Structure-Preserving Signatures on Equivalence Classes and their Application to Anonymous Credentials

Christian Hanser and Daniel Slamanig

Institute for Applied Information Processing and Communications (IAIK),
Graz University of Technology (TUG), Inffeldgasse 16a, 8010 Graz, Austria
{christian.hanser|daniel.slamanig@tugraz.at}

Abstract. Structure-preserving signatures are a quite recent but important building block for many cryptographic protocols. In this paper, we introduce a new type of structure-preserving signatures, which allows to sign group element vectors and to consistently randomize signatures and messages without knowledge of any secret. More precisely, we consider messages to be (representatives of) equivalence classes on vectors of group elements (coming from a single prime order group), which are determined by the mutual ratios of the discrete logarithms of the representative's vector components. By multiplying each component with the same scalar, a different representative of the same equivalence class is obtained. We propose a definition of such a signature scheme, a security model and give an efficient construction, which is secure in the SXDH setting, where EUF-CMA security holds against generic forgers in the generic group model and the so called class hiding property holds under the DDH assumption.
As a second contribution, we use the proposed signature scheme to build an efficient multi-show attribute-based anonymous credential (ABC) system that allows to encode an arbitrary number of attributes. This is – to the best of our knowledge – the first ABC system that provides constant-size credentials and constant-size showings. To allow an efficient construction in combination with the proposed signature scheme, we also introduce a new, efficient, randomizable polynomial commitment scheme. Aside from these two building blocks, the credential system requires a very short and constant-size proof of knowledge to provide freshness in the showing protocol.

1 Introduction

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

structure-preserving signatures [33,6,1,2,21,5,4] can handle messages which are elements of two groups G_1 and G_2 equipped with a bilinear map, without requiring any prior encoding. Basically, in a structure-preserving signature scheme the public key, the messages and the signatures consist only of group elements and the verification algorithm evaluates a signature by deciding group membership of elements in the signature and by evaluating pairing product equations. Such signature schemes typically allow to sign vectors of group elements (from one of the two groups G_1 and G_2, or mixed) and also support some types of *randomization* (inner, sequential, etc., cf. [1,5]).

Randomization is one interesting feature of signatures, as a given signature can be randomized to another unlinkable version of the signature for the same message. Besides randomizable structure-preserving signatures, there are various other constructions of such signature schemes [24,25,18,43]. We emphasize that although these schemes are randomizable, they are still secure digital signatures in the standard sense (EUF-CMA security).

We are interested in constructions of structure-preserving signature schemes that do *not only* allow randomization of the signature, *but also* allow to randomize the signed message in particular ways. Such signature schemes are particularly interesting for applications in privacy-enhancing cryptographic protocols.

1.1 Contribution

This paper has three contributions: A novel type of structure-preserving signatures defined on equivalence classes on group element vectors, a novel randomizable polynomial commitment scheme, which allows to open factors of the polynomial committed to, and a new construction (type) of multi-show attribute-based anonymous credentials (ABCs), which is instantiated from the first two contributions.

Structure-Preserving Signature Scheme on Equivalence Classes: Inspired by *randomizable signatures*, we introduce a novel variant of structure-preserving signatures. Instead of signing particular message vectors as in other schemes, the scheme *produces signatures on classes of an equivalence relation* $\mathcal{R}$ defined on $(G_1^*)^\ell$ with $\ell > 1$ (where we use G_1^* to denote $G_1 \setminus \{0_{G_1}\}$). More precisely, we consider messages to be (representatives of) equivalence classes on $(G_1^*)^\ell$, which are determined by the mutual ratios of the discrete logarithms of the representative's vector components. By multiplying each component with the same scalar, a different representative of the same equivalence class is obtained. Initially, an equivalence class is signed by signing an arbitrary representative. Later, one can obtain a valid signature for every other representative of this class, without having access to the secret key. Furthermore, we require two representatives of the same class with corresponding signatures to be unlinkable, which we call *class hiding*. We present a definition of such a signature scheme along with game based notions of security and present an efficient construction, which produces short and constant-size signatures that are independent of the message vector length ℓ. In the full version [37], we prove the security of our

construction in the generic group model against generic forgers and the DDH assumption, respectively.

Polynomial Commitments with Factor Openings: We propose a new, efficient, randomizable polynomial commitment scheme. It is computationally binding, unconditionally hiding, allows to commit to monic, reducible polynomials and is represented by an element of a bilinear group. It allows to open factors of committed polynomials and re-randomization (i.e., multiplication with a scalar) does not change the polynomial committed to, but requires only a consistent randomization of the witnesses involved in the factor openings. We present a definition as well as a construction of such a polynomial commitment scheme. In the full version [37], we give a security model in which we also prove the construction secure.

A Multi-Show Attribute-Based Anonymous Credential (ABC) System: We describe a new way to build multi-show ABCs (henceforth, we will only write ABCs) as an application of the first two contributions. From another perspective, the signature scheme allows to consistently randomize a vector of group elements and its signature. So, it seems natural to use this property to achieve unlinkability during the showings of an ABC system. To enable a compact attribute representation, which is compatible with the randomization property of the signature scheme, we encode the attributes to polynomials and commit to them using the introduced polynomial commitment scheme. During the issuing, the obtainer is, then, given a set of attributes and the credential, which is a message (vector) consisting of the polynomial commitment and the generator of the group plus the corresponding signature. During a showing, a subset of the issued attributes can be shown by opening the corresponding factors of the committed polynomial. The unlinkability of showings is achieved through the inherent re-randomization properties of the signature scheme and the polynomial commitment scheme, which are compatible to each other. Furthermore, to provide freshness during a showing, we require a very small, constant-size proof of knowledge. We emphasize that our approach to construct ABCs is very different from existing approaches, as we use *neither* zero-knowledge proofs for proving the possession of a signature *nor* for selectively disclosing attributes during showings. Recall that existing approaches rely on signature schemes that allow to sign vectors of attributes and use efficient zero-knowledge proofs to show possession of a signature and to prove relations about the signed attributes during a showing.

Interestingly, in our construction *the size of credentials as well as the size of the showings* are *independent of the number of attributes in the ABC system*, i.e., a small, constant number of group elements. This is, to the best of our knowledge, the first ABC system with this feature. The proposed ABC system is secure in a security model adapted from [23,8,26,27], where we refer the reader to the full version [37] for the proofs and the security model. Finally, we compare our system to other existing multi- and one-show ABC approaches. Although we are only dealing with multi-show credentials, for the sake of completeness, we

also compare our approach to the one-show (i.e., linkable) anonymous credentials of Brands [20] (and, thus, also its provably secure generalization [12]).

1.2 Related Work

In [16], Blazy et al. present *signatures on randomizable ciphertexts* (based on linear encryption [18]) using a variant of Waters' signature scheme [43]. Basically, anyone given a signature on a ciphertext can *randomize the ciphertext* and adapt the signature accordingly, while maintaining public verifiability and neither knowing the signing key nor the encrypted message. However, as these signatures only allow to randomize the ciphertexts and not the underlying plaintexts, this approach is not useful for our purposes.

Another somewhat related approach is the proofless variant of the Chaum-Pedersen signature [31] which is used to build *self-blindable* certificates by Verheul in [42]. The resulting so called certificate as well as the initial message can be randomized using the same scalar, preserving the validity of the certificate. This approach works for the construction in [42], but it does not represent a secure signature scheme (as also observed in [42]) due to its homomorphic property and the possibility of efficient existential forgeries.

Homomorphic signatures for network coding [19] allow to sign any subspace of a vector space by producing a signature for every basis vector with respect to the same (file) identifier. Consequently, the message space consists of identifiers and vectors. These signatures are homomorphic, meaning that given a sequence of scalar and signature pairs $(\beta_i, \sigma_i)_{i=1}^{\ell}$ for vectors $\boldsymbol{v}_i$, one can publicly compute a signature for the vector $\boldsymbol{v} = \sum_{i=1}^{\ell} \beta_i \boldsymbol{v}_i$ (this is called derive). If one was using a unique identifier per signed vector $\boldsymbol{v}$, then such linearly homomorphic signatures would support a functionality similar to the one provided by our scheme, i.e., publicly compute signatures for vectors $\boldsymbol{v}' = \beta \boldsymbol{v}$ (although they are not structure-preserving). It is also known that various existing constructions, e.g., [19,10] are *strong context hiding*, meaning that original and derived signatures are unlinkable. Nevertheless, this does not help in our context, which is due to the following argument: If we do not restrict every single signed vector to a unique identifier, the signature schemes are homomorphic, which is not compatible with our unforgeability goal. If we apply this restriction, however, then we are not able to achieve *class hiding* as all signatures can be linked to the initial signature by the unique identifier. We note that the same arguments also apply to structure-preserving linearly homomorphic signatures [40].

The aforementioned context hiding property is also of interest in more general classes of homomorphic (also called malleable) signature schemes (defined in [7] and refined in [9]). In [29], the authors discuss malleable signatures that allow to derive a signature σ' on a message $m' = T(m)$ for an "allowable" transformation T, when given a signature σ for a message m. This can be considered as a generalization of signature schemes, such as quotable [10] or redactable signatures [38] with the additional property of being context hiding. The authors note that for messages being pseudonyms and transformations that transfer one pseudonym into another pseudonym, such malleable signatures can be used to

construct anonymous credential systems. They also demonstrate how to build delegatable anonymous credential systems [15,14]. The general construction in [29] relies on malleable-ZKPs [28] and is not really efficient, even when instantiated with Groth-Sahai proofs [35]. Although it is conceptually totally different from our approach, we note that by viewing our scheme in a different way, our scheme fits into their definition of malleable signatures (such that their SigEval algorithm takes only a single message vector with corresponding signature and a single allowable transformation). However, firstly, our construction is far more efficient than their approach (and in particular really practical) and, secondly, [29] only focuses on transformations of single messages (pseudonyms) and does not consider multi-show attribute-based anonymous credentials at all (which is the main focus of our construction).

Signatures providing randomization features [24,25,18] along with efficient proofs of knowledge of committed values can be used to generically construct ABC systems. The most prominent approaches based on Σ-protocols are CL credentials [24,25]. With the advent of Groth-Sahai proofs, which allow (efficient) non-interactive proofs in the CRS model without random oracles, various constructions of so called delegatable (hierarchical) anonymous credentials have been proposed [15,14]. These provide per definition a non-interactive showing protocol, i.e., the show and verify algorithms do not interact when demonstrating the possession of a credential. In [34], Fuchsbauer presented the first delegatable anonymous credential system that also provides a non-interactive delegation protocol based on so called commuting signatures and verifiable encryption. We note that although such credential systems with non-interactive protocols extend the scope of applications of anonymous credentials, the most common use-case (i.e., authentication and authorization), essentially relies on interaction (to provide freshness/liveness). We emphasize that our goal is not to construct non-interactive anonymous credentials. Nevertheless, one could generically convert our proposed system to a non-interactive one: in the ROM using Fiat-Shamir or by replacing our single Σ-proof for freshness with a Groth-Sahai proof without random oracles, which is, however, out of scope of this paper.

2 Preliminaries

Definition 1 (Bilinear Map). Let G_1, G_2 and G_T be cyclic groups of prime order p, where G_1 and G_2 are additive and G_T is multiplicative. Let P and P' generate G_1 and G_2, respectively. We call $e : G_1 \times G_2 \to G_T$ *bilinear map* or *pairing* if it is efficiently computable and the following conditions hold:

Bilinearity: $e(aP, bP') = e(P, P')^{ab} = e(bP, aP') \quad \forall a, b \in \mathbb{Z}_p$
Non-degeneracy: $e(P, P') \neq 1_{G_T}$, i.e., $e(P, P')$ generates G_T.

If $G_1 = G_2$, then e is called *symmetric* (Type-1) and *asymmetric* (Type-2 or Type-3) otherwise. For Type-2 pairings there is an efficiently computable isomorphism $\Psi : G_2 \to G_1$, whereas for Type-3 pairings no such efficient isomorphism is assumed to exist. Note that Type-3 pairings are currently the optimal choice [30], with respect to efficiency and security trade-off.

Definition 2 (Decisional Diffie Hellman Assumption (DDH)). Let p be a prime of bitlength κ, G be a group of prime order p generated by P and let $(P, aP, bP, cP) \in G^4$, where $a, b, c \in_R \mathbb{Z}_p^*$. Then, for every PPT adversary $\mathcal{A}$ distinguishing between $(P, aP, bP, abP) \in G^4$ and $(P, aP, bP, cP) \in G^4$ is infeasible, i.e., there is a negligible function $\epsilon(\cdot)$ such that

$$|\Pr[\mathtt{true} \leftarrow \mathcal{A}(P, aP, bP, abP)] - \Pr[\mathtt{true} \leftarrow \mathcal{A}(P, aP, bP, cP)]| \leq \epsilon(\kappa).$$

Definition 3 (Symmetric External DH Assumption (SXDH) [13]). Let G_1, G_2 and G_T be three distinct cyclic groups of prime order p and $e : G_1 \times G_2 \rightarrow G_T$ be a pairing. Then, the SXDH assumption states that in both groups G_1 and G_2 the DDH assumption holds.

Note that the SXDH assumption formalizes Type-3 pairings, i.e., the absence of an efficiently computable isomorphism between G_1 and G_2 as well as between G_2 and G_1.

Definition 4 (Bilinear Group Generator). Let BGGen be a PPT algorithm which takes a security parameter κ and generates a bilinear group $\mathsf{BG} = (p, G_1, G_2, G_T, e, P, P')$ in the SXDH setting, where the common group order p of the groups G_1, G_2 and G_T is a prime of bitlength κ, e is a pairing and P as well as P' are generators of G_1 and G_2, respectively.

Definition 5 (t-Strong DH Assumption (t-SDH) [17]). Let p be a prime of bitlength κ, G be a group of prime order p generated by $P \in G$, $\alpha \in_R \mathbb{Z}_p^*$ and let $(\alpha^i P)_{i=0}^t \in G^{t+1}$ for some $t > 0$. Then, for every PPT adversary $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\left(c, \frac{1}{\alpha + c} P\right) \leftarrow \mathcal{A}((\alpha^i P)_{i=0}^t)\right] \leq \epsilon(\kappa) \quad \text{for some } c \in \mathbb{Z}_p \setminus \{-\alpha\}.$$

This assumption turns out to be very useful in bilinear groups (Type-1 or Type-2 setting). However, in a Type-3 setting (SXDH assumption), where the groups G_1 and G_2 are strictly separated, the presence of a pairing does not give any additional benefit. This is due to the fact that the problem instance is given either in G_1 or in G_2. As our constructions rely on the SXDH assumption, we introduce the following modified assumption, which can be seen as the natural counterpart for a Type-3 setting [30]:

Definition 6 (co-t-Strong DH Assumption (co-t-SDH$_i^*$)). Let G_1 and G_2 be two groups of prime order p (which has bitlength κ) generated by $P_1 \in G_1$ and $P_2 \in G_2$, respectively. Let $\alpha \in_R \mathbb{Z}_p^*$ and let $(\alpha^j P_1)_{j=0}^t \in G_1^{t+1}$ and $(\alpha^j P_2)_{j=0}^t \in G_2^{t+1}$ for some $t > 0$. Then, for every PPT adversary $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\left(c, \frac{1}{\alpha + c} P_i\right) \leftarrow \mathcal{A}((\alpha^j P_1)_{j=0}^t, (\alpha^j P_2)_{j=0}^t)\right] \leq \epsilon(\kappa) \quad \text{for some } c \in \mathbb{Z}_p \setminus \{-\alpha\}.$$

Note that for a compact representation, we make a slight abuse of notation, where it should be interpreted as $P_1 = P$ and $P_2 = P'$. Obviously, we have co-t-SDH$_i^* \leq_p t$-SDH in group G_i. The t-SDH assumption was originally proven to be secure in the generic group model in [17, Theorem 5.1] and further studied in [32]. The proof is done in a Type-2 pairing setting, where an efficiently computable isomorphism $\Psi : G_2 \to G_1$ exists. In the proof, the adversary is given the problem instance in group G_2 and is allowed to obtain encodings of elements in G_1 through isomorphism queries. As we are in a Type-3 setting, there is no such efficiently computable isomorphism. Thus, the problem instance given to the adversary must contain all corresponding elements in both groups G_1 and G_2. Then, the generic group model proof for the co-t-SDH$_i^*$ assumption can be done analogously to the proof in [17, proof of Theorem 5.1]. The main difference is that instead of querying the isomorphism, the adversary must compute the same sequence of computations performed in one group in the other group, in order to obtain an element containing the same discrete logarithm, which, however, preserves the asymptotic number of queries.

3 Structure-preserving Signatures on Equivalence Classes

We are looking for an efficient, randomizable structure-preserving signature scheme for vectors with arbitrary numbers of group elements that allows to randomize messages and signatures consistently in the public. It seems natural to consider such messages as representatives of certain equivalence classes and to perform randomization via a change of representatives. Before we can introduce such a signature scheme and give an efficient construction, we detail these equivalence classes.

All elements of a vector $(M_i)_{i=1}^{\ell} \in (G_1^*)^{\ell}$ (for some prime order group G_1, where we write G_1^* for $G_1 \setminus \{0_{G_1}\}$) share different mutual ratios. These ratios depend on their discrete logarithms and are invariant under the operation $\gamma : \mathbb{Z}_p^* \times (G_1^*)^{\ell} \to (G_1^*)^{\ell}$ with $(s, (M_i)_{i=1}^{\ell}) \mapsto s(M_i)_{i=1}^{\ell}$. Thus, we can use this invariance to partition the set $(G_1^*)^{\ell}$ into classes using the following equivalence relation:

$$\mathcal{R} = \{(M, N) \in (G_1^*)^{\ell} \times (G_1^*)^{\ell} : \exists s \in \mathbb{Z}_p^* \text{ such that } N = s \cdot M\} \subseteq (G_1^*)^{2\ell}.$$

It is easy to verify that $\mathcal{R}$ is indeed an equivalence relation given that G_1 has prime order. When signing an equivalence class $[M]_{\mathcal{R}}$ with our scheme, one actually signs an arbitrary representative $(M_i)_{i=1}^{\ell}$ of class $[M]_{\mathcal{R}}$. The scheme, then, allows to choose different representatives and to update corresponding signatures in the public, i.e., without any secret key. Thereby, one of our goals is to guarantee that two message-signature pairs on the same equivalence class cannot be linked. Note that such an approach only seems to work for structure-preserving signature schemes, where we have no direct access to scalars. Otherwise, if we wanted to sign vectors of elements of $\mathbb{Z}_p^*$, the direct access to the scalars would allow us to decide class membership efficiently. This is also the reason, why we

subsequently define the class hiding property with respect to a random-message instead of a chosen-message attack.

3.1 Defining the Signature Scheme

Now, we formally define a signature scheme for the above equivalence relation and its required security properties.

Definition 7 (Structure-Preserving Signature Scheme for Equivalence Relation $\mathcal{R}$ (SPS-EQ-$\mathcal{R}$)). An SPS-EQ-$\mathcal{R}$ scheme consists of the following polynomial time algorithms:

$\mathsf{BGGen}_{\mathcal{R}}(\kappa)$**:** Is a probabilistic bilinear group generation algorithm, which on input a security parameter κ outputs a bilinear group BG.

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

$\mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk})$**:** Is a probabilistic algorithm, which on input a representative M of an equivalence class $[M]_{\mathcal{R}}$ and a secret key sk, outputs a signature σ for the equivalence class $[M]_{\mathcal{R}}$ (using randomness y).

$\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \rho, \mathsf{pk})$**:** Is a probabilistic algorithm, which on input a representative M of an equivalence class $[M]_{\mathcal{R}}$, the corresponding signature σ, a scalar ρ and a public key pk, returns an updated message-signature pair $(\hat{M}, \hat{\sigma})$ (using randomness $\hat{y}$). Here, $\hat{M}$ is the new representative $\rho \cdot M$ and $\hat{\sigma}$ its updated signature.

$\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk})$**:** Is a deterministic algorithm, which given a representative M, a signature σ and a public key pk, outputs `true` if σ is a valid signature for the equivalence class $[M]_{\mathcal{R}}$ under pk and `false` otherwise.

When one does not care about which new representative is chosen, $\mathsf{ChgRep}_{\mathcal{R}}$ can be seen as consistent randomization of a signature and its message using randomizer ρ without invalidating the signature on the equivalence class. The goal is that the signature resulting from $\mathsf{ChgRep}_{\mathcal{R}}$ is indistinguishable from a newly issued signature for the new representative of the same class.

For security, we require the usual correctness property for signature schemes, but instead of single messages we consider the respective equivalence class and the correctness of $\mathsf{ChgRep}_{\mathcal{R}}$. More formally, we require:

Definition 8 (Correctness). An SPS-EQ-$\mathcal{R}$ scheme is called *correct*, if for all security parameters $\kappa \in \mathbb{N}$, for all $\ell > 1$, for all bilinear groups $\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(\kappa)$, all key pairs $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, \ell)$ and for all $M \in (G_1^*)^{\ell}$ it holds that

$$\mathsf{Verify}_{\mathcal{R}}(\mathsf{ChgRep}_{\mathcal{R}}(M, \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk}), \rho, \mathsf{pk}), \mathsf{pk}) = \mathtt{true} \quad \forall \rho \in \mathbb{Z}_p^*.$$

Furthermore, we require a notion of EUF-CMA security. In contrast to the standard definition of EUF-CMA security, we consider a natural adaption, i.e., outputting a valid message-signature pair, corresponding to an unqueried equivalence class, is considered to be a forgery.

Definition 9 (EUF-CMA). An SPS-EQ-$\mathcal{R}$ scheme is called *existentially unforgeable under adaptively chosen-message attacks*, if for all PPT algorithms $\mathcal{A}$ having access to a signing oracle $\mathcal{O}(\mathsf{sk}, M)$, there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[\begin{array}{c} \mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(\kappa), \quad (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, \ell) \\ (M^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}(\mathsf{sk}, \cdot)}(\mathsf{pk}) : \\ [M^*]_{\mathcal{R}} \neq [M]_{\mathcal{R}} \quad \forall M \in Q \quad \wedge \quad \mathsf{Verify}_{\mathcal{R}}(M^*, \sigma^*, \mathsf{pk}) = \mathtt{true} \end{array}\right] \leq \epsilon(\kappa),$$

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

Subsequently, we let Q be a list for keeping track of queried messages M and make use of the following oracles:

$\mathcal{O}^{RM}(\ell)$**:** A random-message oracle, which on input a message vector length ℓ, picks a message $M \xleftarrow{R} (G_1^*)^\ell$, appends M to Q and returns it.

$\mathcal{O}^{RoR}(\mathsf{sk}, \mathsf{pk}, b, M)$**:** A real-or-random oracle taking input a bit b and a message M. If $M \notin Q$, it returns $\perp$. On the first valid call, it chooses $R \xleftarrow{R} (G_1^*)^\ell$, computes $\mathcal{M} \leftarrow \big((M, \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk})), (R, \mathsf{Sign}_{\mathcal{R}}(R, \mathsf{sk}))\big)$ and returns $\mathcal{M}[b]$. Any next call for $M' \neq M$ will return $\perp$ and $\mathsf{ChgRep}_{\mathcal{R}}(\mathcal{M}[b], \rho, \mathsf{pk})$ otherwise, where $\rho \xleftarrow{R} \mathbb{Z}_p^*$.

Definition 10 (Class Hiding). An SPS-EQ-$\mathcal{R}$ scheme on $(G_1^*)^\ell$ is called *class hiding*, if for every PPT adversary $\mathcal{A}$ with oracle access to $\mathcal{O}^{RM}$ and $\mathcal{O}^{RoR}$, there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{c} \mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(\kappa),\ b \xleftarrow{R} \{0,1\},\ (\mathsf{state}, \mathsf{sk}, \mathsf{pk}) \leftarrow \mathcal{A}(\mathsf{BG}, \ell), \\ \mathcal{O} \leftarrow \{\mathcal{O}^{RM}(\ell), \mathcal{O}^{RoR}(\mathsf{sk}, \mathsf{pk}, b, \cdot)\},\ b^* \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{state}, \mathsf{sk}, \mathsf{pk}) : \\ b^* = b \end{array}\right] - \frac{1}{2} \leq \epsilon(\kappa).$$

Here, the adversary is in the role of a signer, who issues signatures on random messages (in the sense of a random message attack) and can derive signatures for arbitrary representatives of queried classes. Observe that, if the adversary was able to pick messages on its own, e.g., knows the discrete logarithms of the group elements or puts identical group elements on different positions of the message vectors, it would trivially be able to distinguish the classes. Consequently, we define class hiding in a random message attack game and the random sampling of messages makes the probability of identical message elements at different positions negligible.

Definition 11 (Security). An SPS-EQ-$\mathcal{R}$ scheme is *secure*, if it is correct, EUF-CMA secure and class hiding.

3.2 Our Construction

In our construction, we sign vectors of $\ell > 1$ elements of G_1^*, where the public key only consists of elements in G_2 and we require the SXDH assumption to hold.

The signature consists of four group elements, where three elements are from G_1 and one element is from G_2. Two signature elements (Z_1, Z_2) are aggregates of the message vector under ℓ elements of the private key. In order to prevent an additive homomorphism on the signatures, we introduce a randomizer $y \in \mathbb{Z}_p^*$, multiply one aggregate with it and introduce two additional values $Y = yP$ and $Y' = yP'$. The latter elements (besides eliminating the homomorphic property) prevent simple forgeries, where Y' contains an aggregation of the public keys $X', X'_1, \ldots, X'_\ell$ in G_2. This is achieved by verifying whether Y and Y' contain the same unknown discrete logarithms during verification. Our construction lets us switch to another representative $\hat{M} = \rho M$ of M by multiplying M and (Z_1, Z_2) with the respective scalar ρ. Furthermore, a consistent re-randomization of ρZ_2, Y and Y' with a scalar $\hat{y}$ yields a signature $\hat{\sigma}$ for $\hat{M}$ that is unlinkable to the signature σ of M. In Scheme 1, we present the detailed construction of the SPS-EQ-$\mathcal{R}$ scheme.

$\mathsf{BGGen}_{\mathcal{R}}(\kappa)$**:** Given a security parameter κ, output $\mathsf{BG} \leftarrow \mathsf{BGGen}(\kappa)$.

$\mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, \ell)$**:** Given a bilinear group description BG and vector length $\ell > 1$, choose $x \xleftarrow{R} \mathbb{Z}_p^*$ and $(x_i)_{i=1}^{\ell} \xleftarrow{R} (\mathbb{Z}_p^*)^{\ell}$, set the secret key as $\mathsf{sk} \leftarrow (x, (x_i)_{i=1}^{\ell})$, compute the public key $\mathsf{pk} \leftarrow (X', (X'_i)_{i=1}^{\ell}) = (xP', (x_i x P')_{i=1}^{\ell})$ and output $(\mathsf{sk}, \mathsf{pk})$.

$\mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk})$**:** On input a representative $M = (M_i)_{i=1}^{\ell} \in (G_1^*)^{\ell}$ of equivalence class $[M]_{\mathcal{R}}$ and secret key $\mathsf{sk} = (x, (x_i)_{i=1}^{\ell})$, choose $y \xleftarrow{R} \mathbb{Z}_p^*$ and compute

$$Z_1 \leftarrow x \sum_{i=1}^{\ell} x_i M_i, \quad Z_2 \leftarrow y \sum_{i=1}^{\ell} x_i M_i \quad \text{and} \quad (Y, Y') \leftarrow y \cdot (P, P').$$

Then, output $\sigma = (Z_1, Z_2, Y, Y')$ as signature for the equivalence class $[M]_{\mathcal{R}}$.

$\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \rho, \mathsf{pk})$**:** On input a representative $M = (M_i)_{i=1}^{\ell} \in (G_1^*)^{\ell}$ of equivalence class $[M]_{\mathcal{R}}$, the corresponding signature $\sigma = (Z_1, Z_2, Y, Y')$, $\rho \in \mathbb{Z}_p^*$ and public key pk, this algorithm picks $\hat{y} \xleftarrow{R} \mathbb{Z}_p^*$ and returns $(\hat{M}, \hat{\sigma})$, where $\hat{\sigma} \leftarrow (\rho Z_1, \hat{y}\rho Z_2, \hat{y} Y, \hat{y} Y')$ is the update of signature σ for the new representative $\hat{M} \leftarrow \rho \cdot (M_i)_{i=1}^{\ell}$.

$\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk})$**:** Given a representative $M = (M_i)_{i=1}^{\ell} \in (G_1^*)^{\ell}$ of equivalence class $[M]_{\mathcal{R}}$, a signature $\sigma = (Z_1, Z_2, Y, Y')$ and public key $\mathsf{pk} = (X', (X'_i)_{i=1}^{\ell})$, check whether

$$\prod_{i=1}^{\ell} e(M_i, X'_i) \stackrel{?}{=} e(Z_1, P') \quad \wedge \quad e(Z_1, Y') \stackrel{?}{=} e(Z_2, X') \quad \wedge \quad e(P, Y') \stackrel{?}{=} e(Y, P')$$

and if this holds output `true` and `false` otherwise.

Scheme 1: A Construction of an SPS-EQ-$\mathcal{R}$ Scheme

Note that a signature resulting from $\mathsf{ChgRep}_{\mathcal{R}}$ is indistinguishable from a new signature on the same class using the new representative (it can be viewed as issuing a signature with randomness $y \cdot \hat{y}$).

3.3 Security of Our Construction

In our construction, message vectors are elements of $(G_1^*)^{\ell}$, public keys are only available in G_2 and signatures are elements of G_1 and G_2. Furthermore, we rely

on the SXDH assumption, and it seems very hard (to impossible) to analyze the EUF-CMA security of the scheme via a reductionist proof using accepted non-interactive assumptions. Abe et al. [3] show that for optimally short structure-preserving signatures, i.e., three-element signatures, such reductions using non-interactive assumptions cannot exist. But right now, it is not entirely clear how structure-preserving signatures for equivalence relation $\mathcal{R}$ fit into these results and if the lower bounds from [2] also apply. Independently of this, it appears that a reduction to a (non-interactive) assumption is not possible, since due to the class hiding property the winning condition cannot be checked efficiently (without substantially weakening the unforgeability notion). Therefore, we chose to prove the EUF-CMA security of our construction using a direct proof in the generic group model such as for instance the proof of Abe et al. [2, Lemma 1] (cf. [37] for the proof).

Now, we state the security of the signature scheme. The corresponding proofs can be found in the full version [37].

Theorem 1. *The SPS-EQ-$\mathcal{R}$ scheme in Scheme 1 is correct.*

Theorem 2. *In the generic group model for SXDH groups, Scheme 1 is an EUF-CMA secure SPS-EQ-$\mathcal{R}$ scheme.*

Theorem 3. *If the DDH assumption holds in G_1, Scheme 1 is a class hiding SPS-EQ-$\mathcal{R}$ scheme.*

Taking everything together, we obtain the following corollary:

Corollary 1. *The SPS-EQ-$\mathcal{R}$ scheme in Scheme 1 is secure.*

4 Polynomial Commitments with Factor Openings

In [39], Kate et al. introduced the notion of constant-size polynomial commitments. The authors present two distinct commitment schemes, where one is computationally hiding ($\mathsf{PolyCommit}_{\mathsf{DL}}$) and the other one is unconditionally hiding ($\mathsf{PolyCommit}_{\mathsf{Ped}}$). These constructions are very generic, as they allow to construct witnesses for opening arbitrary evaluations of committed polynomials.

Yet, we emphasize that in practical scenarios (and especially in our constructions) it is often sufficient to consider the roots of polynomials for encodings and to open factors of the polynomial instead of arbitrary evaluations. Moreover, we need a polynomial commitment scheme that is easily randomizable. Therefore, we introduce the subsequent commitment scheme for monic, reducible polynomials. Instead of opening evaluations, it allows to open factors of committed polynomials. Hence, we call this type of commitment *polynomial commitment with factor openings*. Our construction is unconditionally hiding, computationally binding and more efficient than the Pedersen polynomial commitment construction $\mathsf{PolyCommit}_{\mathsf{Ped}}$ of [39]. Now, we briefly present this construction, which we denote by PolyCommitFO.

$\mathsf{Setup}_{\mathsf{PC}}(\kappa, t)$**:** It takes input a security parameter $\kappa \in \mathbb{N}$ and a maximum polynomial degree $t \in \mathbb{N}$. It runs $\mathsf{BG} \leftarrow \mathsf{BGGen}(\kappa)$, picks $\alpha \xleftarrow{R} \mathbb{Z}_p^*$ and outputs $\mathsf{sk} \leftarrow \alpha$ as well as $\mathsf{pp} \leftarrow (\mathsf{BG}, (\alpha^i P)_{i=1}^t, (\alpha^i P')_{i=1}^t)$.

$\mathsf{Commit}_{\mathsf{PC}}(\mathsf{pp}, f(X))$**:** It takes input the public parameters pp and a monic, reducible polynomial $f(X) \in \mathbb{Z}_p[X]$ with $\deg f \leq t$. It picks $\rho \xleftarrow{R} \mathbb{Z}_p^*$, computes the commitment $\mathcal{C} \leftarrow \rho \cdot f(\alpha)P \in G_1$ and outputs $(\mathcal{C}, O)$ with opening information $O \leftarrow (\rho, f(X))$. [1]

$\mathsf{Open}_{\mathsf{PC}}(\mathsf{pp}, \mathcal{C}, \rho, f(X))$**:** It takes input the public parameters pp, a polynomial commitment $\mathcal{C}$, the randomizer ρ used for $\mathcal{C}$ and the committed polynomial $f(X)$ and outputs $(\rho, f(X))$.

$\mathsf{Verify}_{\mathsf{PC}}(\mathsf{pp}, \mathcal{C}, \rho, f(X))$**:** It takes input the public parameters pp, a polynomial commitment $\mathcal{C}$, the randomizer ρ used for $\mathcal{C}$ and the committed polynomial $f(X)$. It verifies whether $\rho \stackrel{?}{\neq} 0 \quad \wedge \quad \mathcal{C} \stackrel{?}{=} \rho \cdot f(\alpha)P$ holds and outputs `true` on success and `false` otherwise.

$\mathsf{FactorOpen}_{\mathsf{PC}}(\mathsf{pp}, \mathcal{C}, f(X), g(X), \rho)$**:** It takes input the public parameters pp, a polynomial commitment $\mathcal{C}$, the committed polynomial $f(X)$, a factor $g(X)$ of $f(X)$ and the randomizer ρ used for $\mathcal{C}$. It computes $h(X) \leftarrow \frac{f(X)}{g(X)}$, the witness $\mathcal{C}_h \leftarrow \rho \cdot h(\alpha)P$ and outputs $(g(X), \mathcal{C}_h)$.

$\mathsf{VerifyFactor}_{\mathsf{PC}}(\mathsf{pp}, \mathcal{C}, g(X), \mathcal{C}_h)$**:** It takes input the public parameters pp, a polynomial commitment $\mathcal{C}$ to a polynomial $f(X)$, a polynomial $g(X)$ of positive degree and a corresponding witness $\mathcal{C}_h$. It verifies that $g(X)$ is a factor of $f(X)$ by checking whether $\mathcal{C}_h \stackrel{?}{\neq} 0_{G_1} \quad \wedge \quad e(\mathcal{C}_h, g(\alpha)P') \stackrel{?}{=} e(\mathcal{C}, P')$ holds. It outputs `true` on success and `false` otherwise.

In analogy to the security notion in [39], a polynomial commitment scheme with factor openings is *secure* if it is *correct*, *polynomial binding*, *factor binding*, *factor sound*, *witness sound* and *hiding*. The above scheme can be proven secure under the co-t-SDH$_1^*$ assumption. For the security model and the formal proofs of security, we refer the reader to the full version [37]. Note that one can also define a scheme based on the co-t-SDH$_2^*$ assumption with $\mathcal{C} \in G_1$ and $\mathcal{C}_h \in G_2$. Although this would improve the performance of $\mathsf{VerifyFactor}_{\mathsf{PC}}$, we define it differently to reduce the computational complexity of the prover in the ABC system in Section 5.3. Also note that we use the co-t-SDH$_1^*$ assumption in a static way, as t is a system parameter and fixed a priori. Finally, observe that $\mathsf{sk} = \alpha$ must remain unknown to the committer (and, thus, the setup has to be run by a TTP), since it is a trapdoor commitment scheme otherwise.

5 Building an ABC System

In this section, we present an application of the signature scheme and the polynomial commitment scheme introduced in the two previous sections, by using

[1] Subsequently, we use $f(\alpha)P$ as short-hand notation for $\sum_{i=0}^{\deg f} f_i \cdot \alpha^i P$ even if α is unknown.

them as basic building blocks for an ABC system. ABC systems are usually constructed in one of the following two ways. Firstly, they can be built from blind signatures: A user obtains a blind signature from some issuer on (commitments to) attributes and, then, shows the signature, provides the shown attributes and proves the knowledge of all unrevealed attributes [20,12]. The drawback of such a blind signature approach is that such credentials can only be shown once in an unlinkable fashion (*one-show*). Secondly, anonymous credentials supporting an arbitrary number of unlinkable showings (*multi-show*) can be obtained in a similar vein using different types of signatures: A user obtains a signature on (commitments to) attributes, then *randomizes* the signature (such that the resulting signature is unlinkable to the issued one) and proves in zero-knowledge the possession of a signature and the correspondence of this signature with the shown attributes as well as the undisclosed attributes [24,25]. Our approach also achieves multi-show ABCs, but differs from the latter significantly: We randomize the signature and the message and, thus, do not require costly zero-knowledge proofs (which are, otherwise, at least linear in the number of shown/encoded attributes) for the showing of a credential.

Subsequently, we start by discussing the model of ABCs. Then, we provide an intuition for our construction in Section 5.2 and present the scheme in Section 5.3. In Section 5.4, we discuss the security of the construction. Finally, we give a performance comparison with other existing approaches in Section 5.5.

5.1 Abstract Model of ABCs

In an ABC system there are different organizations issuing credentials to different users. Users can then anonymously demonstrate possession of these credentials to verifiers. Such a system is called multi-show ABC system when transactions (issuing and showings) carried out by the same user cannot be linked. A credential cred_i for user i is issued by an organization j for a set $\mathbb{A} = \{(\mathtt{attr}_k, \mathtt{attrV}_k)\}_{k=1}^{n}$ of attribute labels $\mathtt{attr}_k$ and values $\mathtt{attrV}_k$. By $\#\mathbb{A}$ we mean the size of $\mathbb{A}$, which is defined to be the sum of cardinalities of all second components $\mathtt{attrV}_k$ of the tuples in $\mathbb{A}$. Moreover, we denote by $\mathbb{A}' \sqsubseteq \mathbb{A}$ a subset of the credential's attributes. In particular, for every k, $1 \leq k \leq n$, we have that either $(\mathtt{attr}_k, \mathtt{attrV}_k)$ is missing or $(\mathtt{attr}_k, \mathtt{attrV}'_k)$ with $\mathtt{attrV}'_k \subseteq \mathtt{attrV}_k$ is present. A showing with respect to $\mathbb{A}'$ only proves that a valid credential for $\mathbb{A}'$ has been issued, but reveals nothing beyond (selective disclosure).

We note that in some ABC system constructions, the entire key generation is executed by the Setup algorithm. However, we split these algorithms into three algorithms to make the presentation more flexible and convenient.

Definition 12 (Attribute-Based Anonymous Credential System). An *attribute-based anonymous credential (ABC) system* consists of the following polynomial time algorithms:

Setup: A probabilistic algorithm that gets a security parameter κ, an upper bound t for the size of attribute sets and returns the public parameters pp.

OrgKeyGen: A probabilistic algorithm that takes input the public parameters pp and $j \in \mathbb{N}$, produces and outputs a key pair $(\mathsf{osk}_j, \mathsf{opk}_j)$ for organization j.

UserKeyGen: A probabilistic algorithm that takes input the public parameters pp and $i \in \mathbb{N}$, produces and outputs a key pair $(\mathsf{usk}_i, \mathsf{upk}_i)$ for user i.

(Obtain, Issue): These (probabilistic) algorithms are run by user i and organization j, who interact during execution. Obtain takes input the public parameters pp, the user's secret key usk_i, an organization's public key opk_j and an attribute set $\mathbb{A}$ of size $\#\mathbb{A} \leq t$. Issue takes input the public parameters pp, the user's public key upk_i, an organization's secret key osk_j and an attribute set $\mathbb{A}$ of size $\#\mathbb{A} \leq t$. At the end of this protocol, Obtain outputs a credential cred_i for $\mathbb{A}$ for user i.

(Show, Verify): These (probabilistic) algorithms are run by user i and a verifier, who interact during execution. Show takes input public parameters pp, the user's secret key usk_i, the organization's public key opk_j, a credential cred_i for set $\mathbb{A}$ of size $\#\mathbb{A} \leq t$ and a second set $\mathbb{A}' \sqsubseteq \mathbb{A}$. Verify takes input pp, the public key opk_j and a set $\mathbb{A}'$. At the end of the protocol, Verify outputs `true` or `false` indicating whether the credential showing was accepted or not.

An ABC system is called *secure* if it is *correct*, *unforgeable* and *anonymous* (for formal definitions, we refer the reader to the full version [37]).

5.2 Intuition of Our Construction

Our construction of ABCs is based on the proposed signature scheme, on polynomial commitments with factor openings and on a *single* constant-size proof of knowledge (PoK) for guaranteeing freshness. In contrast to this, the number of proofs of knowledge in other ABC systems, like [23,20] and related approaches, is linear in the number of shown attributes. Nevertheless, aside from selective disclosure of attributes, they allow to prove statements about non-revealed attribute values, such as AND, OR and NOT, interval proofs, as well as conjunctions and disjunctions of the aforementioned. The expressiveness that we achieve with our construction, can be compared to existing alternative constructions of ABCs [26,27]. Namely, our construction supports selective disclosure as well as AND statements about attributes. Thereby, a user can either open some attributes and their corresponding values or solely prove that some attributes are encoded in the respective credential without revealing their concrete values. Furthermore, one may associate sets of values to attributes, such that one is not required to reveal the full attribute value, but only pre-defined "statements" about the attribute value such as {"01.01.1980", "> 16", "> 18"} for attribute `birthdate`. This allows us to emulate proving properties about attribute values and, thus, enhances the expressiveness of the system.

Credential Representation: In our construction, a credential cred_i of user i is a vector of two group elements (C_1, P) together with a signature under the proposed signature scheme (see Section 3.2). During a showing, the credential gets randomized, which is easily achieved by changing the representative. The meaning of its values will be discussed subsequently.

Attribute Representation: We use PolyCommitFO (cf. Section 4) to commit to a polynomial, which encodes a set of attributes $\mathbb{A} = \{(\mathtt{attr}_k, \mathtt{attrV}_k)\}_{k=1}^{n}$ (where the encoding is inspired from [36]). This commitment is represented by the credential value C_1.

Now, we show how we use polynomials to encode this set of attributes and values. Thereby, we use a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p^*$ and the following encoding function to generate the polynomials:

$$\mathsf{enc} : \mathbb{A} \mapsto \prod_{k=1}^{n} \prod_{M \in \mathtt{attrV_k}} \big(X - H(\mathtt{attr}_k \| M)\big).$$

This function is used to encode the set $\mathbb{A}$ in the issued credential, the shown attributes $\mathbb{A}'$ as well as its complement:

$$\overline{\mathbb{A}'} = \{(\mathtt{attr}, \mathtt{attrV} \setminus \mathtt{attrV}') : (\mathtt{attr}, \mathtt{attrV}) \in \mathbb{A}, (\mathtt{attr}, \mathtt{attrV}') \in \mathbb{A}'\} \cup \{(\mathtt{attr}', \mathtt{attrV}) \in \mathbb{A} : (\mathtt{attr}', \cdot) \notin \mathbb{A}'\}$$

in every showing. The idea is that the credential includes a commitment to the encoding of $\mathbb{A}$ and that showings include a witness of the encoding of $\overline{\mathbb{A}'}$ (without opening it) as well as $\mathbb{A}'$ in plain for which the encoding is then recomputed by the verifier. To compute these values, we use the PolyCommitFO public parameters pp, which allow an evaluation of these polynomials in G_1 and G_2 at $\alpha \in \mathbb{Z}_p^*$ (without knowing the trapdoor α). Then, the verifier checks whether the multiplicative relationship $\mathsf{enc}(\mathbb{A}) = \mathsf{enc}(\mathbb{A}') \cdot \mathsf{enc}(\overline{\mathbb{A}'})$ between the polynomials is satisfied by checking the multiplicative relationship between the corresponding commitments and witnesses via a pairing equation. More precisely, the commitment to the encoding of $\mathbb{A}$ is computed as $C_1 = r_i \cdot \mathsf{enc}(\mathbb{A})(\alpha)P$ with r_i being the secret key of user i. We note that since no entity knows α, we must compute

$$C_1 \leftarrow r_i \cdot \mathsf{enc}(\mathbb{A})(\alpha)P = r_i \cdot \sum_{i=0}^{t} e_i \alpha^i P, \quad \text{with } \mathsf{enc}(\mathbb{A}) = \sum_{i=0}^{t} e_i X^i \in \mathbb{Z}_p[X].$$

The verification of a credential, when showing $\mathbb{A}'$, requires checking whether the following holds:

$$\mathsf{VerifyFactor}_{\mathsf{PC}}(\mathsf{pp}, C_1, \mathsf{enc}(\mathbb{A}'), \mathcal{C}_{\overline{\mathbb{A}'}}) \stackrel{?}{=} \mathtt{true},$$

where $\mathcal{C}_{\overline{\mathbb{A}'}} = r_i \cdot \mathsf{enc}(\overline{\mathbb{A}'})(\alpha)P$ is part of the showing. A showing, then, simply amounts to randomizing C_1, opening a product of factors of the committed polynomial (representing the selective disclosure), providing a consistently randomized witness of the complementary polynomial and performing a small, constant-size PoK of the randomizer for freshness, as we will see soon.

Example: For the reader's convenience, we include an example of a set $\mathbb{A}$. We are given a user with the following set of attributes and values:

$$\mathbb{A} = \{(\mathtt{birthdate}, \{\text{"01.01.1980"}, \text{" > 18"}\}), (\mathtt{drivinglicense}, \{\#, \mathtt{car}\})\}.$$

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

$$\mathbb{A}' = \{(\texttt{drivinglicense}, \{\#\})\}$$
$$\overline{\mathbb{A}'} = \{(\texttt{birthdate}, \{\text{"}\texttt{01.01.1980}\text{"}, \text{"} > \texttt{18}\text{"}\}), (\texttt{drivinglicense}, \{\texttt{car}\})\}.$$

Freshness: We have to guarantee that no valid showing transcript can be replayed by someone not in possession of the credential and the user's secret key. To do so, we require the user to conduct a proof of knowledge $\mathsf{PoK}\{\gamma : C_2 = \gamma P\}$ of the discrete logarithm of the second component $C_2 = \rho P$ of a credential, i.e., the value ρ, in the showing protocol. This guarantees that we have a fresh challenge for every showing.

In order to prove the anonymity of the ABC system, we need a little trick. We modify the aforementioned PoK and require that the user delivers a proof of knowledge $\mathsf{PoK}\{\gamma : \ Q = \gamma P \ \vee \ C_2 = \gamma P\}$, where Q is an additional value in the public parameters pp with unknown discrete logarithm q. Consequently, the user needs to conduct the second part of the proof honestly, while simulating the one for Q. In the proof of anonymity, this allows us to let the challenger know q and simulate showings without knowledge of the discrete logarithm of C_2, which is required for our reduction to work. Due to the nature of the OR proof, this cannot be detected by the adversary.

5.3 The Construction of the ABC System

Now, we present our ABC system in Scheme 2, where we use the notation $X \leftarrow f(X)$ to indicate that the value of X is overwritten by the result of the evaluation of $f(X)$. Note that if a check does not yield `true`, the respective algorithm terminates with a failure and the algorithm Verify accepts only if $\mathsf{VerifyFactor}_{\mathsf{PC}}$ and $\mathsf{Verify}_{\mathcal{R}}$ return `true` as well as PoK is valid. Also note that the first move in the showing protocol can be combined with the first move of the proof of knowledge. Therefore, the showing protocol consists of a total of three moves.

5.4 Security

In the full version [37], we introduce a security model for attribute-based anonymous credentials and we provide formal proofs for the following:

Theorem 4. *Scheme 2 is correct.*

Theorem 5. *If* $\mathsf{PolyCommitFO}$ *is factor-sound, H is a collision-resistant hash function, Scheme 1 is secure and the DLP is hard in G_1, then Scheme 2 is unforgeable.*

Theorem 6. *If Scheme 1 is class hiding, then Scheme 2 is anonymous.*

Taking everything together, we obtain the following corollary:

Setup: Given (κ, t), run $\mathsf{pp}' = (\mathsf{BG}, (\alpha^i P)_{i=1}^t, (\alpha^i P')_{i=1}^t) \leftarrow \mathsf{Setup}_{\mathsf{PC}}(\kappa, t)$ and let $H : \{0,1\}^* \to \mathbb{Z}_p^*$ be a collision-resistant hash function used inside $\mathsf{enc}(\cdot)$. Finally, choose $Q \xleftarrow{R} G_1$ and output $\mathsf{pp} \leftarrow (H, \mathsf{enc}, Q, \mathsf{pp}')$.

OrgKeyGen: Given pp and $j \in \mathbb{N}$, return $(\mathsf{osk}_j, \mathsf{opk}_j) \leftarrow \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, 2)$.

UserKeyGen: Given pp and $i \in \mathbb{N}$, pick $r_i \xleftarrow{R} \mathbb{Z}_p^*$, set $R_i \leftarrow r_i P$ and return $(\mathsf{usk}_i, \mathsf{upk}_i) \leftarrow (r_i, R_i)$.

(Obtain, Issue): Obtain and Issue interact in the following way:

$\mathsf{Issue}(\mathsf{pp}, \mathsf{upk}_i, \mathsf{osk}_j, \mathbb{A})$		$\mathsf{Obtain}(\mathsf{pp}, \mathsf{usk}_i, \mathsf{opk}_j, \mathbb{A})$
$e(C_1, P') \stackrel{?}{=} e(R_i, \mathsf{enc}(\mathbb{A})(\alpha)P')$	$\xleftarrow{C_1}$	$C_1 \leftarrow r_i \cdot \mathsf{enc}(\mathbb{A})(\alpha)P$
$\sigma \leftarrow \mathsf{Sign}_{\mathcal{R}}((C_1, P), \mathsf{osk}_j)$	$\xrightarrow{\sigma}$	$\mathsf{Verify}_{\mathcal{R}}((C_1, P), \sigma, \mathsf{opk}_j) \stackrel{?}{=} \mathtt{true}$
		$\mathsf{cred}_i \leftarrow ((C_1, P), \sigma)$

(Show, Verify): Show and Verify interact in the following way:

$\mathsf{Verify}(\mathsf{pp}, \mathsf{opk}_j, \mathbb{A}')$		$\mathsf{Show}(\mathsf{pp}, \mathsf{usk}_i, \mathsf{opk}_j, (\mathbb{A}, \mathbb{A}'), \mathsf{cred}_i)$
		$\rho \xleftarrow{R} \mathbb{Z}_p^*$
		$\mathsf{cred}_i' \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(\mathsf{cred}_i, \rho, \mathsf{opk}_j)$
$\Big[\mathsf{VerifyFactor}_{\mathsf{PC}}(\mathsf{pp}', C_1, \mathsf{enc}(\mathbb{A}'), \mathcal{C}_{\overline{\mathbb{A}'}}) \wedge$	$\xleftarrow{\mathsf{cred}_i', \mathcal{C}_{\overline{\mathbb{A}'}}}$	$\mathcal{C}_{\overline{\mathbb{A}'}} \leftarrow (\rho \cdot \mathsf{usk}_i) \cdot \mathsf{enc}(\overline{\mathbb{A}'})(\alpha)P$
$\mathsf{Verify}_{\mathcal{R}}(\mathsf{cred}_i', \mathsf{opk}_j)\Big] \stackrel{?}{=} \mathtt{true}$		
	$\xleftrightarrow{\mathsf{PoK}\{\gamma : Q = \gamma P \vee C_2 = \gamma P\}}$	

where $\mathsf{cred}_i' = ((C_1, C_2), \sigma)$.

Scheme 2: A Multi-Show ABC System

Corollary 2. *Scheme 2 is a secure ABC system.*

Note that in the proof of Theorem 5, we can distinguish whether a forgery goes back to a signature forgery of Scheme 1 or not. The reason for this is that the knowledge extractor of the PoK gives us the possibility to extract the used credential, which allows us to determine whether a showing is based on a queried credential (and, in further consequence, on a queried signature) or not. Hence, we are able to efficiently check the winning condition of the EUF-CMA game.

5.5 Efficiency Analysis and Comparison

We provide a brief comparison with other ABC approaches and for completeness also include the most popular one-show approach. As other candidates for multi-show ABCs, we take the Camenisch-Lysyanskaya schemes [23,24,25] as well as schemes from BBS$^+$ signatures [18,11] which cover a broad class of ABC schemes from randomizable signature schemes with efficient proofs of knowledge. Furthermore, we take two alternative multi-show ABC constructions [26,27] as well as Brands' approach [20] (also covering the provable secure version [12]) for the sake of completeness, although latter only provides one-show ABCs. We omit other approaches such as [8] that only allow a single attribute per credential. We also omit approaches that achieve more efficient showings for existing ABC systems only in very special cases such as for attribute values that come from a very small set (and are, thus, hard to compare). For instance, the approach in [22] for CL credentials in the strong RSA setting (encoding attributes as prime

numbers) or in a pairing-based setting using BBS$^+$ credentials [41] (encoding attributes using accumulators), where the latter additionally requires very large public parameters (one BB signature [15] for every possible attribute value).

Table 1 gives an overview of these systems. Thereby, Type-1 and Type-2 refer to bilinear group settings with Type-1 and Type-2 pairings, respectively. In a stronger sense, XDH as well as SXDH stand for bilinear group settings, where the former requires the external Diffie-Hellman assumption and the latter requires the SXDH assumption to hold. Furthermore, G_q denotes a group of prime order q (e.g., a prime order subgroup of $\mathbb{Z}_p^*$ or of an elliptic curve group). By $|G|$, we mean the bitlength of the representation of an element from group G and the value c is a constant specified to be approximately 510 bits in [26]. We emphasize that, in contrast to other approaches, such as [25,27], our construction only requires a small and constant number of pairing evaluations in all protocol steps. Note that in the issuing step we always assume a computation of $O(L)$ for the user, as we assume that the user checks the validity of the obtained credential on issuing (most of the approaches, including ours, have cost $O(1)$ if this verification is omitted).

Table 1. Comparison of various approaches to ABC systems.

	Parameter Size (L attributes)			**Issuing**			**Showing** (k-of-L attributes)		
	Setting	pp	Credential Size	Issuer	User	Com	Verifier	User	Com
[23,24]	sRSA	$O(L)$	$O(1)$	$3\|\mathbb{Z}_N\|$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L-k)$
[25]	Type-1	$O(L)$	$O(L)$	$(2L+2)\|G_1\|$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L)$
[18]	Type-2	$O(L)$	$O(1)$	$\|G_1\|+22\|\mathbb{Z}_q\|$	$O(L)$	$O(L)$	$O(1)$	$O(L)$	$O(L)$	$O(L)$
[26]	Type-2	$O(1)$	$O(L)$	$L\|G_1\|+c+\|G_2\|$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(1)$	$O(1)$
[27]	XDH	$O(L)$	$O(L)$	$(2L+2)(\|G_1\|+\|\mathbb{Z}_p\|)$	$O(L)$	$O(L)$	$O(L)$	$O(k)$	$O(k)$	$O(k)$
[20]	G_q	$O(L)$	$O(1)$	$2\|G_q\|+2\|\mathbb{Z}_q\|$	$O(L)$	$O(L)$	$O(1)$	$O(k)$	$O(k)$	$O(L-k)$
Our	SXDH	$O(L)$	$O(1)$	$4\|G_1\|+\|G_2\|$	$O(L)$	$O(L)$	$O(1)$	$O(k)$	$O(L-k)$	$O(1)$

6 Future Work

The proposed signature scheme seems to be powerful and there might be other applications that could benefit, like blind signatures or verifiably-encrypted signatures. We leave a detailed study and the analysis of such applications as future work. Future work also includes constructing revocable and delegatable anonymous credentials from this new approach to ABCs. Furthermore, it is an interesting question whether the proposed construction is already optimal, whether such signatures can be built for other interesting relations and whether it is possible to construct such signature schemes whose unforgeability can be proven under possible non-interactive assumptions or even to show that this is impossible.

Acknowledgments. This work has been supported by the European Commission through project FP7-FutureID, grant agreement number 318424. We thank the anonymous referees for their helpful comments.

References

1. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-Preserving Signatures and Commitments to Group Elements. In: CRYPTO. LNCS,

vol. 6223, pp. 209–236 (2010)
2. Abe, M., Groth, J., Haralambiev, K., Ohkubo, M.: Optimal Structure-Preserving Signatures in Asymmetric Bilinear Groups. In: CRYPTO. LNCS, vol. 6841, pp. 649–666 (2011)
3. Abe, M., Groth, J., Ohkubo, M.: Separating Short Structure-Preserving Signatures from Non-interactive Assumptions. In: ASIACRYPT. LNCS, vol. 7073, pp. 628–646 (2011)
4. Abe, M., Groth, J., Ohkubo, M., Tibouchi, M.: Structure-Preserving Signatures from Type II Pairings. In: CRYPTO (1). LNCS, vol. 8616, pp. 390–407 (2014)
5. Abe, M., Groth, J., Ohkubo, M., Tibouchi, M.: Unified, Minimal and Selectively Randomizable Structure-Preserving Signatures. In: TCC. LNCS, vol. 8349, pp. 688–712 (2014)
6. Abe, M., Haralambiev, K., Ohkubo, M.: Signing on Elements in Bilinear Groups for Modular Protocol Design. IACR Cryptology ePrint Archive (2010)
7. Ahn, J.H., Boneh, D., Camenisch, J., Hohenberger, S., Shelat, A., Waters, B.: Computing on Authenticated Data. In: TCC. LNCS, vol. 7194, pp. 1–20 (2012)
8. Akagi, N., Manabe, Y., Okamoto, T.: An Efficient Anonymous Credential System. In: FC. LNCS, vol. 5143, pp. 272–286 (2008)
9. Attrapadung, N., Libert, B., Peters, T.: Computing on Authenticated Data: New Privacy Definitions and Constructions. In: ASIACRYPT. LNCS, vol. 7658, pp. 367–385 (2012)
10. Attrapadung, N., Libert, B., Peters, T.: Efficient completely context-hiding quotable and linearly homomorphic signatures. In: PKC. LNCS, vol. 7778, pp. 386–404 (2013)
11. Au, M.H., Susilo, W., Mu, Y.: Constant-Size Dynamic k-TAA. In: SCN. LNCS, vol. 4116, pp. 111–125 (2006)
12. Baldimtsi, F., Lysyanskaya, A.: Anonymous Credentials Light. In: CCS. ACM (2013)
13. Ballard, L., Green, M., de Medeiros, B., Monrose, F.: Correlation-Resistant Storage via Keyword-Searchable Encryption. IACR Cryptology ePrint Archive (2005)
14. Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., Shacham, H.: Randomizable Proofs and Delegatable Anonymous Credentials. In: CRYPTO. LNCS, vol. 5677, pp. 108–125 (2009)
15. Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: P-signatures and Noninteractive Anonymous Credentials. In: TCC. LNCS, vol. 4948, pp. 356–374 (2008)
16. Blazy, O., Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Signatures on Randomizable Ciphertexts. In: PKC. LNCS, vol. 6571, pp. 403–422 (2011)
17. Boneh, D., Boyen, X.: Short Signatures Without Random Oracles. In: EUROCRYPT. LNCS, vol. 3027, pp. 56–73 (2004)
18. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: CRYPTO. LNCS, vol. 3152, pp. 41–55 (2004)
19. Boneh, D., Freeman, D.M., Katz, J., Waters, B.: Signing a Linear Subspace: Signature Schemes for Network Coding. In: PKC. LNCS, vol. 5443, pp. 68–87 (2009)
20. Brands, S.: Rethinking public-key Infrastructures and Digital Certificates: Building in Privacy. MIT Press (2000)
21. Camenisch, J., Dubovitskaya, M., Haralambiev, K.: Efficient Structure-Preserving Signature Scheme from Standard Assumptions. In: SCN. LNCS, vol. 7485, pp. 76–94 (2012)
22. Camenisch, J., Groß, T.: Efficient Attributes for Anonymous Credentials. ACM Trans. Inf. Syst. Secur. 15(1), 4 (2012)

23. Camenisch, J., Lysyanskaya, A.: An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In: EUROCRYPT. LNCS, vol. 2045, pp. 93–118 (2001)
24. Camenisch, J., Lysyanskaya, A.: A Signature Scheme with Efficient Protocols. In: SCN. LNCS, vol. 2576, pp. 268–289 (2002)
25. Camenisch, J., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: CRYPTO. LNCS, vol. 3152, pp. 56–72 (2004)
26. Canard, S., Lescuyer, R.: Anonymous credentials from (indexed) aggregate signatures. In: DIM. pp. 53–62. ACM (2011)
27. Canard, S., Lescuyer, R.: Protecting privacy by sanitizing personal data: a new approach to anonymous credentials. In: ASIACCS. pp. 381–392. ACM (2013)
28. Chase, M., Kohlweiss, M., Lysyanskaya, A., Meiklejohn, S.: Malleable Proof Systems and Applications. In: EUROCRYPT. LNCS, vol. 7237, pp. 281–300 (2012)
29. Chase, M., Kohlweiss, M., Lysyanskaya, A., Meiklejohn, S.: Malleable Signatures: Complex Unary Transformations and Delegatable Anonymous Credentials. IACR Cryptology ePrint Archive (2013)
30. Chatterjee, S., Menezes, A.: On cryptographic protocols employing asymmetric pairings - the role of ψ revisited. Discrete Applied Mathematics 159(13), 1311–1322 (2011)
31. Chaum, D., Pedersen, T.P.: Wallet Databases with Observers. In: CRYPTO. LNCS, vol. 740, pp. 89–105 (1992)
32. Cheon, J.H.: Security Analysis of the Strong Diffie-Hellman Problem. In: EUROCRYPT. LNCS, vol. 4965, pp. 1–11 (2006)
33. Fuchsbauer, G.: Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. IACR Cryptology ePrint Archive (2009)
34. Fuchsbauer, G.: Commuting Signatures and Verifiable Encryption. In: EUROCRYPT. LNCS, vol. 6632, pp. 224–245 (2011)
35. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: EUROCRYPT. LNCS, vol. 4965, pp. 415–432 (2008)
36. Hanser, C., Slamanig, D.: Blank Digital Signatures. IACR Cryptology ePrint Archive, Report 2013/130 (2013)
37. Hanser, C., Slamanig, D.: Structure-Preserving Signatures on Equivalence Classes and their Application to Anonymous Credentials. Cryptology ePrint Archive, Report 2014/705 (2014)
38. Johnson, R., Molnar, D., Song, D.X., Wagner, D.: Homomorphic Signature Schemes. In: CT-RSA. LNCS, vol. 2271, pp. 244–262 (2002)
39. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-Size Commitments to Polynomials and Their Applications. In: ASIACRYPT. LNCS, vol. 6477, pp. 177–194 (2010)
40. Libert, B., Peters, T., Joye, M., Yung, M.: Linearly Homomorphic Structure-Preserving Signatures and Their Applications. In: CRYPTO (2). LNCS, vol. 8043, pp. 289–307 (2013)
41. Sudarsono, A., Nakanishi, T., Funabiki, N.: Efficient Proofs of Attributes in Pairing-Based Anonymous Credential System. In: PETS. LNCS, vol. 6794, pp. 246–263 (2011)
42. Verheul, E.R.: Self-Blindable Credential Certificates from the Weil Pairing. In: ASIACRYPT. LNCS, vol. 2248, pp. 533–551 (2001)
43. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: EUROCRYPT. LNCS, vol. 3494, pp. 114–127 (2005)