbraith, S. (eds.) Advances in Cryptology – ASIACRYPT 2018. pp. 405–434. Springer International Publishing, Cham (2018)
3. Backes, M., Hanzlik, L., Schneider-Bensch, J.: Membership Privacy for Fully Dynamic Group Signatures. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. p. 2181–2198. CCS '19, Association for Computing Machinery, New York, NY, USA (2019)
4. Baldimtsi, F., Lysyanskaya, A.: Anonymous credentials light. In: Proceedings of the ACM Conference on Computer and Communications Security. pp. 1087–1098 (11 2013)
5. Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., Shacham, H.: Randomizable Proofs and Delegatable Anonymous Credentials. In: Halevi, S. (ed.) Advances in Cryptology - CRYPTO 2009. pp. 108–125. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
6. Blazy, O., Kiltz, E., Pan, J.: (Hierarchical) Identity-Based Encryption from Affine Message Authentication. In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology – CRYPTO 2014. pp. 408–425. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
7. Bobolz, J., Eidens, F., Krenn, S., Ramacher, S., Samelin, K.: Issuer-hiding attribute-based credentials. In: Conti, M., Stevens, M., Krenn, S. (eds.) Cryptology and Network Security. pp. 158–178. Springer International Publishing, Cham (2021)
8. Boneh, D., Bünz, B., Fisch, B.: Batching Techniques for Accumulators with Applications to IOPs and Stateless Blockchains. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology – CRYPTO 2019. pp. 561–586. Springer International Publishing, Cham (2019)
9. Brands, S.A.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press, Cambridge, MA, USA (2000)
10. Bultel, X., Lafourcade, P., Lai, R.W.F., Malavolta, G., Schröder, D., Thyagarajan, S.A.K.: Efficient Invisible and Unlinkable Sanitizable Signatures. In: Lin, D., Sako, K. (eds.) Public-Key Cryptography – PKC 2019. pp. 159–189. Springer International Publishing, Cham (2019)
11. Camenisch, J., Dubovitskaya, M., Haralambiev, K., Kohlweiss, M.: Composable and Modular Anonymous Credentials: Definitions and Practical Constructions. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology – ASIACRYPT 2015. pp. 262–288. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)

12. Camenisch, J., Lysyanskaya, A.: A Signature Scheme with Efficient Protocols. In: Proceedings of the 3rd International Conference on Security in Communication Networks. p. 268–289. SCN'02, Springer-Verlag, Berlin, Heidelberg (2002)
13. Camenisch, J., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: Franklin, M. (ed.) Advances in Cryptology – CRYPTO 2004. pp. 56–72. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
14. Canard, S., Lescuyer, R.: Anonymous Credentials from (Indexed) Aggregate Signatures. In: Proceedings of the 7th ACM Workshop on Digital Identity Management. p. 53–62. DIM '11, Association for Computing Machinery, New York, NY, USA (2011)
15. Canard, S., Lescuyer, R.: Protecting Privacy by Sanitizing Personal Data: A New Approach to Anonymous Credentials. In: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security. p. 381–392. ASIA CCS '13, Association for Computing Machinery, New York, NY, USA (2013)
16. Chase, M., Kohlweiss, M., Lysyanskaya, A., Meiklejohn, S.: Malleable proof systems and applications. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology – EUROCRYPT 2012. pp. 281–300. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
17. Chase, M., Lysyanskaya, A.: On Signatures of Knowledge. In: Dwork, C. (ed.) Advances in Cryptology - CRYPTO 2006. pp. 78–96. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
18. Connolly, A., Lafourcade, P., Perez Kempner, O.: Improved constructions of anonymous credentials from structure-preserving signatures on equivalence classes. Cryptology ePrint Archive, Report 2021/1680 (2021), `https://ia.cr/2021/1680`
19. Couteau, G., Hartmann, D.: Shorter Non-interactive Zero-Knowledge Arguments and ZAPs for Algebraic Languages. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020. pp. 768–798. Springer International Publishing, Cham (2020)
20. Crites, E.C., Lysyanskaya, A.: Delegatable Anonymous Credentials from Mercurial Signatures. In: Matsui, M. (ed.) Topics in Cryptology – CT-RSA 2019. pp. 535–555. Springer International Publishing, Cham (2019)
21. Crites, E.C., Lysyanskaya, A.: Mercurial signatures for variable-length messages. Proceedings on Privacy Enhancing Technologies **2021**(4), 441–463 (2021)
22. Damgård, I.: Efficient concurrent zero-knowledge in the auxiliary string model. In: Preneel, B. (ed.) Advances in Cryptology — EUROCRYPT 2000. pp. 418–430. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
23. Datta, P., Komargodski, I., Waters, B.: Decentralized Multi-Authority ABE for DNFs from LWE. In: Advances in Cryptology – EUROCRYPT 2021. Springer International Publishing (2021)
24. Derler, D., Hanser, C., Slamanig, D.: A New Approach to Efficient Revocable Attribute-Based Anonymous Credentials. In: Groth, J. (ed.) Cryptography and Coding. pp. 57–74. Springer International Publishing, Cham (2015)
25. Derler, D., Hanser, C., Slamanig, D.: Revisiting Cryptographic Accumulators, Additional Properties and Relations to Other Primitives. In: Nyberg, K. (ed.) Topics in Cryptology — CT-RSA 2015. pp. 127–144. Springer International Publishing, Cham (2015)
26. Derler, D., Slamanig, D.: Highly-Efficient Fully-Anonymous Dynamic Group Signatures. In: Proceedings of the 2018 on Asia Conference on Computer and Communications Security. p. 551–565. ASIACCS '18, Association for Computing Machinery, New York, NY, USA (2018)

27. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An Algebraic Framework for Diffie-Hellman Assumptions. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology – CRYPTO 2013. pp. 129–147. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
28. Fuchsbauer, G., Gay, R.: Weakly Secure Equivalence-Class Signatures from Standard Assumptions. In: Abdalla, M., Dahab, R. (eds.) Public-Key Cryptography – PKC 2018. pp. 153–183. Springer International Publishing, Cham (2018)
29. Fuchsbauer, G., Gay, R., Kowalczyk, L., Orlandi, C.: Access Control Encryption for Equality, Comparison, and More. In: Fehr, S. (ed.) Public-Key Cryptography – PKC 2017. pp. 88–118. Springer Berlin Heidelberg, Berlin, Heidelberg (2017)
30. Fuchsbauer, G., Hanser, C., Kamath, C., Slamanig, D.: Practical Round-Optimal Blind Signatures in the Standard Model from Weaker Assumptions. In: Zikas, V., De Prisco, R. (eds.) Security and Cryptography for Networks. pp. 391–408. Springer International Publishing, Cham (2016)
31. Fuchsbauer, G., Hanser, C., Kamath, C., Slamanig, D.: Practical Round-Optimal Blind Signatures in the Standard Model from Weaker Assumptions. In: Zikas, V., De Prisco, R. (eds.) Security and Cryptography for Networks. pp. 391–408. Springer International Publishing, Cham (2016)
32. Fuchsbauer, G., Hanser, C., Slamanig, D.: Practical Round-Optimal Blind Signatures in the Standard Model. In: Gennaro, R., Robshaw, M. (eds.) Advances in Cryptology – CRYPTO 2015. pp. 233–253. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
33. Fuchsbauer, G., Hanser, C., Slamanig, D.: Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials. Journal of Cryptology **32**, 498–546 (2019)
34. Gay, R., Hofheinz, D., Kohl, L.: Kurosawa-Desmedt Meets Tight Security. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology – CRYPTO 2017. pp. 133–160. Springer International Publishing, Cham (2017)
35. Gay, R., Hofheinz, D., Kohl, L., Pan, J.: More Efficient (Almost) Tightly Secure Structure-Preserving Signatures. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018. pp. 230–258. Springer International Publishing, Cham (2018)
36. Ghosh, E., Ohrimenko, O., Papadopoulos, D., Tamassia, R., Triandopoulos, N.: Zero-Knowledge Accumulators and Set Algebra. In: Proceedings, Part II, of the 22nd International Conference on Advances in Cryptology — ASIACRYPT 2016 - Volume 10032. p. 67–100. Springer-Verlag, Berlin, Heidelberg (2016)
37. Hanser, C., Slamanig, D.: Structure-Preserving Signatures on Equivalence Classes and Their Application to Anonymous Credentials. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology – ASIACRYPT 2014. pp. 491–511. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
38. Hofheinz, D.: Adaptive partitioning. In: Coron, J.S., Nielsen, J.B. (eds.) Advances in Cryptology – EUROCRYPT 2017. pp. 489–518. Springer International Publishing, Cham (2017)
39. Hébant, C., Pointcheval, D.: Traceable Constant-Size Multi-Authority Credentials. Cryptology ePrint Archive, Report 2020/657 (2020)
40. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-Size Commitments to Polynomials and Their Applications. In: Abe, M. (ed.) Advances in Cryptology - ASIACRYPT 2010. pp. 177–194. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
41. Khalili, M., Slamanig, D., Dakhilalian, M.: Structure-Preserving Signatures on Equivalence Classes from Standard Assumptions. In: Galbraith, S.D., Moriai, S.

(eds.) Advances in Cryptology – ASIACRYPT 2019. pp. 63–93. Springer International Publishing, Cham (2019)
42. Kiltz, E., Pan, J., Wee, H.: Structure-Preserving Signatures from Standard Assumptions, Revisited. In: Gennaro, R., Robshaw, M. (eds.) Advances in Cryptology – CRYPTO 2015. pp. 275–295. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
43. Morillo, P., Ràfols, C., Villar, J.L.: The Kernel Matrix Diffie-Hellman Assumption. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology – ASIACRYPT 2016. pp. 729–758. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
44. Nguyen, L.: Accumulators from Bilinear Pairings and Applications. In: Menezes, A. (ed.) Topics in Cryptology – CT-RSA 2005. pp. 275–292. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
45. Papamanthou, C., Tamassia, R., Triandopoulos, N.: Optimal verification of operations on dynamic sets. In: Rogaway, P. (ed.) Advances in Cryptology – CRYPTO 2011. pp. 91–110. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
46. Ràfols, C.: Stretching groth-sahai: Nizk proofs of partial satisfiability. In: Dodis, Y., Nielsen, J.B. (eds.) Theory of Cryptography. pp. 247–276. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
47. Sanders, O.: Efficient Redactable Signature and Application to Anonymous Credentials. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) Public-Key Cryptography – PKC 2020. pp. 628–656. Springer International Publishing, Cham (2020)
48. Shoup, V.: Lower Bounds for Discrete Logarithms and Related Problems. In: Proceedings of the 16th Annual International Conference on Theory and Application of Cryptographic Techniques. p. 256–266. EUROCRYPT'97, Springer-Verlag, Berlin, Heidelberg (1997)
49. Sonnino, A., Al-Bassam, M., Bano, S., Meiklejohn, S., Danezis, G.: Coconut: Threshold Issuance Selective Disclosure Credentials with Applications to Distributed Ledgers. In: The Network and Distributed System Security Symposium (NDSS) (2019)
50. Tan, S., Groß, T.: MoniPoly - An Expressive q-SDH-Based Anonymous Attribute-Based Credential System. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2020, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part III. vol. 12493, pp. 498–526. Springer (2020)
51. Thakur, S.: Batching non-membership proofs with bilinear accumulators. IACR Cryptol. ePrint Arch. **2019**, 1147 (2019)
52. Wesolowski, B.: Efficient Verifiable Delay Functions (extended version). Journal of Cryptology (Sep 2020)
53. Zurich, I.R.: Specification of the identity mixer cryptographic library v2.3.0. (2013)